AVEVA™ Work Tasks

2023 SP1

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of AVEVA Group Limited. No liability is assumed with respect to the use of the information contained herein.

Although precaution has been taken in the preparation of this documentation, AVEVA assumes no responsibility for errors or omissions. The information in this documentation is subject to change without notice and does not represent a commitment on the part of AVEVA. The software described in this documentation is furnished under a license agreement. This software may be used or copied only in accordance with the terms of such license agreement. AVEVA, the AVEVA logo and logotype, OSIsoft, the OSIsoft logo and logotype, ArchestrA, Avantis, Citect, DYNSIM, eDNA, EYESIM, InBatch, InduSoft, InStep, IntelaTrac, InTouch, Managed PI, OASyS, OSIsoft Advanced Services, OSIsoft Cloud Services, OSIsoft Connected Services, OSIsoft EDS, PIPEPHASE, PI ACE, PI Advanced Computing Engine, PI AF SDK, PI API, PI Asset Framework, PI Audit Viewer, PI Builder, PI Cloud Connect, PI Connectors, PI Data Archive, PI DataLink, PI DataLink Server, PI Developers Club, PI Integrator for Business Analytics, PI Interfaces, PI JDBC Driver, PI Manual Logger, PI Notifications, PI ODBC Driver, PI OLEDB Enterprise, PI OLEDB Provider, PI OPC DA Server, PI OPC HDA Server, PI ProcessBook, PI SDK, PI Server, PI Square, PI System, PI System Access, PI Vision, PI Visualization Suite, PI Web API, PI WebParts, PI Web Services, PRiSM, PRO/II, PROVISION, ROMeo, RLINK, RtReports, SIM4ME, SimCentral, SimSci, Skelta, SmartGlance, Spiral Software, WindowMaker, WindowViewer, and Wonderware are trademarks of AVEVA and/or its subsidiaries. All other brands may be trademarks of their respective owners.

U.S. GOVERNMENT RIGHTS

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the license agreement with AVEVA Group Limited or its subsidiaries and as provided in DFARS 227.7202, DFARS 252.227-7013, FAR 12-212, FAR 52.227-19, or their successors, as applicable.

AVEVA Legal Resources: https://www.aveva.com/en/legal/

AVEVA Third Party Software Notices and Licenses: https://www.aveva.com/en/legal/third-party-software-license/

Contents

Welcome to AVEVA Work Tasks **60**
Legal information **60**
Contact information **61**
Acknowledgements **61**
AjaxMin 62
AmqpNet Lite 62
AngleSharp 64
AngleSharp.Css 65
AngularJS 65
AntiXSS 66
AutoMapper 66
Azure.Identity 66
Azure.Security.KeyVault.Secrets 67
bootstrap 67
bootstrap-datepicker 68
chilkat-x64 70
CodeMirror 73
ComponentArt Charting 73
crypto-js 73
datejs 74
EasyNetQ 74
ExcelDataReader 74
FastAndroidCamera 75
HtmlSanitizer 77
Intersoft WebGrid 78
JavaScriptEngineSwitcher 81
jquery 83
jquery dotdotdot 84
jquery easing 84
jshint 85
jszip 85
Kendo UI 93
Knockout 94
Knockout-Kendo 94
LumiSoft.Net 94
Microsoft.Web.WebView2 95
Newtonsoft.Json 95
Oracle.ManagedDataAccess 96
Oracle Providers for ASP.NET 96
Owin 96
Paho MQTT 98
Pako 99
polly.extensions.http 99

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

RabbitMQ.Client . . . 99
SendGrid . . . 100
Spin js . . . 100
SQLite . . . 100
sqlite-net SQLCipher . . . 101
StarkbankEcdsa . . . 101
Telerik® UI for ASP.NET AJAX . . . 102
tensorspace . . . 126
Unity.Abstractions . . . 128
Unity.Container . . . 130
WindowsBase . . . 133
WiX Toolset . . . 133
xamarin.essentials . . . 134
xam.plugin.battery . . . 135
Xml2Json . . . 135
ZXing . . . 135
Other Components . . . 138

What's new in AVEVA Work Tasks? . . . 145
2023 Releases . . . 146
AVEVA Work Tasks 2023 SP1 Release Notes . . . 146
System Requirements . . . 146
Product Compatibility . . . 150
ISO Details . . . 150
Installation Instructions . . . 151
Upgrading to AVEVA Work Tasks 2023 SP1 . . . 153
Key Features . . . 153
General Enhancements . . . 154
Issues Resolved . . . 155
Known Issues . . . 155
AVEVA Work Tasks Pro . . . 156

Administrator Guide . . . 158
Architecture . . . 158
Core Components . . . 166
Microsoft Outlook Integration . . . 168
System Requirements to Install Outlook Integration . . . 169
Outlook Integration Best Practices . . . 169
Outlook Integration Licensing Information . . . 169
Scalability and Availability . . . 169
High Availability and Load on AVEVA Work Tasks Web Servers . . . 170
High Availability and Load on the Services . . . 170
High Availability and Load on the Services and Web Servers . . . 170
High Availability of the Services and High Load on Web Servers . . . 171
Setting up Intranet and Extranet Web Fronts . . . 171
Partitioning Database . . . 175
Overview . . . 175

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Partitioning with Central Configuration . . . 176
Partitioning with Partition Manager . . . 177
Verifying Partition . . . 178
Maintenance Activities . . . 179
Editions . . . 180
Single Server . . . 180
Deployment of AVEVA Work Tasks Web Server, AVEVA Work Tasks Server, and AVEVA Work Tasks Database Server on a Single Machine . . . 180
AVEVA Work Tasks Web Server and AVEVA Work Tasks Server on Different Machines . . . 182
AVEVA Work Tasks Web Server, AVEVA Work Tasks Server, and AVEVA Work Tasks Database Server on Different Machines . . . 184
AVEVA Work Tasks Web Server Farm, Work Tasks Server, and AVEVA Work Tasks Database Server on Different Machines . . . 186
Single Server Licensing Information . . . 188
Load Balanced Server . . . 188
Load Balanced Server Architecture . . . 189
Load Balanced Server Installation Scenarios . . . 192
Single Server Capable of Adding Machines at Any Time . . . 193
Web Server Farm with Clustered Work Tasks Services and Databases on Different Machines . . . 195
AVEVA Work Tasks Web Server Farm Connecting to AVEVA Work Tasks Services with AVEVA Work Tasks Databases on a Different Machine . . . 197
Behavior of AVEVA Work Tasks Services on Enterprise Server . . . 199
Tools Used in Enterprise Server . . . 201
Load Balanced Server Limitations . . . 202
Load Balanced Server Licensing Information . . . 203
Installing Different Editions . . . 203
Layout Editing Modes . . . 203
Identifying Server Roles . . . 204
Identifying User Source . . . 204
Making Database Selection . . . 205
Setting up Communication . . . 205
Integrating Email . . . 205
Setting up Ports . . . 205
Cookies Created by Web Application . . . 207
Prerequisites . . . 208
Configure Internet Information Services . . . 210
Enable and Verify MSDTC . . . 213
Configure Oracle Data Handler . . . 214
Install Single Server . . . 215
Single Machine Installation . . . 216
High-Level View of Installation . . . 216
Installation . . . 218
Install AVEVA Work Tasks . . . 218
Start the Farm Configuration Wizard . . . 218
Configure the License Server . . . 219
Select the type of installation . . . 220
Create Farm Database . . . 220
Define Security Settings . . . 223

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Define Services and Global Settings . . . 224
Provision Services . . . 225
Create a Repository DataSource . . . 225
Create a Repository . . . 227
Verify the Installation . . . 237
Web Server and AVEVA Work Tasks Server on Different Machines . . . 239
High-Level View of the Installation . . . 239
Installation . . . 241
Install AVEVA Work Tasks . . . 241
Start the Farm Configuration Wizard . . . 241
Configure the License Server . . . 242
Select the type of installation . . . 243
Create Farm Database . . . 243
Define Security Settings . . . 246
Define Services and Global Settings . . . 247
Provision Services . . . 248
Create a Repository DataSource . . . 248
Create a Repository . . . 251
Verify the Installation . . . 260
Web Server, AVEVA Work Tasks Server, and Database Server on Different Machines . . . 262
High-Level View of Server Installation . . . 262
High-Level View of Client Installation . . . 264
Installation . . . 266
AVEVA Work Tasks Server . . . 266
Install AVEVA Work Tasks . . . 266
Start the Farm Configuration Wizard . . . 266
Configure the License Server . . . 267
Select the type of installation . . . 268
Create Farm Database . . . 269
Define Security Settings . . . 271
Define Services and Global Settings . . . 273
Define Central Configuration . . . 274
Provision Services . . . 275
AVEVA Work Tasks Client . . . 275
Install AVEVA Work Tasks . . . 275
Start the Farm Configuration Wizard . . . 275
Configure the License Server . . . 276
Select Client Installation . . . 277
Connect to an Existing Farm Database . . . 277
Define and Provision the Enterprise Console . . . 279
Create a Repository DataSource . . . 279
Create a Repository . . . 281
Verify the Installation . . . 291
Web Farm (Multiple Web servers), AVEVA Work Tasks Server, and Database on Different Machines . . . 293
High-Level View of Server Installation . . . 293
High-Level View of Client Installation . . . 295
Installation . . . 297
AVEVA Work Tasks Server . . . 297

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Install AVEVA Work Tasks . . . 297
Start the Farm Configuration Wizard . . . 297
Configure the License Server . . . 298
Select the type of installation . . . 299
Create Farm Database . . . 300
Define Security Settings . . . 302
Define Services and Global Settings . . . 304
Define Central Configuration . . . 305
Provision Services . . . 306
AVEVA Work Tasks Client . . . 306
Install AVEVA Work Tasks . . . 306
Start the Farm Configuration Wizard . . . 306
Configure the License Server . . . 307
Select Client Installation . . . 308
Connect to an Existing Farm Database . . . 308
Define and Provision the Enterprise Console . . . 309
Create a Repository DataSource . . . 310
Create a Repository . . . 312
Verify the Installation . . . 322
Intranet and Extranet Web Servers on Different Machines . . . 324
High-Level View of Server Installation . . . 324
High-Level View of Client Installation . . . 326
Installation . . . 328
AVEVA Work Tasks Server . . . 328
Install AVEVA Work Tasks . . . 328
Start the Farm Configuration Wizard . . . 328
Configure the License Server . . . 329
Select the type of installation . . . 330
Create Farm Database . . . 330
Define Security Settings . . . 333
Define Services and Global Settings . . . 335
Define Central Configuration . . . 335
Provision Services . . . 336
AVEVA Work Tasks Client . . . 337
Install AVEVA Work Tasks . . . 337
Start the Farm Configuration Wizard . . . 337
Configure the License Server . . . 338
Select Client Installation . . . 339
Connect to an Existing Farm Database . . . 339
Define and Provision the Enterprise Console . . . 340
Create a Repository DataSource . . . 341
Create a Repository . . . 343
Verify the Installation . . . 352
Install Load Balanced Server . . . 354
Web Farm with Clustered Services on Web Farm Machines with Database on a Different Machine . . . 355
High-Level View of Server Installation . . . 355
Installation . . . 357
AVEVA Work Tasks Server . . . 357

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Install AVEVA Work Tasks . . . 357
Start the Farm Configuration Wizard . . . 357
Configure the License Server . . . 358
Select the Type of Installation - Enterprise Edition . . . 359
Create Farm Database . . . 360
Define Security Settings . . . 362
Define Services and Global Settings . . . 364
Define Central Configuration . . . 365
Define Enterprise Console . . . 366
Provision Services . . . 367
Create a Repository DataSource . . . 367
Create a Repository . . . 369
Install on Multiple Machines . . . 379
Verify the Installation . . . 379
Web Farm Connecting to Separate Services Cluster with Database on a Different Machine . . . 381
High-Level View of Server Installation . . . 382
High-Level View of Client Installation . . . 384
Installation . . . 386
AVEVA Work Tasks Server . . . 386
Install AVEVA Work Tasks . . . 386
Start the Farm Configuration Wizard . . . 386
Configure the License Server . . . 387
Select the Type of Installation - Enterprise Edition . . . 388
Create Farm Database . . . 389
Define Security Settings . . . 391
Define Services and Global Settings . . . 393
Define Central Configuration . . . 394
Provision Services . . . 395
AVEVA Work Tasks Client . . . 395
Install AVEVA Work Tasks . . . 395
Start the Farm Configuration Wizard . . . 396
Configure the License Server . . . 396
Select Client Installation . . . 397
Connect to an Existing Farm Database . . . 398
Define and Provision the Enterprise Console . . . 399
Create a Repository DataSource . . . 399
Create a Repository . . . 401
Verify the Installation . . . 411
Single Server Capable of Adding Machines at Any Time . . . 413
High-Level View of AVEVA Work Tasks Installation . . . 413
Installation . . . 415
Install AVEVA Work Tasks . . . 415
Start the Farm Configuration Wizard . . . 415
Configure the License Server . . . 416
Select the Type of Installation - Enterprise Edition . . . 417
Create Farm Database . . . 417
Define Security Settings . . . 420
Define Services and Global Settings . . . 422
Provision Services . . . 422

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create a Repository DataSource . . . 422
Create a Repository . . . 425
Verify the Installation . . . 434
Intranet and Extranet Web Servers on Different Machines . . . 436
High-Level View of Server Installation . . . 436
High-Level View of Client Installation . . . 438
Installation . . . 440
AVEVA Work Tasks Server . . . 440
Install AVEVA Work Tasks . . . 440
Start the Farm Configuration Wizard . . . 440
Configure the License Server . . . 441
Select the Type of Installation - Enterprise Edition . . . 442
Create Farm Database . . . 443
Define Security Settings . . . 445
Define Services and Global Settings . . . 447
Define Central Configuration . . . 448
Provision Services . . . 449
AVEVA Work Tasks Client . . . 449
Install AVEVA Work Tasks . . . 449
Start the Farm Configuration Wizard . . . 450
Select Client Installation . . . 450
Connect to an Existing Farm Database . . . 450
Define and Provision the Enterprise Console . . . 452
Create a Repository DataSource . . . 452
Create a Repository . . . 454
Verify the Installation . . . 464
Client and Server Installation . . . 466
High-Level View of Server Installation . . . 466
High-Level View of Client Installation . . . 468
Installation . . . 470
AVEVA Work Tasks Server . . . 470
Install AVEVA Work Tasks . . . 470
Start the Farm Configuration Wizard . . . 470
Configure the License Server . . . 471
Select the Type of Installation - Enterprise Edition . . . 472
Create Farm Database . . . 473
Define Security Settings . . . 475
Define Services and Global Settings . . . 477
Define Central Configuration . . . 478
Provision Services . . . 479
AVEVA Work Tasks Client . . . 479
Install AVEVA Work Tasks . . . 479
Start the Farm Configuration Wizard . . . 479
Configure the License Server . . . 480
Select Client Installation . . . 481
Connect to an Existing Farm Database . . . 481
Define and Provision the Enterprise Console . . . 483
Create a Repository DataSource . . . 483
Create a Repository . . . 485

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Verify the Installation . . . 495
Install and Configure AVEVA Work Tasks in Silent Mode . . . 497
Prerequisites for Installing and Configuring in Silent Mode . . . 497
Install in Silent Mode . . . 497
Configure in Silent Mode . . . 498
Configure in Silent Mode Using XML . . . 498
Configure DataSource in Silent Mode Using XML . . . 531
Configure License Server Details in Silent Mode Using XML . . . 532
Configure Farm in Silent Mode Using XML . . . 533
Configure Repository in Silent Mode Using XML . . . 533
Configure Quickflow in Silent Mode Using XML . . . 534
Configure Cryptography Settings in Silent Mode using XML . . . 536
Configure in Silent Mode using Command-Line Parameters . . . 536
Configure Farm in Silent Mode Using Command-Line Parameters . . . 536
Configure DataSource in Silent Mode Using Command-Line Parameters . . . 538
Configure Repository in Silent Mode Using Command-Line Parameters . . . 539
Return Results for Silent Configuration . . . 540
Sample Batch File Script for Installing and Configuring in Silent Mode . . . 541
Limitations and Exceptions - Silent Installation and Configuration . . . 542
Installing AVEVA Work Tasks Add-in for Microsoft Outlook . . . 542
Installation Prerequisites . . . 543
Install AVEVA Work Tasks Add-in for Microsoft Office Outlook . . . 543
Uninstalling AVEVA Work Tasks Add-in for Microsoft Office Outlook . . . 543
Install Client on Windows Form Based Application . . . 544
Installation Planning Checklist . . . 545
High-Level Installation View . . . 545
Installation . . . 547
Install AVEVA Work Tasks . . . 547
Start the Farm Configuration Wizard . . . 547
Select Client Installation . . . 548
Connect to an Existing Farm Database . . . 548
Define and Provision the Enterprise Console . . . 549
Verify the Installation . . . 549
Install AVEVA Work Tasks Locales for Brazilian Portuguese . . . 551
Set a Repository to Support Multiple Locales . . . 552
AVEVA Licensing . . . 553
About AVEVA Enterprise Licensing . . . 553
How does Licensing work? . . . 554
Licensing for AVEVA Work Tasks . . . 555
Grace Period for AVEVA License . . . 556
Change Load Balance Settings . . . 557
Client Connection License for System Platform IDE . . . 557
Client Connection License for InTouch . . . 558
Session-based Client Connection License . . . 559
AVEVA Work Tasks Pro . . . 560
AVEVA License Utilization . . . 560
Push Notification Architecture for AVEVA Work Tasks Pro . . . 561
Enhanced Security . . . 562
Install AVEVA Work Tasks Pro . . . 563

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Prerequisites . . . 564
High Level View of Configuration and Installation of AVEVA Work Tasks Pro . . . 564
Access AVEVA Work Tasks Pro . . . 565
Access AVEVA Work Tasks Pro within the Corporate Network . . . 566
Access AVEVA Work Tasks Pro Outside the Corporate Network . . . 566
Install and Configure AVEVA Work Tasks Pro App . . . 567
Verify Installation . . . 569
Configure AVEVA Work Tasks Pro . . . 570
Delivery Channel - MobileWorkList . . . 570
Block or Unblock Mobile Application . . . 571
Advanced Settings . . . 572
Service Monitor . . . 572
Service Settings . . . 574
Engine Settings . . . 575
Task Scheduler Settings . . . 576
Communication Service Settings . . . 577
Client Service Settings . . . 583
Types of Restarts . . . 584
Features of Service Monitoring . . . 584
Process Memory Calculation . . . 585
Types of Monitoring . . . 586
Maintenance . . . 586
Critical Memory Limit . . . 587
Warning Memory Limit . . . 587
Service Specific Monitoring . . . 587
Workflow Engine . . . 587
Service Dependencies . . . 588
Load Balanced Environment . . . 589
Configuration Settings . . . 589
Application Constants . . . 591
Default Settings . . . 591
User Scenarios . . . 591
Technical Scenarios and Features . . . 596
Mobile Notification Service . . . 597
Register to Mobile Notification Service . . . 597
Retry Settings . . . 599
Enterprise Server Settings . . . 600
Migrating Notification Services . . . 600
Enterprise Server Settings . . . 601
Global Settings . . . 602
HTTP Settings . . . 603
Log or Transaction Settings . . . 604
File Watcher Settings . . . 606
Machine Setting . . . 607
Azure Setting . . . 607
Create Enterprise Console with Anonymous Authentication . . . 608
Configure Service Bus Listener . . . 609
Generate SAS Token . . . 610
Provide Server Details to Mobile User . . . 611

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

ArchestrA Services . . . 611
OpenID Settings . . . 613
AVEVA Connect Settings . . . 614
Azure Active Directory Settings . . . 615
Create Custom Scope in Azure Active Directory . . . 617
Retrieve Additional User Properties for Azure Active Directory User Provider . . . 618
Auto Resume Settings . . . 618
Cache Refresh Settings . . . 624
Administration Tasks . . . 625
About Forms Designer . . . 625
View Installed Products . . . 625
Manage Databases . . . 626
Change Database User Settings for Windows Authentication . . . 626
Change SQL User Credentials for SQL Authentication . . . 627
Change Pool Size of Farm Connection Settings . . . 628
Change Database Settings . . . 629
Configure Services . . . 630
Two-way Email Notification . . . 630
Change Service Settings . . . 633
Client Service Monitoring Settings . . . 633
Cryptography Settings . . . 634
Migrate Passphrase using Migration Tool . . . 636
Administer Web Servers . . . 636
Change Web Site Settings . . . 636
Set Desired Port Number for Central Configuration and Enterprise Console During Installation . . . 638
Create Website with Desired Port Number Through Farm Configuration Wizard . . . 639
Change Port Number of AVEVA Work Tasks Sites . . . 639
Change Port Number of Central Configuration and Enterprise Console . . . 640
Add a New Enterprise Console . . . 641
Map or Un-map a Repository from an Existing Enterprise Console . . . 642
Edit Client Machines . . . 643
Edit License Server Details . . . 645
Work with Enterprise Console List . . . 645
Work with Central Configuration . . . 646
Create a Repository DataSource . . . 646
Manage Repository DataSource . . . 648
Create a Repository . . . 649
Configure Administrator User . . . 658
Configure System User . . . 659
Configure Exception Mail Templates . . . 660
Change Pool Size of Repository DataSource Connection Settings . . . 661
Change Password of SQL Server User Credentials Used to Connect to Repository DataSource . . . 662
Map or Un-map a Repository from Servers and Enterprise Consoles . . . 662
Hide or Unhide Repository . . . 663
Delete Repository . . . 664
Delete Repository DataSource . . . 664
Diagnostics . . . 664
Mail Server Setting . . . 665
Configuring SMTP Settings . . . 669

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure POP3 Settings . . . 670
Configure OAuth2 to Access Google Account . . . 672
Configure OAuth2 to Access O365 Account . . . 672
View the Enterprise Servers List . . . 674
Configuring and Registering Token Service . . . 675
Enable or Disable Security for a Repository . . . 677
Enable or Disable Web Parts for a Repository . . . 678
User Providers . . . 679
Change User Provider Properties . . . 679
Add Additional User Providers . . . 679
Configure User Providers . . . 682
User List . . . 684
Manage Users in List Provider . . . 685
Working with Repository-Provider List . . . 686
Configure Provider Definition XML . . . 687
Enable an Existing Repository with Cross-domain Role Provider . . . 688
Synchronize Roles . . . 689
Delete Synchronization Cross Domain Tasks . . . 691
Synchronize User Properties with Data Source Providers . . . 691
Synchronize Active or Inactive User Properties with Data Source Providers . . . 691
Schedule Synchronization of User Properties . . . 692
Map Active Directory to SQL Provider . . . 692
Debug . . . 695
Log Files . . . 698
Exception Messages . . . 698
AVEVA Identity Manager . . . 699
Web API with AIM Authentication . . . 701
Consume Access Token from InTouch OMI App in AVEVA Work Tasks Controls . . . 701
Embedding AVEVA Work Tasks . . . 702
Add an Installation Path . . . 702
Add Registry Entries . . . 703
Entry for AVEVA Work Tasks Folders . . . 703
Entry for Farm Wizard Background Image . . . 704
Entry for Central Configuration Shortcut Path . . . 705
Customize UI Text . . . 706
Install Client Service . . . 706
Implement FarmEvents Interface . . . 706
Verify Embedded AVEVA Work Tasks . . . 708
Farm Recovery . . . 708
Move from Development Environment to Production Environment . . . 714
Reuse Repository Data . . . 716
Deployment Options . . . 718
Attach DataSource . . . 718
Steps to Perform Attach DataSource . . . 718
User Scenarios . . . 719
Deploys a Solution from One Machine to Another . . . 719
Solution Requires an Upgrade . . . 720
Disaster Recovery of a Solution . . . 720
BAM Reports from an Archived Database . . . 720

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Uninstall AVEVA Work Tasks **721**
Uninstall Client 721
Uninstall Server 721
Uninstall Farm 722
Uninstall AVEVA Work Tasks 722
Upgrading AVEVA Work Tasks **723**
Upgrade Scenarios 723
Upgrade in Standalone Environment 724
Upgrade in Client - Server Environment 724
Upgrade in Enterprise Server Environment 724
Upgrade Procedure 725
Before You Upgrade 726
Upgrading to AVEVA Work Tasks 2020 727
Upgrading to AVEVA Work Tasks 2020 Update 1 728
Upgrading to AVEVA Work Tasks 2020 Update 2 730
Upgrading to AVEVA Work Tasks 2023 731
Upgrading to AVEVA Work Tasks 2023 SP1 733
Run Upgrade Manager 734
Upgrading AVEVA Work Tasks in Silent Mode 734
Prerequisites to Upgrade AVEVA Work Tasks in Silent Mode 734
Configure License Server Details in Silent Mode Using XML 735
Upgrading in Silent Mode Using Installer 735
Upgrade in Silent Mode 736
Return Results for AVEVA Work Tasks Silent Upgrade 736
Use an Existing Website **737**
Best Practices and Guidelines **737**
Best Practices for Securing AVEVA Work Tasks 738
Best Practices for Maintaining AVEVA Work Tasks Databases 742
Best Practice for Disabling Active Directory Users 747
Best Practices for Handling Different Regional/Cultural Settings 747
Best Practice for Handling Cross Site Request Forgery in Custom Application 748
Guidelines for Sizing Database 748
Guidelines for Archiving Database 764
Archival Strategy for AVEVA Work Tasks Database 765
Archiving Repository Database 765
Archiving Farm Database 767
Archival Strategy for Custom Tables 768
Archival Strategy for Migrated Database 768
Viewing BAM Reports for the Archived Database 768
Application Scenario 769
Guidelines for Partitioning Database 772
Recommendations 773
Best Practices for Features and Services 773
Guidelines on using SQL AlwaysOn Availability Group 774
Guidelines for Configuring AVEVA Work Tasks in DMZ 774
Guidelines for Antivirus 777
Troubleshooting **778**
Prerequisites for Debugging Exceptions 779
Database Connection Errors while Installing Farm 783

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Client Service Errors . . . 786
Windows Service Errors . . . 787
Client Service Monitoring Errors . . . 787
Database Connection Errors . . . 788
IIS Error while Configuring Farm Configuration . . . 791
ASP.NET Error while Configuring Farm Configuration . . . 791
Windows Authentication or Anonymous Authentication Error . . . 792
Non-Query Execution Failure . . . 792
Not Authorized to Access Site Error . . . 792
Enterprise Console Errors . . . 793
Authentication and Personalization . . . 795
Performance Issue While Accessing Enterprise Console . . . 796
Central Configuration is not Loading . . . 797
Communication Service Errors . . . 798
Farm Configuration Error . . . 800
Workflow Test Run Errors . . . 802
Move or Copy Item Activity Errors . . . 803
Changing Resource Account Credentials . . . 804
Error while Installing Configuration Farm or Web Sites . . . 808
Error while reinstalling AVEVA Work Tasks Client . . . 809
Unable to Log on to Enterprise Console . . . 809
Server Error in Application . . . 810
Troubleshooting AVEVA Work Tasks Upgrade . . . 811
Service Bus Listener Error . . . 812
Mobile Notification Service Errors . . . 812
Error: Could not load file or assembly or Error while creating instance . . . 813
Web Part Error: Could not load the required type . . . 814
File Watcher Events Error . . . 814
Access Denied to AVEVA Work Tasks Temp Folder along with System IO Errors . . . 815
Cache Refresh not Reflected after the Product Upgrade . . . 815
Unable to Re-install AVEVA Work Tasks . . . 816
Unexpected Error when Authenticating against Azure Active Directory . . . 816
Swapping Repository Provider . . . 816

Connectors and Extensions User Guide . . . 818
Key Features . . . 818
Overview . . . 818
Connector for System Platform . . . 820
Architecture . . . 820
Installation Scenarios . . . 821
Single Computer . . . 822
GR and System Platform IDE on Computer 1, InTouch and Runtime on Computer 2, and AVEVA Work Tasks on Computer 3 . . . 822
GR and System Platform IDE on Computer 1, System Platform IDE on Computer 2, InTouch and Runtime on Computer 3, and AVEVA Work Tasks on Computer 4 . . . 823
Installation . . . 824
Install Connector for System Platform . . . 825
Install Connector for System Platform in Silent Mode . . . 827

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Verify Installation . . . 828
Upgrade Connector for System Platform . . . 829
Uninstall Connector for System Platform . . . 831
Limitations . . . 832
Extension for System Platform . . . 832
Install Extension for System Platform . . . 832
Install Extension for System Platform . . . 832
Install Extension for System Platform in Silent Mode . . . 834
Verify Installation . . . 835
Uninstall Extension for System Platform . . . 835
Upgrade Extension for System Platform . . . 836
Extension for OPC UA . . . 837
Install Extension for OPC UA . . . 837
Installation Sequence . . . 837
Install AVEVA Work Tasks . . . 838
Install Extension for OPC UA . . . 838
Provision Repository Using OPC UA Repository Provisioning Tool . . . 839
Upgrade Extension for OPC UA . . . 839
Work with Connectors for System Platform . . . 840
Overview . . . 840
Workflow Features Inside System Platform IDE and Runtime . . . 841
Overview of a Gateway Object . . . 842
Initial Gateway Setup . . . 842
AIM Authentication for accessing Enterprise Console in System Platform IDE . . . 848
Configure Gateway . . . 849
Define Workflow Events in AppObject . . . 856
Create New Workflow . . . 859
Associate an Existing Workflow to a Workflow Event . . . 859
Wait-for-AAEvent Activity in a Workflow . . . 860
AttributeReadWrite Activity . . . 861
Configure AttributeReadWrite Activity to Perform Read Operation . . . 863
Configure AttributeReadWrite Activity to Perform Write Operation . . . 864
Relative Referencing . . . 866
Wait-For-AAEvent Activity . . . 867
Workflow Features Inside InTouch . . . 869
Initial Steps . . . 869
AIM Authentication for Controls in InTouch HMI . . . 870
Inbox Inside InTouch . . . 871
Workflow Status View Inside InTouch . . . 878
BAM Inside InTouch . . . 884
Lists Inside InTouch . . . 890
Forms Inside InTouch . . . 896
Pass Contextual Information in Forms from InTouch . . . 902
List Forms Inside InTouch . . . 909
WorkItemInboxCount in InTouch . . . 918
Open Inbox using WorkItemInboxCount . . . 924
Workflow Features Inside InTouch OMI . . . 934
Initial Steps . . . 934
AIM Authentication for Controls in InTouch OMI . . . 936

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Forms Inside InTouch OMI . . . 937
Inbox Inside InTouch OMI . . . 939
List View Inside InTouch OMI . . . 940
List Form Inside InTouch OMI . . . 941
BAM Inside InTouch OMI . . . 942
Workflow Status Inside InTouch OMI . . . 944
Inbox Work Items Count Inside InTouch OMI . . . 945
Cryptography Settings Inside InTouch OMI . . . 946
Historian Server . . . 946
Trigger Workflow from the Historian Server . . . 946
Monitor Multiple Historian Event Tag . . . 950
Monitor Specific Historian Event Tag . . . 954
Retrieve Historian Data Using Database Activity . . . 958
Information Server . . . 961
Configure Repository Using AVEVA Work Tasks Connector for Information Server . . . 961
Configure Workflow Repository and Farm for Information Server . . . 961
Configure Workflow Server to Add Repository Details . . . 962
Manage Connected Workflow Repositories . . . 962
Administrative Activity . . . 962
Add Custom Panel . . . 962
Add User Permissions . . . 963
Perform Workflow Repository Functionality Within Information Server . . . 963
Workflow Farms . . . 963
Workflow Repositories . . . 963
Troubleshoot Issues . . . 963
Intelligence Data Inside Workflows . . . 964
Fetch Data from Intellience Datastore . . . 964
Configure Workflow . . . 964
Add Database Activity . . . 966
Add ForEach Loop . . . 967
Edit Scheduler Activity . . . 968
Diagnostics . . . 969
Move System Platform Galaxy and AVEVA Work Tasks Repository from Engineering Environment to Production Environment . . . 970
Move System Platform Galaxy And Repository . . . 971
Move Repository . . . 971
Move System Platform Galaxy . . . 975
Move only the Workflow . . . 976
Move a Gateway Object, Workflows, and Associated Workflow Events . . . 977
Diagnostics: Logger Integration with ArchestrA Logger . . . 978
Known Issues . . . 979
Limitations . . . 981
Troubleshoot Issues . . . 981
Work with Extension for OPC UA . . . 985
Provision Repository for OPC UA . . . 985
OPC UA Repository Provisioning Tool . . . 985
OPC UA Connections . . . 986
Activity: OPC UA Tag Read Write Activity . . . 987
OPC UA Server Connection . . . 988

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Select Attribute . . . 988
Mapped Error Output . . . 989

Developer Guide . . . 990
Get Started . . . 990
Prerequisites to Access AVEVA Work Tasks Features . . . 990
Installation and Administration . . . 990
Interfaces and Features . . . 990
Overview of User Interfaces . . . 991
Access and Manage AVEVA Work Tasks Features . . . 991
AVEVA Work Tasks Services . . . 992
Enterprise Console . . . 992
Message Queue . . . 992
Message Queue Connections . . . 993
Message Queue Repository Provisioning Tool . . . 994
MQTT Connections . . . 995
Logic App Connections . . . 998
Controls . . . 998
Add Controls in the Application . . . 999
Create an ASP.NET Project in Visual Studio .NET . . . 1000
Create User Context . . . 1003
Process Designer Control . . . 1005
Work Items List Control . . . 1008
Use Events of Work Item List Control . . . 1017
Layout Customization . . . 1019
Create Custom Link in Work Item List Control . . . 1021
Calendar Control . . . 1023
BAM Control . . . 1025
Add Individual BAM Report to aspx Page . . . 1026
View Specific Workflow Instances in BAM Control . . . 1030
Queue Control . . . 1030
List Control . . . 1031
Form List Webpart Control . . . 1032
Workflow Execution View Control . . . 1033
Configure New Repository . . . 1034
Set up Source Elements Directory . . . 1034
Set up Email Addresses for Repositories . . . 1034
Set up User Providers for Repositories . . . 1035
Set up Document Web View Control for Repositories . . . 1035
Activities . . . 1038
Start Activity . . . 1039
Start - Variables . . . 1044
Start - Contents . . . 1045
XML Variables . . . 1045
About XML Variables . . . 1045
XML Variable User Interface . . . 1046
XML Variable of Type 'XML Document' . . . 1048
Create XML Variable of Type 'XML Document' . . . 1049

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Build XML Schema . . . 1050
Use XML Document of Type 'Variables' . . . 1052
Determine XML Variable Schema from Forms . . . 1053
XML Variable for List . . . 1054
ListItem Scope of XML List of Type XML Variable . . . 1056
Schema of XML List of Type XML Variable . . . 1056
Create an XML List of Type XML Variable . . . 1057
Change Scope of the XML List of Type XML Variable . . . 1057
Set up List Query Scope . . . 1059
Create an XML List Iterator of Type XML Variable . . . 1062
Create an XML Iterator of Type XML Variable for Grid Items . . . 1064
Create Workflow for a List in a Single Step . . . 1067
Understand XML Node Mapping . . . 1070
XML List of Type XML Variable Scenarios . . . 1070
Add New List Item . . . 1070
Update the List Item that Alerted the Workflow . . . 1074
Save the List Item as Different Version as Draft/Published Using Xml Variable . . . 1078
Create New Folder Using Xml Variable . . . 1082
Add New List Item with Grid Items . . . 1084
Add New Grid Item to the Existing List Item . . . 1090
Copy List Item with Grid Items . . . 1094
Iterate through the List Item and Grid Item of the List . . . 1099
XML Variable of Type XML Iterator . . . 1101
Create XML Variable of Type XML Iterator . . . 1102
Use Variables of Type XML Iterator . . . 1103
XML Variable of Type XML Forms Data Grid . . . 1109
Create XML Variable of Type XML Forms Data Grid . . . 1110
Use XML Variable of Type XML Forms Data Grid . . . 1111
Delete XML Variables . . . 1123
Initialize Variables . . . 1124
Mapped Error Outputs . . . 1125
Human Activities . . . 1126
Approval Activity . . . 1126
Approval - Delivery Channels . . . 1140
Delivery Channels - MobileWorkList . . . 1142
Approval - Work Item Fields . . . 1142
Approval - Assign Actor(s) . . . 1143
Approval - Assign Queue(s) . . . 1147
Approval - Subject . . . 1148
Approval - Body . . . 1150
Approval - Redirect URL . . . 1151
Approval - Custom Document View URL . . . 1153
Custom Document View URL . . . 1154
Approval - HTML Template . . . 1157
Approval - Attach Reports . . . 1160
Approval - Set Calendar for Timeout . . . 1161
Approval - Multiple Timeout Warning . . . 1162
Checklist Activity . . . 1163
CheckList - Delivery Channels . . . 1179

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

CheckList - Questions . . . 1180
CheckList - Work Item Fields . . . 1183
CheckList - Assign Actor(s) . . . 1184
CheckList - Assign Queue(s) . . . 1186
CheckList - Subject . . . 1188
CheckList - Body . . . 1189
CheckList - Redirect URL . . . 1191
CheckList - Custom Document View URL . . . 1192
CheckList - Attach Reports . . . 1193
CheckList - HTML Template . . . 1195
CheckList - Set Calendar for Timeout . . . 1198
CheckList - Multiple Timeout Warning . . . 1199
Choice Activity . . . 1200
Choice - Delivery Channels . . . 1213
Choice - Action Outputs . . . 1215
Choice - Work Item Fields . . . 1216
Choice - Assign Actor(s) . . . 1217
Choice - Assign Queue(s) . . . 1221
Choice - Subject . . . 1222
Choice - Body . . . 1224
Choice - Redirect URL . . . 1225
Choice - Custom Document View URL . . . 1227
Choice - HTML Template . . . 1228
Choice - Attach Reports . . . 1231
Choice - Set Calendar for Timeout . . . 1233
Choice - Multiple Timeout Warning . . . 1234
Information Activity . . . 1235
Info - Delivery Channels . . . 1241
Delivery Channels - MobileWorkList . . . 1243
Info - Work Item Fields . . . 1243
Info - Assign Actor(s) . . . 1244
Info - Assign Queue(s) . . . 1248
Info - Subject . . . 1249
Info - Body . . . 1251
Info - Redirect URL . . . 1252
Info - Custom Document View URL . . . 1254
Info - HTML Template . . . 1255
Info - Attach Reports . . . 1258
Resource Set Activity . . . 1260
ResourceSet - Delivery Channels . . . 1271
ResourceSet - Actor Selection Filter . . . 1273
ResourceSet - Work Item Fields . . . 1277
ResourceSet - Assign Actor(s) . . . 1278
ResourceSet - Assign Queue(s) . . . 1281
ResourceSet - Subject . . . 1283
ResourceSet - Body . . . 1284
ResourceSet - Redirect URL . . . 1286
ResourceSet - Custom Document View URL . . . 1287
ResourceSet - Set Calendar for Timeout . . . 1289

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

ResourceSet - Multiple TimeOut Warning . . . 1289
Task Activity . . . 1290
Task - Delivery Channels . . . 1303
Task - Task Type . . . 1304
Task - Work Item Fields . . . 1306
Task - Assign Actor(s) . . . 1307
Task - Assign Queue(s) . . . 1311
Task - Subject . . . 1312
Task - Body . . . 1314
Task - Redirect URL . . . 1315
Task - Custom Document View URL . . . 1317
Task - Attach Reports . . . 1318
Task - HTML Template . . . 1320
Task - Set Calendar for Timeout . . . 1323
Task - Multiple Timeout Warning . . . 1324
Work Activity . . . 1325
Work - Delivery Channel . . . 1336
Work - AdvanceUI Fields . . . 1338
Work - Work Item Fields . . . 1345
Work - Assign Actor(s) . . . 1346
Work - Assign Queue(s) . . . 1349
Work - Subject . . . 1351
Work - Body . . . 1352
Work - Redirect URL . . . 1354
Work - Custom Document View URL . . . 1355
Work - Set Calendar for Timeout . . . 1357
Work - Multiple Timeout Warning . . . 1357
Invoke Form Activity . . . 1358
Invoke Form - Delivery Channels . . . 1379
Delivery Channels - MobileWorkList . . . 1380
Invoke Form - Possible Distinct Outputs . . . 1381
Invoke Form - Possible Distinct Outputs Scenario . . . 1383
Invoke Form - Work Item Fields . . . 1417
Invoke Form - Assign Actor(s) . . . 1418
Invoke Form - Assign Queue(s) . . . 1422
Invoke Form - Form Name . . . 1423
Invoke Form - Output Instance XML . . . 1424
Invoke Form - Individual Actor's Response Collaboration in Output Instance XML . . . 1426
Invoke Form - Update Form Parameters . . . 1439
Invoke Form - Subject . . . 1440
Invoke Form - Body . . . 1442
Invoke Form - Template Path . . . 1445
Invoke Form Activity Email Template . . . 1450
Use Variables in Invoke Form Activity . . . 1455
Use Contents in Invoke Form Activity . . . 1459
Use XML Variables in Invoke Form Activity . . . 1463
Invoke Form - Redirect URL . . . 1468
Invoke Form - Custom Document View URL . . . 1470
Invoke Form - Set Calendar for Timeout . . . 1472

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Invoke Form - Multiple Timeout Warning . . . 1473
Invoke Form - Mail Integration Scenario . . . 1474
Use 'Body' Property . . . 1475
Use 'Template Path' Property . . . 1481
Assign a Task to a Manager . . . 1486
Enterprise Console Control (EC Plug-in) . . . 1487
Integration Activities . . . 1491
Create-Open-Office-Document Activity . . . 1492
Create-Open-Office-Document: Destination Details . . . 1494
Create-Open-Office-Document: Select Template . . . 1495
InvokeWebService Activity . . . 1496
InvokeWebService: Web Service Details . . . 1500
Invoke Web API Activity . . . 1503
Configure Web API . . . 1505
Invoke Web API: Configure OpenAPI . . . 1507
PATCH Method . . . 1511
Create XML Variable . . . 1513
Script Activity . . . 1514
Script: Code . . . 1517
Script: References . . . 1518
Script: Return Values . . . 1519
File Operations Activity . . . 1520
File Operations: Source Details . . . 1523
File Operations: Destination Details . . . 1525
Change Workflow Status Activity . . . 1526
XML Action Activity . . . 1528
XML Action: Get Data . . . 1531
XML Action: Set Variables . . . 1533
Engine Activities . . . 1534
End Activity Execution Activity . . . 1535
End Activity Execution: Activity Details . . . 1536
End Child Workflow Execution Activity . . . 1538
End Child Workflow: Workflow Details . . . 1540
BlockingOutput Activity . . . 1541
Database Activity . . . 1542
Decision Activity . . . 1550
Decision: Decisions . . . 1551
ForEach-Loop Activity . . . 1555
Example of Serial Execution . . . 1560
For-Loop Activity . . . 1561
For - Loop Count . . . 1563
Logger Activity . . . 1564
Log - Log Message . . . 1566
Loop Execution Step Entry Activity . . . 1567
Next Loop Activity . . . 1568
Resource Variable Activity . . . 1569
Resource Variable - Actors . . . 1571
Rule Activity . . . 1575
Rule - Condition . . . 1578

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Workflow Activity . . . 1579
Workflow - Contents for ChildWorkflow . . . 1586
Workflow - Variables for ChildWorkflow . . . 1588
Workflow - XmlVariables for ChildWorkflow . . . 1589
Workflow - Return Values . . . 1589
How to Pass Variables/Contents from Parent Workflow to Child Workflow? . . . 1590
Get Execution ID of the Child Workflow . . . 1594
Update Variable Activity . . . 1594
Variables . . . 1596
Forms Xml Variables . . . 1598
Update Parent Variables Activity . . . 1602
Update Parent Variables - List Parent Variables . . . 1605
How to Update a Parent Workflow Variable from a Child Workflow? . . . 1606
Synchronize Activity . . . 1608
Set Workflow Output Variable . . . 1609
XML Navigator . . . 1615
Scheduler Activities . . . 1619
Control Custom Timer Activity . . . 1619
Set Custom Timer Activity . . . 1622
Set Custom Timer: Schedule Time . . . 1625
Scheduler Activity . . . 1626
Scheduler: Scheduler Details . . . 1629
Reschedule Failed Tasks . . . 1631
Time Trigger Activity . . . 1632
Time Trigger: Timer Trigger . . . 1634
Communication Activities . . . 1635
Email Activity . . . 1635
Email: ToResource . . . 1643
Email: BCC Resource . . . 1646
Email: CCResource . . . 1651
Email: Subject . . . 1656
Email: Body . . . 1657
Email: Attach Reports . . . 1659
Configure Communication Activities . . . 1660
Prerequisites . . . 1660
Configure Farm Configuration Wizard for Receiving Emails . . . 1661
Configure Farm Configuration Wizard for Receiving SMS . . . 1661
Configure Farm Configuration Wizard for Sending SMS . . . 1662
SendSMS and WaitForSMS . . . 1666
Disable Reply Email for Invalid Emails . . . 1669
Notification Activity . . . 1669
Notification: ToResource . . . 1673
Notification: ToResource . . . 1676
Send SMS Activity . . . 1678
Send SMS: RecipientResource . . . 1681
Send SMS: Message . . . 1685
Wait for Email Activity . . . 1686
Wait for Email: Filter Expression . . . 1690
Wait for Email: Set Variables And Contents from Email . . . 1691

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Wait for Email: Attachment Save Options . . . 1693
Wait for Email: Attachment Save Location . . . 1694
Wait for Email: Set Variables And Contents from Email . . . 1695
Wait for SMS Activity . . . 1695
Wait for SMS: Filter Expression . . . 1700
Wait for SMS: Set Variables and Contents from SMS . . . 1701
Wait for SMS: Set Calendar for Timeout . . . 1702
AVEVA Work Tasks Pro Push Notification Activity . . . 1703
Configure Recipient Resource . . . 1704
Set Notification Type . . . 1707
AVEVA Work Tasks Pro Push Notification Activity Scenarios . . . 1708
Configure Notification Message . . . 1709
BPMN Elements . . . 1709
Gateway Placeholder . . . 1710
Group Box . . . 1710
Intermediate Event Placeholder . . . 1712
Lane . . . 1713
Placeholder . . . 1714
Placeholder: Action Outputs . . . 1715
Pool . . . 1716
List Activities . . . 1717
List Navigator Activity . . . 1717
XML List Navigator Activity - XMLVariable of List Type . . . 1720
List Navigator: List Name or ID . . . 1721
List Navigator: List Query . . . 1723
List Navigator: Query Builder . . . 1724
Wait for Event Activity . . . 1727
Wait for Event: List ID . . . 1731
Wait for Event: No of Events . . . 1733
Wait for Event: Effective From . . . 1734
Wait for Event: Effective To . . . 1735
Wait for Event: Workflow Event Data XML Variable . . . 1737
Security Activities . . . 1738
Cancel Approval Workflow . . . 1738
Security Instruction Handler . . . 1740
Report Activities . . . 1742
Report Generator Activity . . . 1742
Configure Report Template . . . 1746
Create a Historian Connection . . . 1751
SOA Activities . . . 1752
SOA Execute Activity . . . 1752
Execute Methods with Out Parameters in SOA . . . 1752
Retrieve Multiple Records from Database Using SOA . . . 1761
How to Iterate Collection Object? . . . 1769
How to Design a Workflow? . . . 1775
ArchestrA Events . . . 1776
ArchestrA Wait for Event . . . 1776
Set ArchestrA Event Filters . . . 1779
Message Queue Activities . . . 1781

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SendMQData . . . 1781
Configuring Message Queue Connection . . . 1782
WaitForMQData . . . 1783
Set Message Queue Filters . . . 1784
Logic App Activities . . . 1786
Call Logic App Activity . . . 1786
Configuring Logic App Connection and Data . . . 1787
Wait for Logic App Alert Activity . . . 1789
Design and Execute Workflows . . . 1790
Create a New Workflow Process . . . 1791
Execute the Workflow . . . 1791
List of Internal Status Codes . . . 1791
Execute Workflows . . . 1793
Execute Workflows Asynchronously . . . 1793
Use Redirection Enabled Property for Synchronous Execution . . . 1794
Enable Human-Workflow Interaction . . . 1797
Human Workflow Services . . . 1797
About Channels . . . 1797
About Forms . . . 1798
Create Channels and Forms . . . 1798
Use Channels and Forms . . . 1798
Run-time Behavior of Channels and Forms . . . 1798
Set Access Rights for Form Instances . . . 1799
Use ASPX Form as the Detailed View Page of Work Item List Control . . . 1802
Email and Microsoft Office Outlook Integration . . . 1804
Introduction to Sending Activities by Email . . . 1804
Configure Activities to be Sent by Email . . . 1805
Date Format in Email Template . . . 1805
Integration with Microsoft Office Outlook . . . 1806
Respond to Approval Activities from Email . . . 1806
Respond to Checklist Activities from Email . . . 1807
Respond to Choice Activities from Email . . . 1808
View Information Activities from Email . . . 1810
Respond to Task Activities from Email . . . 1810
Configure Repository Database to View Process Execution View, Document View, and Work Items View from Email . . . 1812
Support Variables, Contents, and XML Variables in Email Templates . . . 1813
Approval Activity Email Template . . . 1814
Use Variables in the Approval Email Template . . . 1819
Use Contents in the Approval Email Template . . . 1828
Use XML Variables in the Approval Email Template . . . 1833
Choice Activity Email Template . . . 1838
Use Variables in Choice Email Template . . . 1843
Use Contents in Choice Email Template . . . 1847
Use XML Variables in Choice Email Template . . . 1852
Check-List Activity Email Template . . . 1857
Use Variables in Check-List Email Template . . . 1859
Use Contents in Check-List Email Template . . . 1864
Use XML Variables in Check-List Email Template . . . 1868

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task Activity Email Template . . . 1874
Use Variables in Task Email Template . . . 1879
Use Contents in Task Email Template . . . 1884
Use XML Variables in Task Email Template . . . 1889
Information Activity Email Template . . . 1895
Use Variables in Information Email Template . . . 1896
Use Contents in Information Email Template . . . 1899
Use XML Variables in Information Email Template . . . 1901
Different Email Template for Different Activity in a Workflow . . . 1905
Use 'HTML Template' Property of Approval Activity . . . 1906
Use 'HTML Template' Property of Choice Activity . . . 1915
Use 'HTML Template' Property of Checklist Activity . . . 1923
Use 'HTML Template' Property of Task Activity . . . 1930
Use 'HTML Template' Property of Information Activity . . . 1938
Add-in for Microsoft Office Outlook . . . 1942
Context Menu Options . . . 1942
Approval Activity . . . 1942
Checklist Activity . . . 1942
Choice Activity . . . 1943
Task Activity . . . 1943
Microsoft Office Outlook Follow up . . . 1943
Ribbon Bar Options . . . 1944
Database . . . 1944
Use Database Scripts to create Database Tables . . . 1944
Advanced Features . . . 1945
Events Framework . . . 1945
Event Framework Components . . . 1947
Events Framework Interfaces and Classes . . . 1949
Event Provider Architecture . . . 1950
Inbound Event Provider . . . 1951
Outbound Event Provider . . . 1951
Link Event Binding and Provider . . . 1952
Use Default Event Providers . . . 1952
Add a Custom Event Provider . . . 1953
Steps to Create Custom Event Provider . . . 1953
Code Sample - Class Library for Custom Event Provider . . . 1958
Code Sample - Files for File Watcher Event Binding Page . . . 1965
Create File Watcher Initiated Workflow or Custom Workflow . . . 1979
Code Sample - File Watcher Initiated Workflow . . . 1983
Import BPMN Process Models from Microsoft Visio . . . 2005
Overview of Visio Process Import . . . 2005
Map BPMN Events to AVEVA Work Tasks Actions . . . 2006
BPMN End Events . . . 2007
End Event . . . 2007
Raise Exception . . . 2007
Cancel Transaction . . . 2008
Compensate Transaction . . . 2009
Terminate Workflow . . . 2009
Link Source . . . 2010

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

End With Message . . . 2010
Multiple (Not Mapped) . . . 2011
BPMN Intermediate Events . . . 2011
Intermediate Events Overview . . . 2011
On Exception . . . 2011
Wait for Event . . . 2012
Wait Timer . . . 2012
On Transaction Canceled . . . 2013
On Compensation . . . 2013
Map BPMN Activities to AVEVA Work Tasks Activities . . . 2014
BPMN Task . . . 2014
Sub-Process . . . 2015
Map Connectors . . . 2016
Sequence Flows . . . 2016
Message Logs . . . 2016
Set Message Log Debug Mode Level . . . 2016
Log Message Categories . . . 2017
View Message Log Files . . . 2017
Logger Console . . . 2018
Option to Disable Logging of Strings . . . 2018
Manage and Synchronize Threads in Engine . . . 2018
Limit In-Memory Activity/Steps Count for Repository . . . 2019
Thread Pools at Repository Level . . . 2020
Activity Concurrency Limit . . . 2020
Entities . . . 2021
Basic Understanding of the Need for Data Bridge Interface . . . 2022
DataBridge Provider . . . 2022
User Entity . . . 2023
XML Schema for User Entity and User Entity Data Source Provider . . . 2024
Configure Forms Provider for Authentication . . . 2030
Map Active Directory and SQL Provider . . . 2032
Role Entity . . . 2035
XML Schema Used by Role Entity Data Source Provider . . . 2035
Synchronizing Roles of Queue Participants . . . 2036
Entity Data Source Provider Code Samples . . . 2037
Create User Entity Data Source Provider . . . 2037
Create User and Role Entity Data Source Provider . . . 2053
Step 1: Create Data Source for the Repository . . . 2054
Step 2: Create a Repository and Map the Data Source to the Repository . . . 2055
Step 3: Create Custom User Entity Data Source Provider for Authentication and User Information . . . 2056
Step 4: Create Default Admin for the Repository . . . 2073
Step 5: Create Custom Role Entity Data Source Provider . . . 2077
Custom Role Provider Fetching Roles from SQL and Users from Active Directory . . . 2088
Associate Workflow to Scheduler Event . . . 2104
Set up a Schedule . . . 2104
Edit Scheduler Association . . . 2105
Example of Scheduling Tasks Using Scheduler Event . . . 2106
Manage Work Items . . . 2106

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Forward Work Items to Other Users . . . 2106
Abort Work Items . . . 2107
Add New Tab and Custom Logic to Process Execution View . . . 2107
Customize the 'Step Details' Window . . . 2109
Customization . . . 2111
Customize User Interfaces . . . 2111
Style Sheets . . . 2111
Customize Farm Configuration Wizard User Interface . . . 2113
Customize Enterprise Console Menu . . . 2113
Customize Fonts . . . 2114
Customize Work Item Control . . . 2114
Customize Left Menu Grid . . . 2115
Customize Grid Details View . . . 2115
Change Height and Width of the Grid Details View . . . 2117
Change Font Style Based on the Status of the Work Item . . . 2117
Add or Remove Images . . . 2118
Customize User Name . . . 2119
Customize Document View . . . 2119
Themes . . . 2120
Apply Theme to the Core Product . . . 2120
Themes in Work Item Control . . . 2121
Apply Themes in Grid Left Menu . . . 2121
Apply Themes in Grid Details View . . . 2121
Themes in Calendar Control . . . 2122
Themes in Queue Control . . . 2122
Themes in Process Designer Control . . . 2123
Change Style of New Menu in Enterprise Console . . . 2124
Change Style of Action Bar in Lists . . . 2125
Change Style of Window . . . 2125
Change Style of Buttons . . . 2125
Content Handler . . . 2125
Use Content Handler . . . 2126
XML to Import WF which Implements Custom Content Handler . . . 2128
Use Custom Content Handler . . . 2138
Code Sample - Custom Content Handler Class File . . . 2138
Persist Logged Exceptions to Database . . . 2150
Enable Exception Persistence . . . 2150
Different Views for Viewing the Exceptions . . . 2151
Workflow Level . . . 2151
Instance Level Exception . . . 2151
Workflow Action Step Level . . . 2151
AVEVA Work Tasks Pro Application . . . 2152
Custom Conflict Management . . . 2152
Providers for Controls . . . 2153
Option to Disable Logging of Strings . . . 2155
Use Local Variables . . . 2156
Set Custom Properties for Repository Data Source . . . 2157
Enable Xcopy Deployment . . . 2158
Human Activity - Enhancement for Escalation and Time out . . . 2159

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Modify the Number of Work Item Fields Limit Per Data Type . . . 2159
Manager Lookup for Users List . . . 2160
Redirect the Log On Page . . . 2160
About Role Lookup . . . 2161
Use Work Activity . . . 2161
Auto Mapping of Active Directory Providers . . . 2168
Customize the Log On Page . . . 2169
Packages . . . 2171
About Package . . . 2171
Package Template . . . 2172
Manage Package . . . 2172
Best Practices of Packaging . . . 2173
Work with Package Template . . . 2175
Create a Package Template . . . 2177
Edit a Package Template . . . 2178
Configure Package Template . . . 2178
Package Template Level Configuration Settings . . . 2179
Prerequisites . . . 2179
Runtime Initialization Settings . . . 2180
Exceptions . . . 2183
Additional Package Files . . . 2183
Configure Settings for Installer Files . . . 2184
Additional Settings . . . 2186
Validate Connections . . . 2186
Validate SQL Scripts . . . 2187
Execute SQL Scripts . . . 2188
Create Package Template . . . 2189
Artifact Level Configuration Settings . . . 2190
Configure the Artifacts . . . 2190
Rebuild the Artifacts . . . 2193
Delete the Artifacts . . . 2195
Save Package Template . . . 2195
Publish a Package Template . . . 2196
Unpublish a Package Template . . . 2197
Generate a Package Template . . . 2197
Export a Package Template . . . 2197
Mark Package Template as Shipped . . . 2198
Include Associated and Child Artifacts to a List Item . . . 2199
Include Security Artifacts . . . 2202
Include Automatically Generated SQL Script . . . 2203
Manage Package . . . 2204
Import a Package . . . 2205
Manage Versions of a Package . . . 2206
Store Package Information for Lists . . . 2207
Is Artifact Modified Column for Workflow . . . 2210
Edit a Package . . . 2211
View Prerequisites of a Package . . . 2211
Install a Package . . . 2211
Configure the Package . . . 2213

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Package Installation - Prerequisites . . . 2213
Runtime Initialization Values . . . 2214
Additional Package Files . . . 2214
View Validation Log . . . 2214
View Package History . . . 2215
Upgrade Scenarios . . . 2216
Uninstall a Package . . . 2216
Delete a Package . . . 2218
Migration of Packages . . . 2218
Remotely Managing Packages using Console Application . . . 2218
Remotely Export, Import, Install, or Uninstall Packages using Console Application . . . 2219
Configure WWSKBPMRemotePackageManager Using XML . . . 2220
Import, Install, or Uninstall Packages using Console Application . . . 2224
Configure WWSKBPMPackageManager Using XML . . . 2225
Add Artifacts to a Package Template . . . 2226
ArchestrA Events . . . 2228
BAM . . . 2231
Calendar . . . 2233
Database Connection . . . 2236
Event Associations . . . 2242
Event Associations of Type Scheduler . . . 2242
Event Associations of Type Enterprise Console . . . 2246
Event Associations of Type File Watcher . . . 2251
Event Associations of Type System Platform Events . . . 2256
Event Associations of Type Logic App . . . 2260
Event Associations of Type Message Queue . . . 2264
Event Association of Type BAM . . . 2267
File Watcher . . . 2270
Resolve Physical Path of File Watcher . . . 2273
Folder . . . 2273
Child Folder . . . 2275
Forms . . . 2276
Forms . . . 2276
Resolve Action Bar Settings . . . 2283
Resolve Print Forms Details . . . 2284
Resolve Navigate Form Details . . . 2284
Configure Artifact Type Settings for Forms . . . 2285
Resolve Custom Dependent Files for Forms . . . 2286
Resolve Invoke Workflow . . . 2286
Resolve Image . . . 2287
Resolve Hyperlink . . . 2288
Forms Association . . . 2288
Form Controls . . . 2290
Resolve Custom Dependent Files for Form Controls . . . 2293
Global Work Item Layout . . . 2294
Grid Configuration . . . 2296
Logic App Connections . . . 2299
Lookup Settings . . . 2302
Lookup Example Scenarios . . . 2306

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Manage List . . . 2307
Manage View . . . 2313
Message Queue Connections . . . 2317
MQTT Connections . . . 2320
Navigation List . . . 2324
    Navigation List Folders . . . 2327
    Resolve Custom Dependent Files for Navigation Links . . . 2330
Queue Management . . . 2330
    Resolve Associated Workflows for Queue . . . 2333
    Resolve Custom Dependent Files for Queue . . . 2334
Reports . . . 2334
Report Connections . . . 2337
Report Templates . . . 2341
    Resolve Report Template Path . . . 2344
Security Group . . . 2345
    Security Group - Repository Level . . . 2346
    Security Group - List Level . . . 2351
    Security Group - List Item Level . . . 2355
    Security Group with Forms as Associated Artifacts . . . 2359
    Security Group with Workflow as Associated Artifacts . . . 2360
Security Right . . . 2360
SOA Assembly . . . 2363
    Resolve Custom Dependent Files for SOA Assembly . . . 2367
SOA Folder . . . 2368
Web API . . . 2371
Web API Source . . . 2374
Workflows . . . 2376
    Configure Artifact Type Settings for Workflow . . . 2384
    Resolve Workflow Activities . . . 2385
        Map the Version of the Associated Workflow to the Parent Workflow . . . 2389
        Map the Version of the Associated Form . . . 2390
        Resolve Web Service Configurations . . . 2391
        Resolve File Details . . . 2392
        Resolve Custom Document View URL . . . 2393
        Resolve Redirect URL . . . 2393
        Resolve Template Path . . . 2394
        Resolve Workflows with Database Activity . . . 2394
        Resolve Script Activity References . . . 2396
            Resolve Custom Dependent Files for Script Activity . . . 2397
        Resolve User or Role Based Exception Template . . . 2397
        Resolve Delivery Channels . . . 2397
        Resolve XML Variables . . . 2398
        Resolve Custom Activities . . . 2398
            Correct Custom Activities . . . 2399
        Resolve MES XML Command Activity . . . 2402
        Resolve Wait for MES Event Activity . . . 2402
        Resolve Start MES Transaction Activity . . . 2403
        Resolve Commit MES Transaction Activity . . . 2403
        Resolve Rollback MES Transaction Activity . . . 2404

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Resolve ArchestrA Activities . . . 2404
Resolve Wait For Logic App Alert . . . 2406
Resolve Call Logic App Connection . . . 2406
Work Item Field . . . 2407
Include Child Item . . . 2410
File Distribution . . . 2410
SQL Script Generation and Execution . . . 2411
Package in a Distributed Environment . . . 2412
Troubleshoot Packages . . . 2412
Event Association with Enterprise Console at List item Level . . . 2414
Packageable Artifacts . . . 2415
Localization . . . 2416
Scope of Localization . . . 2417
Steps to Localize AVEVA Work Tasks . . . 2417
Steps to Localize Grid Control . . . 2421
Steps to Localize Editor Control in WorkflowMemo and Wizard . . . 2422
Steps to Localize the Escalation Email Wizard . . . 2422
Steps to Localize Repository Name in Enterprise Console . . . 2422
Steps to Localize List Names . . . 2423
Steps to Localize Forms . . . 2423
Steps to Localize Default Properties of Web Part Editor . . . 2423
Steps to Localize Provider in Role Lookup User Lookup and Assign Actor Property . . . 2425
Non-supported Localization Tags . . . 2426
Localization Folders . . . 2427
Locales Folder . . . 2427
Workflow Elements Folder . . . 2427
List of Keys to be Localized for Work Item List Control . . . 2427
Keys for Localizing WebGrid Control . . . 2429
Keys for Localizing List Names . . . 2431
Localize the Response Sent for an Invalid Email to the Communication Engine . . . 2433
Code Samples . . . 2433
Restart Services . . . 2433
Execute Workflow . . . 2435
Custom Activity . . . 2435
Configure Custom Activity without any Properties . . . 2435
Custom Approval Activity . . . 2436
Custom Choice Activity . . . 2449
Implementation of Custom WorkItemForm . . . 2464
Configure Custom ASPX Property . . . 2471
Custom Property . . . 2483
Custom Engine Activity . . . 2495
Configure ASPX Form as Work Item Detail View Page . . . 2501
Get Client Credential Token . . . 2507
IPropertyRoutingCondition . . . 2508
Lists . . . 2508
Workflow List . . . 2508
Add Workflow Item to Workflow List . . . 2508
Get All the Workflows Present in the Selected Repository . . . 2509
Get All the Versions of a Workflow . . . 2511

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Get All the Published Versions of a Workflow . . . 2512
Add Folder Item to Workflow List . . . 2513
Edit Existing Folder Item in Workflow List . . . 2514
Import the Workflow from Workflow Definition Xml . . . 2515
Create New Version . . . 2516
Create ListItem . . . 2517
Save ListItem . . . 2518
Data Source . . . 2520
Implement IDBConnectionString Interface . . . 2520
Create Data Source Using ConnectionString . . . 2521
Create Data Source Using Assembly File . . . 2523
Subscribe AssemblyResolve Event while Using APIs for Database-related Activities . . . 2526
XML Variable API . . . 2528
Create Web Service Using Windows Application . . . 2532
SOA . . . 2536
How to Use SOA in a Simple Method? . . . 2537
How to Use a Web Service or Remote Service using SOA Infrastructure? . . . 2541
How to Use an Application that Interacts with the SQL Server? . . . 2546
How to Use Static Members of the Class? . . . 2549
How to Pass the Object as Parameter to the Constructor? . . . 2550
How to Use Scripting in Expression Editor? . . . 2552
How to Use Expression in Constructor Parameter? . . . 2553
SOA Object Support . . . 2554
Features Supported for SOA . . . 2554
Scenario . . . 2554
How to Create a Class for Handling the 'Product' Object? . . . 2555
How to Create an Object Instance? . . . 2556
How to Create a Variable? . . . 2559
How To . . . **2560**
Add Users to Active Directory . . . 2560
Create Custom Wizards . . . 2561
Return Values from Engine to Client . . . 2574
Implement Interface to Get Action and Workflow Status Change Information from Engine . . . 2577
Get User Name with Domain Name . . . 2579
Best Practices . . . **2580**
Database Tips . . . 2581
Coding Tips . . . 2582
Performance Tips . . . 2587
Troubleshoot Issues . . . 2587
Limitations . . . **2590**

Forms and Lists User Guide . . . 2591

Features available in Forms and Lists . . . **2591**
Forms . . . **2592**
Introduction . . . 2592
Get Started with Forms . . . 2593
Access Forms . . . 2595
Create and Organize Forms . . . 2601

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create Forms . . . 2601
Create Folders for Forms . . . 2602
View Child Items of Forms . . . 2602
Organize Forms . . . 2602
Import and Save Forms . . . 2603
Import Forms . . . 2603
Export Forms . . . 2604
View Versions of Forms . . . 2604
Design and Publish Forms . . . 2604
Design Forms . . . 2605
Edit Forms . . . 2605
Delete Forms . . . 2605
Preview Forms . . . 2606
Publish Forms . . . 2606
Unpublish Forms . . . 2607
Publish Folders . . . 2607
Unpublish Folders . . . 2608
Manage and Secure Forms . . . 2608
Lock Forms . . . 2608
Unlock Forms . . . 2608
Disable Forms . . . 2609
Enable Forms . . . 2609
Filter Forms . . . 2610
Change Ownership of Forms . . . 2611
Set Security of Forms . . . 2612
View Properties of Forms . . . 2613
View Audit History of Forms . . . 2614
Create Forms Links . . . 2614
Dynamic List Forms . . . 2617
Create Dynamic List Forms . . . 2617
Validate Dynamic List Forms . . . 2618
Associate Forms to Workflows . . . 2619
Associate Forms to Workflows . . . 2619
Bind a Form to the Workflow . . . 2620
Form XML Mapping . . . 2621
Edit Association of the Already Associated Form . . . 2621
Edit Mappings of the Already Associated Form . . . 2622
View Association of Forms with Workflows . . . 2622
Dissociate Forms from Workflows . . . 2623
Design Workflows for Associated Forms . . . 2623
Forms Designer . . . 2624
Manage Forms . . . 2630
Edit Properties of Forms . . . 2630
Preview Forms . . . 2631
Publish Forms . . . 2631
List Structure . . . 2631
Manage Controls . . . 2632
Add Controls . . . 2632
Move Controls . . . 2632

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Clone Controls . . . 2633
Delete Controls . . . 2633
Clear Controls . . . 2633
Replace Controls . . . 2634
Edit Name of Controls . . . 2634
Edit Properties of Controls . . . 2635
Design Guidelines . . . 2635
Custom Styles . . . 2635
Arc . . . 2636
Button . . . 2637
Card . . . 2638
Gantt Chart . . . 2639
Gauge . . . 2640
Form . . . 2640
Base Form . . . 2642
Panel Form . . . 2643
Embed Form . . . 2644
Progress Bar . . . 2645
Tabs . . . 2646
Other Controls . . . 2648
Boundaries and Limits . . . 2649
Test Environment . . . 2649
Boundaries and Limits by Property . . . 2650
Control Classification . . . 2651
Boundaries and Limits by Control . . . 2651
Boundaries and Limits by Form . . . 2653
Best Practices . . . 2654
Form Controls . . . 2655
Containers . . . 2655
Form . . . 2655
Action Bar . . . 2661
MQTT Topic Configuration . . . 2667
Data Types in MQTT Topic Configuration . . . 2669
Specify JSON Path . . . 2671
Base Form . . . 2672
Panel Form . . . 2679
Tabs . . . 2684
Tab . . . 2688
Controls . . . 2690
Arc . . . 2690
Example Arc Rendering . . . 2697
Attachment . . . 2698
Boolean . . . 2708
Button . . . 2711
Card . . . 2720
Chart . . . 2728
Lookup Source Configuration for Sankey Chart . . . 2732
Check Box . . . 2733
Data Grid . . . 2738

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure Grids . . . 2751
Update Grid Configuration for Lookup Modification . . . 2753
Highlight Records . . . 2753
Data Lookup . . . 2756
Date . . . 2763
Date Time . . . 2768
Drop-Down . . . 2773
Email . . . 2777
Embed Form . . . 2782
Embed Page . . . 2787
Set Embed Page Url and Trigger the On Data Change Script . . . 2790
E-signature . . . 2791
Gantt Chart . . . 2795
Example Gantt Chart Rendering . . . 2802
Gauge . . . 2802
Geolocation . . . 2807
Hidden Field . . . 2812
Hierarchical List . . . 2813
Hyperlink . . . 2816
Image . . . 2820
Invoke Workflow . . . 2825
Overwrite Workflows . . . 2836
Overwrite XML Variables . . . 2837
Label . . . 2838
List . . . 2842
Number . . . 2848
Password . . . 2853
Pop-Up . . . 2858
Progress Bar . . . 2865
Example Scripting Scenarios and Rendering . . . 2871
Example Progress Bar Rendering . . . 2871
QR Code . . . 2873
Radio Button . . . 2879
Report Viewer . . . 2884
Report Viewer Configuration . . . 2887
Embed Reports . . . 2891
Link Pop-up Reports . . . 2892
Link Downloadable Reports . . . 2892
RFID . . . 2893
Rich Text . . . 2898
Role Lookup . . . 2903
Signature . . . 2907
Slider . . . 2911
Text . . . 2914
Time . . . 2919
Tree . . . 2923
User Lookup . . . 2928
Widget . . . 2933
Dynamic Widget Properties and Events . . . 2938

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Add or Edit a Dynamic Property or Event . . . 2939
Grid Widget . . . 2940
Grid Widget Properties Usage . . . 2941
Grid Widget Event . . . 2944
On Cell Change . . . 2944
On Initialize . . . 2945
On Row Selection . . . 2945
Configure the Grid Widget . . . 2945
Pass Data to or from the Widget . . . 2945
Define Data Columns . . . 2946
Clear Data Filters . . . 2947
Reset Grid Data Sort Order Setting . . . 2947
Validate Data . . . 2948
Grid Widget Configurable Parameters . . . 2949
Modify Grid Widget Title . . . 2949
Modify Column Width and Column Headers . . . 2950
Set Number of Records to Display in a Grid Page . . . 2950
Configure Cell Datatypes and Input Field . . . 2951
Configure Column Background Color . . . 2952
Add Column-value Tooltip . . . 2952
Grid widget UI controls . . . 2952
Add Progress Bar in Grid Column . . . 2953
Add Datetime Picker . . . 2953
Add Drop-down List . . . 2954
Sort, Filter, and Reorder . . . 2955
Change Default Sorting Behavior . . . 2955
Filter Data through Scripting . . . 2955
Column Data Alignment . . . 2955
Hierarchy Selector Widget . . . 2956
Hierarchy Selector Widget Properties Usage . . . 2957
Hierarchy Selector Events . . . 2959
On Initialize . . . 2960
On Selection Change . . . 2960
Configure Hierarchy Selector Widget . . . 2961
Pass Data to and from Widget . . . 2961
Load Hierarchical Dataset . . . 2962
Override the Hierarchy Selector's Default Behavior for Selected Child Node . . . 2964
Modify Widget Title, Header, and Captions . . . 2964
Set Color for Parent Node . . . 2966
Set Color for Child Node . . . 2966
Localize Widgets . . . 2966
Example localization text file . . . 2967
Configure Communication between Widget Control and Form . . . 2968
Custom Controls . . . 2969
Create Custom Controls . . . 2970
Extend Existing Controls . . . 2972
Custom Properties . . . 2974
Attributing Properties . . . 2974

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Index . . . 2975
Create Custom Properties . . . 2977
IControlProperty . . . 2978
Custom Bindings . . . 2980
IBindingProperty . . . 2982
Rules . . . 2982
Define Rules . . . 2983
Reference External CSS and JS Files . . . 2984
Localization . . . 2985
Custom Categories . . . 2985
Create Custom Categories . . . 2986
Add Custom Controls in Custom Categories . . . 2987
Add Custom Controls in the Controls Category . . . 2987
Organize Form Controls . . . 2988
Example: Masked Input Control . . . 2988
SFControl.cs . . . 2989
MaskedInput.cs . . . 2989
skelta.maskedinput.js . . . 2992
Add Masked Input Property for Check Syntax Validation . . . 2993
Custom Controls with Kendo UI . . . 2993
Create Custom Controls with Kendo UI . . . 2994
Create Knockout Bindings for Kendo UI . . . 2996
Knockout Bindings for Kendo UI . . . 2997
Example: Barchart Control . . . 3015
SFControl.cs . . . 3015
BarChart.cs . . . 3016
BarChartBindingProperty.cs . . . 3019
knockout-kendo.kendochart.js . . . 3019
corp.customcontrol.barchart.js . . . 3020
Add Barchart Properties for Check Syntax Validation . . . 3022
Example: RichTextBox Control . . . 3023
SFControl.cs . . . 3023
RichTextBox.cs . . . 3024
knockout-kendo.editor.js . . . 3027
richtextbox.js . . . 3028
Custom Controls with Angular 12 . . . 3029
Create Custom Controls with Angular 12 . . . 3029
Custom Controls with Custom Properties . . . 3032
Example: PropertyLookup Control . . . 3034
SFControl.cs . . . 3034
PropertyLookup.cs . . . 3034
PropertyLookup.js . . . 3037
PropertyLookup.css . . . 3038
Example: Autocomplete Control . . . 3038
SFControl.cs . . . 3038
Autocomplete.cs . . . 3039
GridProperty.cs . . . 3044
PopupProperty.cs . . . 3047
Autocomplete.js . . . 3048

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

knockout-kendo.autocompletebinding.js . . . 3049
BaseFormWithValidationScript.xml . . . 3050
PropertyLookup.htm . . . 3050
Custom Control with Standard Properties . . . 3052
Example: TreeView Control . . . 3053
SFControl.cs . . . 3054
TreeView.cs . . . 3054
TreeViewBinding.cs . . . 3056
corp.customControls.treeView.js . . . 3057
kendo.treeview.js . . . 3058
knockout-kendo.treeview.js . . . 3089
Add Custom Control Members as White List for Check syntax validations . . . 3089
Default Values . . . 3090
Validation of Values . . . 3091
Form Layouts . . . 3092
Design Form Layouts . . . 3092
Example: Sale Registration Form . . . 3093
Forms in Custom Pages . . . 3097
Load Single Form . . . 3097
Single Form Custom Page Template . . . 3098
Localization . . . 3101
Localize Forms . . . 3101
Localize Control Names and Descriptions . . . 3102
Localize Standard Validation Messages . . . 3103
Localize Messages in Validation Scripts . . . 3103
Localize Column Names in Data Grid . . . 3103
Number and Date Formats . . . 3104
Number Formats . . . 3104
Date and Time Formats . . . 3106
Form Scripts . . . 3108
Framework . . . 3108
JavaScript . . . 3109
Script Editor . . . 3109
Validate Scripts . . . 3111
SOA through Scripts . . . 3112
Guidelines . . . 3112
Properties and Events . . . 3112
Enable . . . 3112
Visible . . . 3113
Read-Only . . . 3114
Mandatory . . . 3114
Validate . . . 3115
Options . . . 3116
Behavior . . . 3118
Extended Information . . . 3119
On Click . . . 3121
On Data Bound . . . 3121
On Data Change . . . 3122
On Form Load . . . 3124

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

On Record Add . . . 3124
On Record Delete . . . 3125
On Selection Change . . . 3125
On Tab Index Change . . . 3126
On Check Change . . . 3126
On Initialize . . . 3127
Record Visible . . . 3127
Range Color . . . 3128
Containers . . . 3129
Base Form . . . 3129
Panel Form . . . 3132
Action Bar . . . 3134
Tabs . . . 3134
Tab . . . 3135
Controls . . . 3137
Arc . . . 3137
Attachment . . . 3140
Boolean . . . 3144
Button . . . 3146
Card . . . 3149
Example Scripting Scenarios and Rendering . . . 3152
Scripting Scenarios . . . 3152
Example Card Control Rendering . . . 3152
Chart . . . 3154
Check Box . . . 3156
Data Grid . . . 3162
Data Lookup . . . 3167
Date . . . 3170
Date Time . . . 3174
Drop-Down . . . 3178
Email . . . 3183
Embed Form . . . 3186
Embed Page . . . 3188
E-signature . . . 3190
Gantt Chart . . . 3192
Gauge . . . 3196
Geolocation . . . 3200
Hidden Field . . . 3202
Hierarchical List . . . 3203
Hyperlink . . . 3211
Image . . . 3213
Invoke Workflow . . . 3215
Label . . . 3219
List . . . 3221
Number . . . 3225
Password . . . 3229
Pop-Up . . . 3232
Progress Bar . . . 3235
Radio Button . . . 3238

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Report Viewer . . . 3242
Rich Text . . . 3243
Role Lookup . . . 3245
Signature . . . 3248
Slider . . . 3251
Text . . . 3252
Time . . . 3256
Tree . . . 3259
User Lookup . . . 3261
Widget . . . 3265
Forms Parameters . . . 3268
Add Forms Parameters . . . 3268
Delete Forms Parameters . . . 3269
Set Forms Parameters . . . 3269
Forms Parameters for Invoke Form Activity . . . 3270
Best Practices . . . 3271
APIs . . . 3278
Forms APIs . . . 3278
SFU.closeForm() . . . 3278
SFU.decodeXmlValue() . . . 3278
SFU.encodeXmlValue() . . . 3279
SFU.getClientCredentialToken() . . . 3280
SFU.getDateTimeInStringFormat() . . . 3281
SFU.getDateTimeInServerUTCFormat() . . . 3281
SFU.getDateTimeInUTCFormat() . . . 3282
SFU.getFormUrl() . . . 3283
SFU.getGridValue() . . . 3283
SFU.getImagePath() . . . 3284
SFU.getJSONObjectFromXml() . . . 3285
SFU.getLookupSchemaAndData() . . . 3285
SFU.getServerUtcOffset() . . . 3286
SFU.getStorageItem() . . . 3287
SFU.getUserAccessToken() . . . 3287
SFU.getUserLookupExtendedInformation() . . . 3287
SFU.getUserMemberOf() . . . 3288
SFU.getUserRoles() . . . 3289
SFU.getUserSecurityGroup() . . . 3290
SFU.getValueForRightCode() . . . 3291
SFU.getVirtualActorId() . . . 3292
SFU.getXmlFromJSONObject() . . . 3292
SFU.hideLoader() . . . 3293
SFU.includeCustomCssFiles() . . . 3293
SFU.includeCustomJsFiles() . . . 3294
SFU.initializeWidgetFormCommunication() . . . 3295
SFU.invokeWorkflow() . . . 3295
SFU.isArray() . . . 3296
SFU.isControlInNewMode() . . . 3297
SFU.isEmpty() . . . 3297
SFU.isFormInEditMode() . . . 3298

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SFU.isMobileDisplayMode() . . . 3298
SFU.isNumber() . . . 3299
SFU.isUndefined() . . . 3300
SFU.isWorkTasksPro() . . . 3300
SFU.isWorkTasksProConnectedToInternet() . . . 3301
SFU.logMessageToWMLogger() . . . 3301
SFU.parseNumberValue() . . . 3302
SFU.reloadForm() . . . 3303
SFU.setStorageItem() . . . 3303
SFU.soaMethod() . . . 3304
SFU.showAlert() . . . 3305
SFU.showConfirmation() . . . 3305
SFU.showError() . . . 3306
SFU.showLoader() . . . 3307
SFU.showPopupControl() . . . 3308
SFU.showWarning() . . . 3308
SFU.subscribeToControlParameters() . . . 3309
SFU.updateFormHeaderForWorkTasksPro() . . . 3310
control.addRecord() . . . 3310
control.addRecordToBottom() . . . 3311
control.addRecordToPosition() . . . 3312
control.addRecordToTop() . . . 3313
control.applyFilter() . . . 3313
control.associatedForm . . . 3314
control.clearContent() . . . 3315
control.clearFilter() . . . 3315
control.controls() . . . 3316
control.dataContainerForm . . . 3316
control.findAllById() . . . 3318
control.findAllByName() . . . 3318
control.findAllByXmlNode() . . . 3319
control.findById() . . . 3319
control.findByName() . . . 3320
control.findByXmlNode() . . . 3320
control.findChildRecordControlById() . . . 3321
control.findChildRecordControlByName() . . . 3322
control.findChildRecordControlByXmlNode() . . . 3322
control.findImmediateChildRecordControlById() . . . 3323
control.findImmediateChildRecordControlByName() . . . 3323
control.findImmediateChildRecordControlByXmlNode() . . . 3324
control.getFormSubmissionData() . . . 3324
control.getType() . . . 3325
control.refreshCard() . . . 3326
control.refreshData() . . . 3326
control.refreshGrid() . . . 3327
control.refreshTree() . . . 3327
control.remove() . . . 3328
control.removeAll() . . . 3328
control.removeAt() . . . 3329

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

control.removeRecord() . . . 3329
control.submitClick() . . . 3330
control.selectedTabIndex() . . . 3331
control.selectRecord() . . . 3331
control.selectTab() . . . 3332
control.submitForm() . . . 3332
control.sumById() . . . 3333
control.sumByName() . . . 3333
control.sumByXmlNode() . . . 3334
control.topLevelForm() . . . 3335
control.topLevelForm.actionbar.getActionItemByName() . . . 3335
control.updateLevel1ProgressDisplay() . . . 3336
control.updateLevel2() . . . 3337
control.updateLevel2ProgressDisplay() . . . 3337
control.updateLevel2SelectItem() . . . 3338
control.updateLevel2ToggleDisplay() . . . 3339
control.updateLevel3() . . . 3339
control.updateLevel3ProgressDisplay() . . . 3340
control.updateLevel3SelectItem() . . . 3341
control.updateLevel3ToggleDisplay() . . . 3341
control.validateControl() . . . 3342
control.validateForm() . . . 3343
control.validateRecord() . . . 3343
isControlVisible() . . . 3344
skelta.localize.getString() . . . 3344
skelta.userContext.getUserContextFor() . . . 3345
skelta.grid.getDataSource() . . . 3346
Web APIs . . . 3346
Web API Certificate-based Authentication . . . 3348
Web API Server Configuration . . . 3349
Web API Client Configuration . . . 3353
Web API Type files for Swagger/OpenAPI based Web APIs . . . 3355
Web API Source . . . 3356
Import a New Form using an API . . . 3357
Scenarios . . . 3358
Scenario 1: Purchase Order Form . . . 3358
Scenario 2: Vehicle Registration Form . . . 3360
Scenario 3: EMI Form . . . 3363
Examples . . . 3366
Accessing Values of a Selected Row Through Other Controls . . . 3366
Add Data Grid to Workflows . . . 3367
Create a Hierarchical Data Grid . . . 3373
Customize Forms Using Custom Styles . . . 3373
Data Binding Drop-Down Controls Using Lookups . . . 3378
Data Binding Drop-Down Controls Using Lookup Name (Non-parameterized) . . . 3378
Data Binding Drop-Down Controls Using a Lookup Name (Parameterized) . . . 3378
Delete Record in a Base Form Control . . . 3379
Get User Information of Logged-on User . . . 3380
Hide the Column Header in a Data Grid . . . 3381

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Localize BlockingOutput Messages for Invoke Workflow Control . . . 3381
Prevent Duplicate Data Entries in Controls . . . 3381
Print a Form . . . 3383
Select and Act on Multiple Work Items Using the Enable Multi Select option . . . 3385
Set the Consider Available Height and Dynamic Page Size Properties . . . 3386
Set Values for a Data Grid on Post Execution of a Workflow . . . 3386
Show Embedded Data from Web API Lookup in Data Grid . . . 3388
Show Icons in Data Grid Columns . . . 3389
Update Column Names in a Base Form . . . 3391
Use Data from Data Grid Set As Data Source . . . 3391
Use Data from Parameterized Data Grid in Base Form . . . 3392
Use the Security APIs . . . 3393
View Attachments uploaded from Forms . . . 3394
Access Attachments from Another Form . . . 3396
Troubleshooting Forms . . . 3399
Troubleshoot . . . 3399
Supported Library Versions . . . 3405
FAQs . . . 3414
Lists . . . 3414
Introduction . . . 3414
Access Lists . . . 3415
Create and Organize . . . 3419
Map XML Nodes . . . 3419
Create List . . . 3420
Create Child List Items . . . 3422
Create Folders for Lists . . . 3423
Organize Lists . . . 3423
View Child Items of Lists . . . 3423
Manage Data . . . 3424
Import and Export . . . 3425
Import Lists . . . 3425
Export Lists . . . 3425
Design and Preview . . . 3426
Design Lists . . . 3426
Edit Lists . . . 3426
Delete Lists . . . 3427
Preview Lists . . . 3427
Manage and Secure . . . 3427
Delete List Items . . . 3428
Lock Lists . . . 3428
Unlock Lists . . . 3428
Filter Lists . . . 3429
Change Ownership of Lists . . . 3430
Set Security of Lists . . . 3431
View Audit History of Lists . . . 3433
Regenerate Forms . . . 3433
Lists Designer . . . 3434
Manage Lists . . . 3437
Import Lists . . . 3437

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Export Lists . . . 3438
Edit Properties of Lists . . . 3438
View Form Structure . . . 3438
Auto Generating Forms . . . 3439
Manage Controls . . . 3439
Add Controls . . . 3440
Replace Controls . . . 3441
Move Controls . . . 3442
Delete Controls . . . 3442
Edit Properties of Controls . . . 3442
Lists API . . . 3443
Create List using ListDefinition . . . 3443
Update List Item by Validating Security . . . 3450
Delete List Item by Validating Security . . . 3452
Get Values of Default Field of List . . . 3453
Save ListItem . . . 3455
Advanced List Features . . . 3457
List Control Properties . . . 3457
List(Parent) . . . 3457
Attachment . . . 3460
BooleanInput . . . 3461
DateInput . . . 3463
DecimalInput . . . 3465
GUIDInput . . . 3467
IntegerInput . . . 3469
UserLookup . . . 3471
LookupInput . . . 3474
TextInput . . . 3477
List Table . . . 3479
XML Variables for Lists . . . 3480
Scope of List XML Variable . . . 3480
Create a Workflow for a List in a Single Step . . . 3480
Schema of XML List XML Variable . . . 3482
Create an XML List Type XML Variable . . . 3482
Change the Scope of the XML List XML Variable . . . 3483
Set up the List Query Scope . . . 3484
Create an XML List Iterator Type XML Variable . . . 3485
Create an XML Iterator Type XML Variable for Grid Items . . . 3486
XML Variable Scenarios . . . 3487
Add a New List Item . . . 3487
Update the List Item that Alerted the Workflow . . . 3488
Save the List item as Different Version as Draft/Published using XML Variable . . . 3490
Add New Item to a List Folder using List XML Variable . . . 3490
Delete an Item from a List Folder using XML Variable . . . 3491
Move an Item to a Folder using List XML Variable . . . 3492
Create a New Folder using XML Variable . . . 3493
Add a New List Item with Grid Items . . . 3494
Add a New Grid Item to the Existing List Item . . . 3496
Copy a List Item with Grid Items . . . 3498

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Iterate through the List Item and Grid Item of the List . . . 3501
List Views . . . 3502
List View Introduction . . . 3503
Work with List Views . . . 3503
About Creating List Views . . . 3503
Create a New View . . . 3504
Create a Default View . . . 3505
Create a Public View . . . 3506
Create a Private View . . . 3507
Add Filters . . . 3507
Filter Grouping . . . 3509
Use Date Time Variable to Filter Columns . . . 3510
Sort the Columns . . . 3511
System Variables . . . 3511
Date Time Variables . . . 3512
Workflow Variables . . . 3513
User Variables . . . 3514
Group Columns . . . 3514
Display All Items . . . 3515
Create a Read-Only View . . . 3515
Arrange Grid Columns . . . 3516
Copy from Existing Views . . . 3517
Manage Views . . . 3517
Update a View . . . 3518
Delete a View . . . 3518
Secure a View . . . 3519
List Lookups . . . 3521
Supported Data Types . . . 3521
Start Lookup Settings . . . 3522
Lookup Settings Interface . . . 3522
Lookup Settings . . . 3524
Add a New Repository List Lookup . . . 3524
Add a List Query . . . 3527
Add an Order By Query . . . 3528
Add a New Database Lookup . . . 3529
Add a New Database Lookup Using Query . . . 3529
Add a New Database Lookup Using Stored Procedure . . . 3532
Add a New Web API Lookup . . . 3533
Edit a List Lookup . . . 3535
Organize Lookup Lists . . . 3535
Delete a List Lookup . . . 3536
List Lookup Scenarios . . . 3536
Create a Lookup for a Database Table . . . 3536
Create a Lookup for a Database Table with Composite Primary Key . . . 3539
Create Dependent Lookup from Database Tables . . . 3544
Create a Lookup for a List . . . 3546
Search or Sort within a Lookup Data . . . 3550
Add an Item from Lookup Control . . . 3550
Use ExtendedInformation Property for List Lookups . . . 3551

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Use ExtendedInformation Property for Database Lookups . . . 3552
Use Aliases, Joins, Aggregate Functions in Database Lookup Query . . . 3553
Use CASE Expression in the Database Lookup Query . . . 3554
List Scenarios . . . 3555
Create a Leave Application List . . . 3555
Create a List Mapped to Multiple Tables . . . 3559
Modify the Default List Form . . . 3564
Sort List Data . . . 3565
Export List Item Data to an Excel Sheet . . . 3565
Enable Versioning for List Item . . . 3566
Sort List items using DisplaySortOrder Property . . . 3575
Set Custom View for a List Control from a Custom ASPX Page using API . . . 3576
Lists Integration . . . 3577
Introduction . . . 3577
Need for Lists Integration . . . 3578
About Lists Integration Components . . . 3578
Key Features and Benefits . . . 3579
Prerequisites . . . 3580
Installation and Configuration . . . 3580
64-Bit Installation . . . 3580
About Configuration . . . 3580
List Source Data Flow . . . 3584
About List Source Data Flow . . . 3584
List Destination Data Flow . . . 3586
Import and Export Data . . . 3589
How to Export Data to Excel? . . . 3589
How to Import Data from Excel? . . . 3595
Troubleshooting Lists . . . 3604
Known Issues . . . 3604
Troubleshoot Lists . . . 3604

Quickflow User Guide . . . 3606

Getting Started . . . 3606
Prerequisites . . . 3606
Deploy Quickflow . . . 3608
Create Quickflow . . . 3609
Save Workflow as Quickflow . . . 3613
Activities . . . 3613
Supported Activities . . . 3613
Start Activity . . . 3617
Start - Variables . . . 3620
XML Variables . . . 3620
About XML Variables . . . 3620
XML Variable User Interface . . . 3621
Introduction to XML Document XML Variable . . . 3623
Create XML Variable of Type 'XML Document' . . . 3624
Build XML Schema . . . 3625
Determine XML Variable Schema from Forms . . . 3627

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Introduction to 'XML Iterator' XML Variable . . . 3628
Create XML Variable of Type 'XML Iterator' . . . 3629
Use 'XML Iterator' Type Variables . . . 3630
Delete XML Variables . . . 3635
Communication Activities . . . 3636
Notification Activity . . . 3636
Notification: ToResource . . . 3640
Notification: Message . . . 3643
AVEVA Work Tasks Pro Push Notification Activity . . . 3645
Configure Recipient Resource . . . 3646
Set Notification Type . . . 3649
AVEVA Work Tasks Pro Push Notification Activity Scenarios . . . 3650
Configure Notification Message . . . 3651
Engine Activities . . . 3651
BlockingOutput Activity . . . 3652
Database Activity . . . 3653
Decision Activity . . . 3660
Decision: Decisions . . . 3660
ForEach-Loop Activity . . . 3665
For-Loop Activity . . . 3668
For-Loop Count . . . 3669
Logger Activity . . . 3670
Log - Log Message . . . 3672
Next Loop Activity . . . 3673
Rule Activity . . . 3674
Rule - Condition . . . 3677
Set Workflow Output Variable . . . 3678
Synchronize Activity . . . 3683
Update Parent Variables Activity . . . 3684
Update Parent Variables - List Parent Variables . . . 3687
How to Update a Parent Workflow Variable from a Child Workflow? . . . 3688
Update Variable Activity . . . 3690
Workflow Activity . . . 3690
Workflow - Variables for ChildWorkflow . . . 3695
Workflow - XmlVariables for ChildWorkflow . . . 3696
How to Pass Variables from Parent Workflow/Quickflow to Child Workflow? . . . 3696
Getting Execution ID of the Child Quickflow . . . 3698
XML Navigator . . . 3699
Integration Activities . . . 3703
Invoke Web API Activity . . . 3703
Invoke Web API: Configure Web API . . . 3705
Invoke Web API: Configure OpenAPI . . . 3706
Script Activity . . . 3710
Script: Code . . . 3712
Script: References . . . 3714
Script: Return Values . . . 3715
XML Action Activity . . . 3716
Message Queue Activities . . . 3720
SendMQData . . . 3720

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure Message Queue Connection . . . 3721
Logic App Activities . . . 3722
Call Logic App Activity . . . 3722
Configure Logic App Connection and Data . . . 3723
AttributeReadWrite Activity . . . 3724
Configure AttributeReadWrite Activity to Perform Read Operation . . . 3726
Configure AttributeReadWrite Activity to Perform Write Operation . . . 3728
Relative Referencing . . . 3730
Custom Activity . . . 3731
Process Designer . . . **3732**
Export Process Definition . . . 3732
Import Process Definition . . . 3732
Build Expression . . . 3733
Support . . . **3734**
Diagnostics . . . 3734
Limitations . . . 3735
Troubleshoot . . . 3735

User Guide . . . 3737

Administration . . . **3737**
Prerequisites . . . **3737**
Get Started . . . **3737**
Create a Workflow . . . 3737
Use Variables in Workflow . . . 3740
Use XML Variables in Workflow . . . 3742
Familiarize with XML Variables . . . 3742
Loop Through an XML Document . . . 3745
Use Forms . . . 3750
Invoke Web Service from Workflows . . . 3750
Interfaces . . . **3754**
Access Features . . . 3754
Enterprise Console . . . **3755**
Enterprise Console Authentication . . . 3755
Enterprise Console Home Page . . . 3756
Workflow Management . . . 3758
What is a Workflow List . . . 3758
Group Workflow Items within a Folder . . . 3758
Create a Folder . . . 3759
Publish a Folder . . . 3759
Delete a Folder . . . 3759
Enable or Disable a Folder . . . 3759
Change the Owner of the Folder . . . 3760
Apply Security Settings to a Folder . . . 3760
Navigate to Different Folders in Workflows List . . . 3760
Create a Workflow . . . 3760
Create an Association for a Workflow . . . 3761
Associate Workflow with Enterprise Console Events . . . 3761
BAM Events Associations . . . 3762

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Associate Workflow with ArchestrA Events . . . 3762
Associate Workflow with Message Queue . . . 3764
XML Schema Example . . . 3766
JSON Sample Example . . . 3766
Associate Workflow with Logic App Events . . . 3767
View all Associations of a Workflow . . . 3768
Delete Workflow Associations . . . 3768
Disable Workflow Associations . . . 3768
Create different versions of a Workflow . . . 3769
View All the Versions of a Workflow . . . 3769
Modify a Workflow Design . . . 3770
Edit a Workflow . . . 3770
Delete a Workflow . . . 3770
Publish a Workflow . . . 3772
Unpublish a Workflow . . . 3772
Save Workflow as Quickflow . . . 3773
Import a Workflow . . . 3773
Export a Workflow . . . 3773
Lock a Workflow for Editing . . . 3773
Enable and Disable Workflow . . . 3775
Access BAM Configuration . . . 3775
Change Owner Information of a Workflow . . . 3776
Access Workflow Tasks . . . 3777
Different Views Available in a Workflow List . . . 3777
View a Workflow . . . 3777
View the Execution of the Published Workflow . . . 3777
View Child Items of a Workflow . . . 3777
View Audit History . . . 3778
View the Properties of a Workflow . . . 3778
View Exceptions . . . 3778
Set Security at Workflow Listitem Level . . . 3779
Calendars . . . 3779
Resource Calendar . . . 3783
Configure Resource Calendar by Administrator . . . 3784
Check Resource Availability . . . 3787
Queue Management . . . 3788
Configure Queue Management . . . 3789
Participants Tab . . . 3790
Dispatch Patterns Tab . . . 3795
Settings Tab . . . 3797
Escalation Tab . . . 3798
Use Queue Control to Create and Manage Queues . . . 3800
Add Participants to the Queue . . . 3800
Configure Dispatch Patterns . . . 3802
Set the Queue Preference . . . 3804
Set Escalations for Queues . . . 3805
Use Queues in Workflows . . . 3806
Access Queues in the Work Items List . . . 3807
Database Connections . . . 3807

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create a Database Connection . . . 3807
Work Items List . . . 3811
Manage Work Item - View All Users . . . 3812
Work Items List User Interface Elements . . . 3812
Work Items List Folders . . . 3813
Work Items List View . . . 3814
Work Item Details . . . 3814
Work Items List Toolbar . . . 3815
Work Items List Manager View . . . 3817
Queue Multi-Select and Reassign Queue Work Items . . . 3817
Access Queue Items . . . 3818
Customize Work Items List . . . 3820
Customize Work Items List Layouts . . . 3820
Work Items List Layout Customization . . . 3820
Define Layouts at Workflow Level . . . 3821
Define Global Layouts . . . 3823
Define Layouts for Queues . . . 3823
Define Work Items List Layouts for a User . . . 3824
Customize Work Items List Views . . . 3826
Display Work Items List by Application . . . 3826
Display Work Items List Grouped by Workflows . . . 3826
Display Work Items List for Specified Workflows . . . 3827
Display Work Items List for Specified Queues . . . 3827
Display Inbox View . . . 3827
Display Manager's View . . . 3827
Preferences for Work Item Management . . . 3827
Auto-ownership on Row Click . . . 3828
Enable Multi-Select Option in Work Item List . . . 3828
Filter Data in Work Items List . . . 3829
Manage Work Items . . . 3831
Forward Work Items . . . 3831
Abort Work Items . . . 3834
Remove Work Items . . . 3834
Alternate Actors . . . 3834
Alternate Actors . . . 3835
Configuration Scenarios . . . 3838
Business Activity Monitoring Alert . . . 3844
Set the Threshold Value in BAM . . . 3844
Set BAM Alert for Overdues and KPIs . . . 3845
BAM Events Associations . . . 3846
BAM Workflow . . . 3847
Example to Configure BAM Alerts . . . 3847
BAM Alert Settings . . . 3847
Enabling and Disabling SOA Folder . . . 3848
Create KPI Data Points . . . 3849
Create KPI(s) and Set the Threshold Value . . . 3850
BAM Alert Configuration Settings . . . 3850
BAM Rule Settings . . . 3850
BAM Time Settings . . . 3851

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

BAM Action Settings . . . 3851
BAM Events in Enterprise Console Site . . . 3851
Trigger a BAM Alert . . . 3852
Business Activity Monitoring and Reports . . . 3852
BAM Dashboard View . . . 3856
BAM Execution Reports . . . 3857
BAM Analyze Reports . . . 3859
BAM Process Execution View . . . 3864
File Watcher . . . 3870
File Watcher Operations . . . 3870
Add Folder . . . 3871
Edit Folder Configuration Settings . . . 3872
Delete Folder . . . 3873
File Watcher Events . . . 3874
Created Event . . . 3874
Changed Event . . . 3876
Renamed Event . . . 3877
Deleted Event . . . 3879
Limitations of File Watcher . . . 3880
Frequently Asked Questions . . . 3881
Object Access . . . 3881
SOA Folders . . . 3883
Add New SOA Folder Details . . . 3883
View Registered SOA Folder Details . . . 3884
Delete an SOA Folder . . . 3884
Enable and Disable SOA Folder . . . 3884
Security Settings for SOA Folder . . . 3884
Assemblies . . . 3885
Register Assemblies From GAC . . . 3885
Register Assemblies From Assembly Folder . . . 3885
Register Assemblies From Web . . . 3886
Create Folder for Registered Objects . . . 3886
View Registered Assembly Details . . . 3887
Edit an Object in SOA Assembly . . . 3887
Organize SOA Assembly . . . 3887
Delete a Registered Assembly . . . 3888
Manage Security for a Assembly . . . 3888
Instantiate an Assembly Class in Workflow . . . 3889
SOA Object Instances in SOA Property . . . 3889
SOA Object Instances in Expression Editor . . . 3892
SOA Web Objects in Expression Editor . . . 3892
Difference Between Integration and SOA . . . 3892
Security Features for Folders and Assemblies . . . 3892
Limitations of SOA . . . 3892
SOA Activity . . . 3893
Use SOA Execute Activity in Workflows . . . 3893
Security Framework . . . 3898
Introduction to Security Framework . . . 3900
View Security Mapping at Repository Level . . . 3900

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Components of Security Framework 3901
Security Rights 3901
Security Rights Available for a Repository 3902
Types of Security Rights 3909
Difference between Value for the Rights and Follower Specific Value for Rights 3909
Dictator and Follower 3910
Security Group 3911
Manage Security Groups 3912
Create Security Group 3912
Modify Security Group 3914
Example for Allow or Disallow 3914
Example for Requires Approval and Requires Approval With Prompt 3915
View Security Group 3916
Delete Security Group 3917
Effective Mapping of a Security Group 3917
Security Resource Mapping 3918
Manage Security Group - Resource Mapping 3919
Create a Resource Mapping 3920
Edit Resource Mapping 3921
Delete Resource Mapping 3922
View Effective Security Rights for User or Role 3922
View Effective Security Rights for Roles 3922
View Effective Security Rights for User/Role 3923
Levels of Security 3923
Set Security at Repository Level 3924
Set Security at List Level 3924
Set Security at ListItem Level 3925
Overwrite Definitions from Above Level 3925
Choose between Mapping and Customization 3925
Enable Security at List Level 3926
Enable Security at List Level by Enabling Resource Mapping 3926
Add a Resource Mapping 3927
Edit a Resource Mapping 3928
Delete a Resource Mapping 3928
Enable Security at List Level by Customizing the Security Group Definition . . . 3929
Remove Mapping Available at List Level 3931
Remove Customization Available at List Level 3931
Enable Security at ListItem Level 3931
Enable Security at ListItem Level by Enabling Resource Mapping 3932
Add a Resource Mapping 3933
Edit a Resource Mapping 3934
Delete a Resource Mapping 3934
Enable Security at List Item Level by Customizing the Security Group Definition 3935
Remove Mapping Available at ListItem Level 3937
Remove Customization Available at ListItem Level 3937
Effect on Resource Rights after Deleting the Mapping 3938

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Synchronization . . . 3939
Secure Repository Using Security Framework . . . 3940
Secure List and ListItems . . . 3940
Secure AVEVA Work Tasks Server Controls . . . 3940
Secure Menus . . . 3941
Manage Navigation List . . . 3941
Create Folder . . . 3942
Create Navigation Link . . . 3942
Create Form Link . . . 3945
Create List Link . . . 3947
Create Dashboard Link . . . 3948
Edit Navigation List Items . . . 3949
Change Owner Information . . . 3949
Organize Navigation List . . . 3950
Secure Event Providers . . . 3950
Manage Event Providers . . . 3950
Create an Event Provider . . . 3950
Edit an Event Provider . . . 3951
Change Owner Information . . . 3952
Secure Event Provider . . . 3952
Secure Queue Control . . . 3953
Control New Queue Addition . . . 3953
Control Queue View . . . 3954
Secure Work Item Control . . . 3955
Secure Work Item Control - Show Queue . . . 3957
Secure Work Item Control - Show Alternate Actor . . . 3957
Secure Work Item Control - Show Resource Calendar . . . 3958
Secure Work Item Control - Set Preferences . . . 3958
Secure Work Item Control - View . . . 3959
Secure Work Item Control - Forward . . . 3959
Secure Package . . . 3960
Secure Actions in Package Template . . . 3960
Control Generation of Package Template . . . 3961
Control Package Template Export . . . 3962
Control Package Template Marking . . . 3962
Secure Actions in Manage Package . . . 3963
Control Package Install . . . 3964
Control Package Uninstall . . . 3964
Secure Calendar Control . . . 3965
Control Adding a New Calendar . . . 3966
Control Deleting an Existing Calendar . . . 3966
Control Viewing All Calendars . . . 3967
Control Viewing Base Calendar . . . 3967
Control Viewing Calendar Control . . . 3968
Secure BAM Control . . . 3968
Use BAM Security . . . 3968
Default Rights for Different Security Groups . . . 3971
BAM Report Rights . . . 3971
BAM Tree Menu Rights . . . 3972

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Views . . . 3974
Actions . . . 3975
BAM Configuration . . . 3976
Execution Details Window . . . 3977
Can View Global Reports . . . 3980
Workflowscope_InitiatedWorkflow . . . 3983
Workflowscope_OwnedWorkflow . . . 3985
Workflowscope_Any . . . 3988
Can View BAM Control . . . 3991
Secure Process Designer . . . 3993
Process Designer Rights . . . 3994
Inclusion and Exclusion . . . 3995
Action Rights . . . 3996
Secure Categories . . . 3996
Secure Activities of Categories . . . 3997
Secure Properties of Activities . . . 3998
Menu Rights . . . 3999
Import Behavior for Restricted Activities and Activity Properties . . . 4000
Design Behavior for Restricted Activities and Activity Properties . . . 4001
Secure Process Designer Control at Workflow Level . . . 4002
Can Overwrite Published Workflow Right . . . 4010
Advanced Security Options . . . 4012
Set up Security Approval Workflow . . . 4012
Manage Security Rights . . . 4013
Create a Security Right . . . 4014
Modify a Security Right . . . 4016
View Security Right . . . 4017
Delete a Security Right . . . 4018
Scope of Rights . . . 4018
Create User Defined Operations and Rights . . . 4019
Enable and Disable Security at List/List item level . . . 4024
Dynamic Form Rendering Based on Security . . . 4028
Enable and Disable Security for a Repository . . . 4034
Http Strict Transport Security Enforcement . . . 4036
Access-Control-Allow-Origin . . . 4037
Security Vulnerabilities . . . 4037
Remove Unwanted HTTP Response Headers . . . 4037
Prevent Clickjacking Attack . . . 4038
SDL Cookie with Secure Flag . . . 4039
Troubleshoot Security Issues . . . 4039
View Others Items in the Lookup Control . . . 4039
View Others Items in the Grid Control . . . 4040
View Others Items in the List . . . 4040
View Others Items Irrespective of Status . . . 4040
Navigation List Menus . . . 4040
Synchronize Roles . . . 4041
Delete Synchronization Cross Domain Tasks . . . 4043
Web Parts . . . 4043
Use Web Parts Infrastructure . . . 4045

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Web Parts Framework . . . 4045
Out-of-the-Box Web Parts . . . 4046
Other Web Part Properties . . . 4052
Authentication and Personalization . . . 4053
Create Custom Web Parts . . . 4054
Web Part Display Modes . . . 4055
Browse Mode . . . 4055
Design Mode . . . 4055
Edit Mode . . . 4056
Web Part Enabled Home Pages . . . 4056
Inbox . . . 4056
Calendar . . . 4058
Queue Management . . . 4060
Business Activity Monitoring . . . 4061
Page 1 . . . 4063
Page 2 . . . 4064
Page 3 . . . 4065
Create Custom Page . . . 4065
Add a Web Part to the Web Part Zone . . . 4067
Delete Web Part . . . 4068
Edit a Web Part . . . 4068
Minimize and Restore Web Parts . . . 4068
Web Part Rights . . . 4069
Personalize Scope in Web Parts . . . 4071
Recommendations in Web Parts . . . 4071
Report Service . . . 4072
Sample Business Case Study . . . 4073
Connections . . . 4073
Configure Report Connection . . . 4073
Create a New Report Connection Item . . . 4074
Modify Report Connection Item . . . 4075
Delete Report Connection Item . . . 4075
Report Templates . . . 4076
Configure Report Templates . . . 4076
Create a New Report Template Item . . . 4076
Modify a Report Template Item . . . 4078
Delete a Report Template Item . . . 4078
Reports . . . 4078
Configure Reports . . . 4078
Create a New Report Item . . . 4078
Modify a Report Item . . . 4080
Delete a Report Item . . . 4080
Organize Reports . . . 4080
Configure Reports for Reporting Service . . . 4081
User Tools . . . 4082
Replace User . . . 4083
Replace User in Artifacts of a Repository . . . 4086
Lists . . . 4086
Replace List Item Owners . . . 4086

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Replace User in Security Resource Mapping . . . 4087
Other Artifacts . . . 4088
Queue . . . 4088
BAM . . . 4089
Work Items . . . 4090
Alternate Actor . . . 4090
Repository Configuration . . . 4091
System User . . . 4091
Repository Administrator . . . 4091
Repository Level Security Group Resource Mapping . . . 4092
Exception Mail Settings . . . 4092
View User Reference . . . 4093
Unlock Locked Items . . . 4094
Deleted List Items . . . 4095
Introduction . . . 4095
Use Deleted List Items . . . 4096
Enable or Disable Deleted List Items feature for a Repository . . . 4097
Package . . . 4099
Advantages of Package . . . 4099
About Package . . . 4100
Package Template . . . 4100
Manage Package . . . 4101
ArchestrA Events . . . 4101
ArchestrA Events and AVEVA Work Tasks . . . 4101
Associate Workflow with ArchestrA Events . . . 4103
ArchestrA Events Lookup List . . . 4105
Create ArchestrA Events List Item . . . 4106
Edit ArchestrA Events List Item . . . 4108
Delete ArchestrA Events List item . . . 4108
Process Designer . . . 4109
About the Process Designer . . . 4109
Save File as EMF Image . . . 4124
Save File as PNG Image . . . 4125
Import Process Definition XML File . . . 4125
Export Process Definition XML file . . . 4126
Import Process Definition from Visio . . . 4128
Add Work Item Fields . . . 4129
Alternate Actors . . . 4130
BAM Configuration . . . 4132
Overdue Analysis . . . 4133
KPI Data Points . . . 4135
KPI(s) . . . 4138
Performance . . . 4138
Alert Configuration . . . 4140
Basic Configuration . . . 4140
Overdue Analysis Report . . . 4144
Link Actions . . . 4146
Process Designer File Operations . . . 4146
Reuse Existing Process Designs by Renaming Them . . . 4146

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Clear the Process Diagram . . . 4147
Save the Process Diagram as Image . . . 4147
Process Designer Edit Operations . . . 4147
Copy and Paste Activities . . . 4147
Undo and Redo Actions . . . 4149
Set Properties in the Activity Properties Window . . . 4149
Import and Export Functions in Process Designer . . . 4149
Import from XML . . . 4149
Export to XML . . . 4150
Import from Visio (BPMN Stencil) . . . 4150
Add Activities in Process Designer . . . 4150
Drag and Drop Activities . . . 4151
Add Activities Using the Context Menu . . . 4152
Link Activities in Process Designer . . . 4153
Link Activities by Clicking and Linking . . . 4154
Link Activities Using the Link Tool . . . 4155
Set Link Options in the Activities Context Menu . . . 4155
Set Activity and Link Properties . . . 4156
Identify Mandatory Properties . . . 4156
Method 1: Right Click and Select Activity Properties . . . 4157
Method 2: Select Activity and then Select Activity Properties from Edit Menu . . . 4157
Method 3: Use Wizards For Resource Activities (where applicable) . . . 4158
Introduction to Property Setting Wizards . . . 4159
Use the Actors Selection Wizard . . . 4160
Use the Escalation Email Wizard . . . 4163
Use the Expression Editor . . . 4167
About the Expression Editor . . . 4167
Use Expression Editor to Build Expressions for Properties . . . 4169
Set Appearance Properties for Activity Icons . . . 4176
Search for Content in Workflow . . . 4178
Use the Process Map Image for Navigation . . . 4178
Process Modeling and Saving the Analyst's View . . . 4179
Lock and Unlock Designers . . . 4179
Enable and Disable Lock for Repository . . . 4180
Different Types of Locks . . . 4181
Manual Lock and Unlock . . . 4182
Auto Lock and Unlock . . . 4182
Use Lock and Unlock on Designers . . . 4182
Process Designer . . . 4182
Forms Designer . . . 4189
List Designer . . . 4191
Use Lock and Unlock on User Defined Lists . . . 4193
Enabling Lock for User Defined List . . . 4194
Effects of Lock and Unlock on User Defined List . . . 4194
User Defined List when Lock is Enabled . . . 4194
User Defined List when Lock is Disabled . . . 4195
Update a Locked List Item Through Workflows . . . 4196
Update a Locked List Item from Invoke Form Activity . . . 4196
Update a Locked List Item from Engine Activity . . . 4196

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

View Lock and Unlock History of List Items . . . 4197
AVEVA Work Tasks Pro . . . 4197
Introduction . . . 4197
Get Started with AVEVA Work Tasks Pro . . . 4198
Install and Configure AVEVA Work Tasks Pro Application . . . 4198
Mobile Device Management . . . 4200
View the Demo of AVEVA Work Tasks Pro App . . . 4203
Log On and Log Off . . . 4203
Online and Offline Support for Controls and Containers . . . 4204
Inbox . . . 4208
Categories . . . 4210
Queue . . . 4212
Fill Form . . . 4213
Drafts . . . 4214
Sent Items . . . 4214
Outbox . . . 4215
Sync . . . 4216
Save Settings . . . 4217
Change Notification Tone for iOS Devices . . . 4217
Configure Save Settings . . . 4217
Configure Number of Items to Sync for the Lookup Control . . . 4218
Configure Thresholds . . . 4218
Configure Drafts . . . 4218
Configure Sent Items . . . 4219
Configure Inbox Settings . . . 4219
Configure Offline Support . . . 4220
Use Themes . . . 4220
Apply Theme to AVEVA Work Tasks Pro App . . . 4220
Change Style . . . 4221
Logs . . . 4221
Troubleshooting . . . 4223
Debugging . . . 4223

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Welcome to AVEVA Work Tasks

AVEVA Work Tasks is an advanced Business Process and Workflow Management software to manage process flows that include people and systems.

You can model, execute, monitor, analyze, and improve processes that drive higher levels of efficiency, productivity, and responsiveness.

Users can interact with notifications, information, and tasks anytime anywhere in an access-controlled environment at the tip of their finger on phones, through emails, or using a browser on a Desktop computer.

The web-based architecture offers scalability and integration with business applications, third-party systems, and many data sources such as, Office Outlook, Azure Logic Apps, and AVEVA's industrial software platforms.

AVEVA Work Tasks is built for extensibility including connectors and can be integrated with industrial connectors like System Platform, MES, and has indirect integration through AMQP to AVEVA Production Management, AVEVA Batch Management, and AVEVA Recipe Management.

AVEVA Work Tasks offers the following features:

Central Configuration: Enables you to define, create and manage repositories for workflows and execute them. A Repository refers to a database that contains workflow definitions, queues, calendars, settings, and workflow execution data for an application.

Enterprise Console: A web-based interface that allows authorized users to build workflow-driven applications and to participate in these applications. The console enables central administration of all workflows associated with a repository. Activities that can be performed from the Enterprise Console include workflow categorization, workflow design, associating workflows with lists, events, initiating workflows for new workflow creation, publishing workflows, creation, management, and association of Forms with workflows and viewing reports. The Enterprise Console also enables administrators to set up and manage queues and calendars for use in workflows.

Repository: A database that contains workflow definitions, queues, calendars, settings, workflow execution data and Business Activity Monitoring (BAM) data. Data Sources for Repositories are defined in the Central Configuration. A Repository can also be created for each client in a hosted environment (Application Service Provider environment).

Repository Data Source: A Repository Data Source is connection information specific to a repository database. AVEVA Work Tasks services use a Repository Data Source to connect to the database.

Forms: The Forms is an easy to use drag and drop interface used for designing forms. You can use forms designer to quickly design forms, maintain form templates, and define business logic. The Forms are used to trigger Workflows.

This documentation is intended for the system administrators, process developers, system integrators, and AVEVA Work Task users. It assumes that you have basic knowledge of business process design and are familiar with Business Process Management Notation (BPMN).

Legal information

© 2015-2025 AVEVA Group Limited or its subsidiaries. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of AVEVA

Group Limited. No liability is assumed with respect to the use of the information contained herein.

Although precaution has been taken in the preparation of this documentation, AVEVA assumes no responsibility for errors or omissions. The information in this documentation is subject to change without notice and does not represent a commitment on the part of AVEVA. The software described in this documentation is furnished under a license agreement. This software may be used or copied only in accordance with the terms of such license agreement. AVEVA, the AVEVA logo and logotype, OSIsoft, the OSIsoft logo and logotype, ArchestrA, Avantis, Citect, DYNSIM, eDNA, EYESIM, InBatch, InduSoft, InStep, IntelaTrac, InTouch, Managed PI, OASyS, OSIsoft Advanced Services, OSIsoft Cloud Services, OSIsoft Connected Services, OSIsoft EDS, PIPEPHASE, PI ACE, PI Advanced Computing Engine, PI AF SDK, PI API, PI Asset Framework, PI Audit Viewer, PI Builder, PI Cloud Connect, PI Connectors, PI Data Archive, PI DataLink, PI DataLink Server, PI Developers Club, PI Integrator for Business Analytics, PI Interfaces, PI JDBC Driver, PI Manual Logger, PI Notifications, PI ODBC Driver, PI OLEDB Enterprise, PI OLEDB Provider, PI OPC DA Server, PI OPC HDA Server, PI ProcessBook, PI SDK, PI Server, PI Square, PI System, PI System Access, PI Vision, PI Visualization Suite, PI Web API, PI WebParts, PI Web Services, PRiSM, PRO/II, PROVISION, ROMeo, RLINK, RtReports, SIM4ME, SimCentral, SimSci, Skelta, SmartGlance, Spiral Software, WindowMaker, WindowViewer, and Wonderware are trademarks of AVEVA and/or its subsidiaries. All other brands may be trademarks of their respective owners.

U.S. GOVERNMENT RIGHTS

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the license agreement with AVEVA Group Limited or its subsidiaries and as provided in DFARS 227.7202, DFARS 252.227-7013, FAR 12-212, FAR 52.227-19, or their successors, as applicable.

AVEVA Legal Resources: https://www.aveva.com/en/legal/

AVEVA Third Party Software Notices and Licenses: https://www.aveva.com/en/legal/third-party-software-license/

Publication date: Wednesday, January 29, 2025

Publication ID: 1223079

Contact information

AVEVA Group Limited
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

https://sw.aveva.com/

For information on how to contact sales and customer training, see https://sw.aveva.com/contact.

For information on how to contact technical support, see https://sw.aveva.com/support.

To access the AVEVA Knowledge and Support center, visit https://softwaresupport.aveva.com.

Acknowledgements

AVEVA Work Tasks may include third party code and other materials, subject to separate licensing. Third party notices and license information are displayed below.

Note:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Website addresses in this section are from the respective companies at the time of publication and subject to change at owners' discretion.
- AVEVA Work Tasks uses Telerik Web UI library that includes a customized jQuery version 1.12.4 with backport fixes for known vulnerabilities in jQuery 1.12.4. For more information, refer to jQuery 1.12.4 Vulnerabilities Detected in Telerik Web UI for ASP.NET AJAX.

AjaxMin

MIT License

Copyright (c) Microsoft Corporation. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE

AmqpNet Lite

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

AngleSharp

The MIT License (MIT)

Copyright (c) 2013 - 2021 AngleSharp

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

AngleSharp.Css

The MIT License (MIT)

Copyright (c) 2013 - 2022 AngleSharp

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

AngularJS

The code in this repository is governed by two licenses

Portions of the code found here was obtained from Google and is provided under and subject to the terms of the MIT License set forth below and all other portions of the code are original works of XLTS.dev and are provided subject to the terms of an end user license agreement, which may be obtained from XLTS.dev. Find more details at https://xlts.dev/angularjs.

For code sourced from https://github.com/angular/angular.js

The MIT License

Copyright (c) 2010-2021 Google LLC. http://angularjs.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

For all other code

The XLTS for AngularJS Software End User License Agreement

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Copyright (c) 2022 XLTS.dev All Rights Reserved.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Use of this source code is governed by a commercial license that must be obtained from XLTS.dev.

Find more details at https://xlts.dev/angularjs.

AntiXSS

The MIT License (MIT)

Copyright (c) 2015 Lars Moelleken

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

AutoMapper

The MIT License (MIT)

Copyright (c) 2010 Jimmy Bogard

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Azure.Identity

The MIT License (MIT)

Copyright (c) 2015 Microsoft

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated

documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Azure.Security.KeyVault.Secrets

The MIT License (MIT)

Copyright (c) 2015 Microsoft

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

bootstrap

The MIT License (MIT)

Copyright (c) 2011-2020 Twitter, Inc.

Copyright (c) 2011-2020 The Bootstrap Authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

bootstrap-datepicker

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

chilkat-x64

Chilkat Software License

PLEASE READ THIS AGREEMENT BEFORE OPENING THIS SOFTWARE PACKAGE. IF YOU OPEN THIS PACKAGE OR KEEP IT FOR MORE THAN THIRTY (30) DAYS, YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THESE TERMS AND CONDITIONS, DO NOT OPEN THIS SOFTWARE PACKAGE. YOU MAY ONLY UNLOCK AND/OR USE THE SOFTWARE FOR WHICH YOU HAVE A PAID- UP LICENSE OR FOR WHICH YOU HAVE LEGALLY RECEIVED AN UNLOCK KEY.

(1) DEFINITION OF TERMS

"Documentation": any explanatory written or on- line material including, but not limited to, user guides, reference manuals and HTML files.

"Licensee": shall refer to the individual licensee, whether as an individual programmer, company, or other organization.

"Software": All material in this distribution including, but not limited to, one or more of the following: source code, object code, byte code, dynamic-link libraries, shared libraries, static libraries, header files, executables, scripts, sample programs, utility programs, makefiles and Documentation.

"Licensed Software": the Software for which Licensee has paid the applicable license fee and received an authorized unlock key.

"Software Application Programming Interface ("API")": the set of access methods, whether provided by Chilkat Software, third parties, or developed by Licensee, through which the programmatic services provided by the Licensed Software are made available.

"End-User Software Product": an application developed by Licensee intended for execution on a computer, that makes use of the Licensed Software in its implementation.

The Licensed Software contains certain runtime libraries and files intended for duplication and distribution by Licensee within End User Software Products to the user(s) of the End User Software Product(s) (the "Redistributable Components").

The Redistributable Components are those files specifically designated as being distributable as part of the Licensed Software.

SPECIAL LIMITED TERM EVALUATION LICENSE

If Licensee has been provided with a copy of the Software for evaluation purposes, Chilkat Software, Inc. ("Chilkat") grants to Licensee, subject to the terms of this Single User License Agreement (excluding Section 3, under which Licensee has no rights) a non-exclusive, non- transferable, non-concurrent limited internal use license for evaluation purposes only. This license is for a period of thirty (30) days, commencing upon receipt of the Software, or, if received electronically, from Licensee's initial downloading date, to evaluate the Software. If

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the Software is acceptable, Licensee agrees to promptly notify his Chilkat Sales Representative. Otherwise, Licensee shall immediately cease any further use of the Software and destroy all copies of the Software (including the original) and related Documentation provided to Licensee by Chilkat.

(2) GENERAL

The Software is owned by Chilkat Software, Inc. ("Chilkat") and is protected by U.S. copyright laws and other laws and by international treaties. It is intended for use by a software programmer who has experience using development tools and class libraries.

(3) LICENSE GRANTS

(a) Per-developer license. Subject to the terms and conditions of this Agreement, Chilkat grants to Licensee the perpetual, non-exclusive, non-transferable, world-wide license for one (1) developer to (i) install and use the Licensed Software on any number of computers, and (ii) use the associated user documentation and online help.

(b) Site-wide license. If you have purchased a site-wide license, the following rights apply notwithstanding section 3(a): Subject to the terms and conditions of this Agreement, Chilkat grants to Licensee the perpetual, non-exclusive, non-transferable, world-wide license for any number of developers at a single Licensee's office location to (i) install the Licensed Software on any number of computers across Licensee's enterprise, and (ii) use the associated user documentation and online help.

(c) Licensee may also:

(i) Make one backup copy of the Licensed Software solely for archival and disaster-recovery purposes, or transfer the Licensed Software to a hard disk and keep the original copy solely for archival and disaster-recovery purposes; and (ii) Reproduce and distribute the Redistributable Components directly or indirectly for any number of applications to any number of end users and Licensee's Authorized OEMs, VARs and Distributors, through customary distribution channels, world wide, on a royalty free basis provided that such distribution is (i) in conjunction with an End User Software Product developed by Licensee using the Licensed Software and (ii) the Licensed Software is not the sole or primary component of such End User Software Product.

(iii) The license rights granted under this Agreement do not apply to development and distribution of software development products or toolkits of any kind that are destined to be used by software developers other than Licensee(s) that are Authorized.

Licensee has no rights to use the Licensed Software beyond those specifically granted in this section.

(4) LICENSE RESTRICTIONS

EXPORT CONTROLS: If the Software is for use outside the United States of America, Licensee agrees to comply with all relevant regulations of the United States Department of Commerce and with the United States Export Administration Act to insure that the Software is not exported in violation of United States law.

Notwithstanding any provisions in this Agreement to the contrary, Licensee may not distribute any portion of the Software other than the Redistributable Components.

In addition, Licensee may not decompile, disassemble, or reverse engineer any object code form of any portion of the Software.

(5) TITLE

Licensee acknowledges and agrees that all right, title and interest in and to the Software, including all intellectual property rights therein, are the property of Chilkat, subject only to the licenses granted to Licensee under this

Agreement. This Agreement is not a sale and does not transfer to the Licensee any title or ownership in or to the Software or any patent, copyright, trade secret, trade name, trademark or other proprietary or intellectual property rights related thereto.

(6) NON-TRANSFERABILITY

Except for Licensee's rights to distribute the Redistributable Components, Licensee may not rent, transfer, assign,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

sublicense or grant any rights in the Software, in full or in part, to any other person or entity without Chilkat's written consent, except that this agreement may be assigned to a successor of Licensee in the case that all or substantially all of the assets or equity of Licensee are acquired by the successor.

(7) LIMITED WARRANTIES

Chilkat warrants to Licensee that the Licensed Software will substantially perform the functions described in the Documentation for a period of thirty (30) days after the date of delivery of the Licensed Software to Licensee. Chilkat's sole and exclusive obligation, and Licensee's sole and exclusive remedy, under this warranty is limited to Chilkat's using reasonable efforts to correct material, documented, reproducible defects in the Licensed Software that Licensee describes and documents to Chilkat during the thirty (30) day warranty period. In the event that Chilkat fails to correct a material, documented, reproducible defect during this period, Chilkat may, at Chilkat's discretion, replace the defective Licensed Software or refund to Licensee the amount that Licensee paid Chilkat for the defective Licensed Software and cancel this Agreement and the licenses granted herein. In such event, Licensee agrees to return to Chilkat all copies of the Licensed Software (including the original).

EXCEPT AS EXPRESSLY SET FORTH ABOVE, CHILKAT EXPRESSLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, RESULTS, OR OTHERWISE

(8) LIMITATION OF LIABILITY

IN NO EVENT SHALL CHILKAT BE LIABLE FOR ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS, REVENUES, DATA OR OTHER ECONOMIC ADVANTAGE) WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY, EVEN IF CHILKAT WAS ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. NOTWITHSTANDING THE FOREGOING, THE TOTAL AMOUNT OF CHILKAT'S LIABILITY TO LICENSEE SHALL BE LIMITED TO THE AMOUNT USER PAID FOR THE USE OF THE SOFTWARE, IF ANY.

(9) TERMINATION

Chilkat reserves the right, at its sole discretion, to terminate this Agreement upon written notice if Licensee has breached the terms and conditions hereof. Licensee may terminate this Agreement at any time by ceasing to use the

Licensed Software and by destroying all copies of the Licensed Software (including the original). Sections 4, 5, 6, 7, 8, 9 and 10 survive any termination of this Agreement and apply fully to any termination. Unless terminated as

provided herein by either party, this Agreement shall remain in effect. Termination will not affect end user licenses of the End User Software Product which contain the Redistributable Components which were distributed by

Licensee prior to termination.

(10) MISCELLANEOUS

Applicable Law and Jurisdiction. This Agreement will be governed by and construed in accordance with the laws of the State of Illinois without regard to conflict of laws principles and without regard to the 1980 U.N. Convention on Contracts for the International Sale of Goods. The federal and state courts of Illinois shall have exclusive jurisdiction and venue to adjudicate any dispute arising out of this Agreement, and Licensee expressly consents to (i) the personal jurisdiction of the state and federal courts of Illinois, and (ii) service of process being effected upon Licensee by registered mail.

Limitation of Actions. No action, regardless of form, may be brought by either party more than twelve (12) months after the cause of action has arisen. No such claim may be brought unless Chilkat has first been given commercially reasonable notice, a full written explanation of all pertinent details (including copies of all materials), and a good faith opportunity to resolve the matter.

Invalidity and Waiver. Should any provision of this Agreement be held by a court of law to be illegal, invalid, or unenforceable, the legality, validity, and enforceability of the remaining provisions of this Agreement will not be

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

affected or impaired thereby. The failure of any party to enforce any of the terms or conditions of this Agreement, unless waived in writing, will not constitute a waiver of that party's right to enforce each and every term and

condition of this Agreement.

U.S. Government Restricted Rights. The Licensed Software is provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software clause at DFARS 52.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software Restricted Rights at 48 CFR 52.227- 19, as applicable. Manufacturer is Chilkat Software, Inc., 1719 E Forest Ave, Wheaton, Illinois 60187 USA.

LICENSEE ACKNOWLEDGES THAT HE HAS READ THIS AGREEMENT, UNDERSTANDS IT AND AGREES TO BE BOUND BY ITS TERMS AND CONDITIONS. LICENSEE FURTHER AGREES THAT IT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN LICENSEE AND CHILKAT WHICH SUPERSEDES ANY PROPOSAL OR PRIOR OR CONTEMPORANEOUS AGREEMENT, ORAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT

CodeMirror

MIT License

Copyright (C) 2017 by Marijn Haverbeke <marijnh@gmail.com> and others Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ComponentArt Charting

For the ComponentArt Charting licensing details, see https://www.componentsource.com/product/componentart-charting.

crypto-js

[The MIT License (MIT)](http://opensource.org/licenses/MIT)

Copyright (c) 2009-2013 Jeff Mott

Copyright (c) 2013-2016 Evan Vosberg

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

datejs

Copyright 2008-2015 Object.NET, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Contact:

https://object.net

hello@object.net

+1(888)775-5888

EasyNetQ

The MIT License

Copyright (c) 2015 Mike Hadlow

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

ExcelDataReader

The MIT License (MIT)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Copyright (c) 2014 ExcelDataReader

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR MPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

FastAndroidCamera

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

HtmlSanitizer

The MIT License (MIT)

Copyright (c) 2013-2016 Michael Ganss and HtmlSanitizer contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Intersoft WebGrid

ACCEPTANCE

YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT BY SELECTING THE "ACCEPT" OPTION IN THE INSTALLATION OF THE SOFTWARE. YOU MUST AGREE TO ALL OF THE TERMS OF THIS AGREEMENT BEFORE YOU WILL BE ALLOWED TO INSTALL THE SOFTWARE. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, YOU MUST SELECT "DECLINE" AND YOU MUST NOT INSTALL, OR USE THE SOFTWARE.

LICENSE GRANT

This Software License Agreement ("Agreement") is a legal agreement between You (an entity or a person) and Intersoft Solutions Corporation. ("Intersoft"). This Copy of Intersoft Premier Studio and accompanying documentation ("Software") is licensed and not sold. This software is protected by the copyright laws and treaties of the United States ("U.S.") and other countries and is subject to the terms of this Agreement.

The Software is licensed per developer seat. This Agreement entitles You to use one copy of the Software at any given time. You may install the Software on up to three computers such as on office PC, work laptop, and personal laptops, or to a virtual machine solely for testing purpose; however, You can only use one installation of the Software at a time. The Software is in "use" on a computer when it is loaded onto a development environment that is used to develop applications using the Software, or build source code that depends on the Software.

This Agreement does not permit the use of the Software on a system that allows shared used of applications, on a multi-user network, or on any configuration or system of computers that allows multiple users. Multiple copy use or installation is only allowed if You obtain an appropriate licensing agreement for each user and each copy of the Software. For further information regarding multiple copy licensing of the Software, please contact us directly via live chat.

This Agreement entitles You to deploy your applications built with the Software to any number of production server, devices or clients with no additional charges. The Software runtime files should not be modified in any ways prior to deploying and all copyright notices should be retain.

RESTRICTIONS ON EVALUATION LICENSE

If the Software is provided to You for evaluation purposes, then Your license to use the Software is limited solely to internal evaluation purposes, and in accordance with the terms of the evaluation offering under which You received the Software, and Your license expires 30 days from installation (or such other period as may be indicated within the Software). Upon expiration of the evaluation period, You must discontinue use of the Software and return to an original state any actions performed by the Software. The Software may contain an automatic disabling mechanism that prevents its use after a certain period of time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

RESTRICTIONS ON TRANSFER

Without first obtaining the express written consent of Intersoft, You may not assign your rights and obligations under this Agreement, or redistribute, encumber, sell, rent, lease, sublicense, or otherwise transfer your rights to the Software.

RESTRICTIONS ON USE

You may not use, copy, or install the Software to be used by more than one user. If You hold multiple, validly licensed copies, You may not use, copy, or install the Software on any system with more than the number of computers permitted by the license, or permit the use, copying, or installation by more users than the number permitted by license.

You may not decompile, "reverse-engineer", disassemble, or otherwise attempt to derive the source code for the Software.

RESTRICTIONS ON ALTERATION

You may not modify the Software or create any derivative work of the Software or its accompanying documentation. Derivative works include but are not limited to translations. You may not alter any files or libraries in any portion of the Software.

RESTRICTIONS ON COPYING

You may not copy any part of the Software except to the extent that licensed use inherently demands the creation of a temporary copy stored in computer memory and not permanently affixed on storage medium. You may make one archival copy which must be stored on a medium other than a computer hard drive.

MAINTENANCE AND SUPPORT

Each license includes updates to the Software that Intersoft makes generally commercially available for a period of one (1) year from the date of Your License. If You renew the license before its expiration date, the maintenance and support for Your License is automatically extended to another one (1) year from the date of the renewal. For more information how to obtain support, please visit Intersoft Support. If You do not renew the license, You may continue to use the license in normal conditions according to the terms in this agreement; however Intersoft has no obligation to provide maintenance or support for You. You may participate in the Intersoft Community Forums or other community sites for discussion and insights. To access Intersoft Community Forums, visit Intersoft Community.

OWNERSHIP

No title to or ownership of the Software is transferred to You. Intersoft retains all right, title and interest in and to all intellectual property rights in the Software, including any adaptations or copies thereof. You acquire only a conditional license to use the Software.

USE OF INTERNET CONNECTION

You acknowledge and agree that the Software may make Internet connections in order to: 1) Check for updates; 2) Anonymously activating and deactivating licenses; 3) Anonymously validate license keys in order to prevent unauthorized use; and 4) Provide other services to You.

LIMITED SOFTWARE WARRANTY

For a period of 60 days from the date of shipment or from the date that You download the Software, as applicable, Intersoft warrants that when properly installed and used under normal conditions, the Software will perform substantially as advertised.

THE SOFTWARE IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE OR DISTRIBUTION WITH ON-LINE CONTROL EQUIPMENT IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION, COMMUNICATION, OR CONTROL SYSTEMS, DIRECT LIFE SUPPORT MACHINES, WEAPONS SYSTEMS, OR OTHER USES IN WHICH FAILURE OF THE SOFTWARE COULD

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

LEAD DIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE.

THE SOFTWARE IS ONLY COMPATIBLE WITH CERTAIN COMPUTERS AND OPERATING SYSTEMS. THE SOFTWARE IS NOT WARRANTED FOR NON-COMPATIBLE SYSTEMS. Contact Intersoft representative for information about compatibility.

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITY

UNLESS OTHERWISE EXPLICITLY AGREED TO IN WRITING BY INTERSOFT, INTERSOFT MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, IN FACT OR IN LAW, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OTHER THAN AS SET FORTH IN THIS AGREEMENT OR IN THE LIMITED WARRANTY DOCUMENTS PROVIDED WITH THE SOFTWARE.

Intersoft makes no warranty that the Software will meet your requirements or operate under your specific conditions of use. Intersoft makes no warranty that operation of the Software will be secure, error free, or free from interruption. YOU MUST DETERMINE WHETHER THE SOFTWARE SUFFICIENTLY MEETS YOUR REQUIREMENTS FOR SECURITY AND UNINTERRUPTABILITY. YOU BEAR SOLE RESPONSIBILITY AND ALL LIABILITY FOR ANY LOSS INCURRED DUE TO FAILURE OF THE SOFTWARE TO MEET YOUR REQUIREMENTS. INTERSOFT WILL NOT, UNDER ANY CIRCUMSTANCES, BE RESPONSIBLE OR LIABLE FOR THE LOSS OF DATA ON ANY COMPUTER OR INFORMATION STORAGE DEVICE.

UNDER NO CIRCUMSTANCES SHALL INTERSOFT, ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE TO YOU OR ANY OTHER PARTY FOR INDIRECT, CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE, OR EXEMPLARY DAMAGES OF ANY KIND (INCLUDING LOST REVENUES OR PROFITS OR LOSS OF BUSINESS) RESULTING FROM THIS AGREEMENT, OR FROM THE FURNISHING, PERFORMANCE, INSTALLATION, OR USE OF THE SOFTWARE, WHETHER DUE TO A BREACH OF CONTRACT, BREACH OF WARRANTY, OR THE NEGLIGENCE OF INTERSOFT OR ANY OTHER PARTY, EVEN IF INTERSOFT IS ADVISED BEFOREHAND OF THE POSSIBILITY OF SUCH DAMAGES. TO THE EXTENT THAT THE APPLICABLE JURISDICTION LIMITS INTERSOFT'S ABILITY TO DISCLAIM ANY IMPLIED WARRANTIES, THIS DISCLAIMER SHALL BE EFFECTIVE TO THE MAXIMUM EXTENT PERMITTED.

LIMITATION OF DAMAGES

Any claims must be made within the applicable warranty period. All warranties cover only defects arising under normal use and do not include malfunctions or failure resulting from misuse, abuse, neglect, alteration, problems with electrical power, acts of nature, unusual temperatures or humidity, improper installation, or damage determined by Intersoft to have been caused by You. All limited warranties on the Software are granted only to You and are non-transferable. You agree to indemnify and hold Intersoft harmless from all claims, judgments, liabilities, expenses, or costs arising from your breach of this Agreement and/or acts or omissions.

GOVERNING LAW, JURISDICTION AND COSTS

This Agreement is governed by the laws of California, without regard to California's conflict or choice of law provisions.

SEVERABILITY

If any provision of this Agreement shall be held to be invalid or unenforceable, the remainder of this Agreement shall remain in full force and effect. To the extent any express or implied restrictions are not permitted by applicable laws, these express or implied restrictions shall remain in force and effect to the maximum extent permitted by such applicable laws.

ENTIRE AGREEMENT

This Agreement becomes effective on the date You legally acquire the Software and will automatically terminate if You breach any of its terms.

This Agreement sets forth the entire understanding and agreement between You and Intersoft regarding the subject matter of this Agreement, and it supersedes all prior agreements between You and Intersoft regarding the Software. This Agreement may only be amended or modified only by a written agreement agreed to by You

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

and an authorized representative of Intersoft. NO LICENSOR, DISTRIBUTOR, DEALER, RETAILER, RESELLER, SALES PERSON, OR EMPLOYEE IS AUTHORIZED TO MODIFY THIS AGREEMENT OR TO MAKE ANY REPRESENTATION OR PROMISE THAT IS DIFFERENT FROM, OR IN ADDITION TO, THE TERMS OF THIS AGREEMENT.

JavaScriptEngineSwitcher

Copyright (c) 2013-2020 Andrey Taritsyn - http://www.taritsyn.ru

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "{}" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright {yyyy} {name of copyright owner}

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

jquery

Copyright OpenJS Foundation and other contributors, https://openjsf.org/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

jquery dotdotdot

This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License.

To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

jquery easing

jQuery Easing - http://gsgd.co.uk/sandbox/jquery/easing/

Uses the built in easing capabilities added In jQuery 1.1 to offer multiple easing options

TERMS OF USE - jQuery Easing

Open source under the 3-Clause BSD License.

Copyright © 2008 George McGinley Smith

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the author nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

TERMS OF USE - EASING EQUATIONS

Open source under the MIT License and the 3-Clause BSD License.

MIT License

Copyright © 2001 Robert Penner

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,

DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

BSD License

Copyright © 2001 Robert Penner

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the author nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

jshint

Copyright 2012 Anton Kovalyov (http://jshint.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE

LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

jszip

The MIT License

Copyright (c) 2009-2016 Stuart Knightley, David Duponchel, Franz Buchinger, António Afonso

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

GPL version 3

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. http://fsf.org/ Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7

additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

Kendo UI

Kendo UI v2021.3.1207 (http://www.telerik.com/kendo-ui)

Copyright 2021 Progress Software Corporation and/or one of its subsidiaries or affiliates. All rights reserved.

Kendo UI commercial licenses may be obtained at http://www.telerik.com/purchase/license-agreement/kendo-ui-complete

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If you do not own a commercial license, this file shall be governed by the trial license terms.

Knockout

The MIT License (MIT) - http://www.opensource.org/licenses/mit-license.php

Copyright (c) 2010 Steven Sanderson, the Knockout.js team, and other contributors

http://knockoutjs.com/

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Knockout-Kendo

This project has been released under the Apache License, version 2.0, the text of which is included below. This license applies ONLY to the project-specific source of each repository and does not extend to Kendo UI itself, or any other 3rd party libraries used in a repository. For licensing information about Kendo UI, see the License Agreements page at KendoUI.com.

Copyright © 2015 Telerik

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

LumiSoft.Net

General usage terms:

*) If you use/redistribute compiled binary, there are no restrictions.

You can use it in any project, commercial and no-commercial.

Redistributing compiled binary not limited any way.

*) It's allowed to complile source code parts to your application, but then you may not rename class names and namespaces.

*) Anything is possible, if special agreement between LumiSoft.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Microsoft.Web.WebView2

Copyright (C) Microsoft Corporation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* The name of Microsoft Corporation, or the names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Newtonsoft.Json

The MIT License (MIT)

Copyright (c) 2007 James Newton-King

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Oracle.ManagedDataAccess

For licensing details, refer to Oracle Technology Network License Agreement.

Oracle Providers for ASP.NET

The MIT License (MIT)

Copyright (c) 2015 Oracle

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Owin

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Paho MQTT

Copyright (c) 2012-2019 Roger Light and others

All rights reserved. This program and the accompanying materials are made available under the terms of the Eclipse Public License v1.0 and Eclipse Distribution License v1.0 which accompany this distribution.

The Eclipse Public License is available at http://www.eclipse.org/legal/epl-v10.html and the Eclipse Distribution License is available at http://www.eclipse.org/org/documents/edl-v10.php.

Contributors:

Roger Light - initial API and implementation

Ian Craggs - MQTT V5 support

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Pako

(The MIT License)

Copyright (C) 2014-2017 by Vitaly Puzrin and Andrei Tuputcyn

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

polly.extensions.http

New BSD License

Copyright (c) 2015-2018, App vNext

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of App vNext nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

RabbitMQ.Client

This package, the RabbitMQ .NET client library, is dual-licensed under the Apache License v2 and the Mozilla Public License v2.0.

For the Apache License, please see the file LICENSE-APACHE2.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For the Mozilla Public License, please see the file LICENSE-MPL-RabbitMQ.

For attribution of copyright and other details of provenance, please refer to the source code.

If you have any questions regarding licensing, please contact us at info@rabbitmq.com.

SendGrid

MIT License

Copyright (C) 2022, Twilio SendGrid, Inc. <help@twilio.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in

the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies

of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Spin js

The MIT License

Copyright (c) 2011-2018 Felix Gnass [fgnass at gmail dot com]

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SQLite

MIT License

Copyright (c) 2016 Game Analytics

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

sqlite-net SQLCipher

Copyright (c) Krueger Systems, Inc.

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

StarkbankEcdsa

MIT License

Copyright (c) 2018 Stark Bank S.A.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Telerik® UI for ASP.NET AJAX

Progress Telerik UI for ASP.NET AJAX 2021

Copyright © 2007-2021 Progress Software Corporation and/or one of its subsidiaries or affiliates. All rights reserved.

Portions of the Product include certain open source and commercial third-party components listed below ("Third-Party Components"). The authors of the Third-Party Components require Progress Software Corporation ("PSC") to include the following notices and additional licensing terms as a condition of PSC's use of such Third-Party Components. You acknowledge that the authors of the Third-Party Components have no obligation to provide support to you for the Third-Party Components or the Product. You hereby undertake to comply with all licenses related to the applicable Third-Party Components. Notwithstanding anything to the contrary, to the extent that any of the terms and conditions of the Product Agreement conflict, vary, or are in addition to the terms and conditions of the aforementioned third-party licenses for these technologies, such terms and conditions are offered by PSC alone and not by any other party.

NOTE: Progress Telerik UI for ASP.NET AJAX 2021 contains and MD5 (as part of TDP (for .NET Framework) 2021 R3), Google Charts for Asp.Net v1.0, DLINQ, and DotNet Color Quantization Code. These third-party technologies are licensed to PSC and are hereby distributed to User only under the terms of a third-party license, a copy of which is included herein below. PSC will, at Licensee's request, provide copies of the source code for these third-party technologies, including modifications, if any, made by PSC. PSC may charge reasonable shipping and handling charges for such distribution. Licensee may also obtain the source code for these third-party technologies through http://iue.progress.com/3dpartysoftwares/Pages/default.aspx by following the instructions set forth therein.

Progress Telerik UI for ASP.NET AJAX 2021 incorporates MD5 (as part of TDP (for .NET Framework) 2021 R3) and Google Charts for Asp.Net v1.0. These third-party technologies are licensed to PSC and are hereby distributed to User only under the terms of the Microsoft Public License (MS-PL) (copied herein below).

Progress Telerik UI for ASP.NET AJAX 2021 incorporates DLINQ. This third-party technology is licensed to PSC and is hereby distributed to User only under the terms of the MICROSOFT LIMITED PUBLIC LICENSE version 1.1 (copied herein below).

Progress Telerik UI for ASP.NET AJAX 2021 incorporates DotNet Color Quantization Code. This third-party technology is licensed to PSC and is hereby distributed to User only under the terms of The Code Project Open License (CPOL) 1.02 (copied herein below).

1. Special Notices Regarding Open-Source Third-Party Components incorporated into the Product:

(1) The Apache Software License, Version 2.0:

Progress Telerik UI for ASP.NET AJAX 2021 incorporates

pdf.js v1.4.20, PDFBox v2.0.21, and NPOI v2.4.1 (as part of TDP (for .NET Framework) 2021 R3) and

AWS SDK.Net v2.0.5.0.

Such technology is subject to the following terms and conditions:

Apache Software License Version 2.0. Copyright 2001-2004 The Apache Software Foundation. Such Apache Technology is subject to the following terms and conditions: The Apache Software License, Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity,

whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof. "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work. 2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed. 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions: (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and (b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License. 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions. 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file. 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License. 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages. 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability. END OF TERMS AND CONDITIONS APPENDIX: How to apply the Apache License to your work. To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives. Copyright [yyyy] [name of copyright owner] Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Contents of NOTICE text file for PDFBox v2.0.21 (as part of TDP (for .NET Framework) 2021 R3):

Apache PDFBox

Copyright 2014 The Apache Software Foundation

This product includes software developed at The Apache Software Foundation (http://www.apache.org/).

Based on source code originally developed in the PDFBox and

FontBox projects.

Copyright (c) 2002-2007, www.pdfbox.org

Based on source code originally developed in the PaDaF project.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Copyright (c) 2010 Atos Worldline SAS

Includes the Adobe Glyph List

Copyright 1997, 1998, 2002, 2007, 2010 Adobe Systems Incorporated.

Includes the Zapf Dingbats Glyph List

Copyright 2002, 2010 Adobe Systems Incorporated.

Includes OSXAdapter

Copyright (C) 2003-2007 Apple, Inc., All Rights Reserved

Add'l text from end of license for PDFBox v2.0.21 (as part of TDP (for .NET Framework) 2021 R3):

EXTERNAL COMPONENTS

Apache PDFBox includes a number of components with separate copyright notices and license terms. Your use of these components is subject to the terms and conditions of the following licenses.

Contributions made to the original PDFBox and FontBox projects:

Copyright (c) 2002-2007, www.pdfbox.org All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of pdfbox; nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Adobe Font Metrics (AFM) for PDF Core 14 Fonts

This file and the 14 PostScript(R) AFM files it accompanies may be used, copied, and distributed for any purpose and without charge, with or without modification, provided that all copyright notices are retained; that the AFM files are not distributed without this file; that all modifications to this file or any of the AFM files are prominently noted in the modified file(s); and that this paragraph is not modified. Adobe Systems has no responsibility or obligation to support the use of the AFM files.

CMaps for PDF Fonts (http://opensource.adobe.com/wiki/display/cmap/Downloads)

Copyright 1990-2009 Adobe Systems Incorporated. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Adobe Systems Incorporated nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PaDaF PDF/A preflight (http://sourceforge.net/projects/padaf)

Copyright 2010 Atos Worldline SAS

Licensed by Atos Worldline SAS under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership.

Atos Worldline SAS licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

OSXAdapter

Version: 2.0

Disclaimer: IMPORTANT: This Apple software is supplied to you by Apple Inc. ("Apple") in consideration of your agreement to the following terms, and your use, installation, modification or redistribution of this Apple software constitutes acceptance of these terms. If you do not agree with these terms, please do not use, install, modify or redistribute this Apple software.

In consideration of your agreement to abide by the following terms, and subject to these terms, Apple grants you a personal, non-exclusive license, under Apple's copyrights in this original Apple software (the

"Apple Software"), to use, reproduce, modify and redistribute the Apple Software, with or without modifications, in source and/or binary forms;

provided that if you redistribute the Apple Software in its entirety and without modifications, you must retain this notice and the following text and disclaimers in all such redistributions of the Apple Software.

Neither the name, trademarks, service marks or logos of Apple Inc. may be used to endorse or promote products derived from the Apple Software without specific prior written permission from Apple. Except as expressly stated in this notice, no other rights or licenses, express or implied, are granted by Apple herein, including but not limited to any patent rights that may be infringed by your derivative works or by other works in which the Apple Software may be incorporated.

The Apple Software is provided by Apple on an "AS IS" basis. APPLE MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND OPERATION ALONE OR IN COMBINATION WITH YOUR PRODUCTS.

IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION, MODIFICATION AND/OR DISTRIBUTION OF THE APPLE SOFTWARE, HOWEVER CAUSED AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (C) 2003-2007 Apple, Inc., All Rights Reserved

Contents of NOTICE text file for AWS SDK.NET v2.0.5.0:

AWS SDK for .NET

Copyright 2009-2017 Amazon.com, Inc. or its affiliates. All Rights Reserved.

****

THIRD PARTY COMPONENTS

****

This software includes third party software subject to the following copyrights:

-Json processing from LitJson –

All the source code and related files distributed with this software have been dedicated to the public domain by the authors.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute the software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

- Parsing PEM files from Bouncy Castle -

Copyright (c) 2000 - 2011 The Legion Of The Bouncy Castle (http://www.bouncycastle.org)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

- Performing CRC32 checks from vbAccelerator.com

vbAccelerator Software License Version 1.0

Copyright (c) 2002 vbAccelerator.com

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

The end-user documentation included with the redistribution, if any, must include the following

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

acknowledgment:

"This product includes software developed by vbAccelerator (http://vbaccelerator.com/)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

The names "vbAccelerator" and "vbAccelerator.com" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact vbAccelerator through steve@vbaccelerator.com.

Products derived from this software may not be called "vbAccelerator", nor may "vbAccelerator" appear in their name, without prior written permission of vbAccelerator.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL VBACCELERATOR OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the vbAccelerator. For more information, please see http://vbaccelerator.com/ .

The vbAccelerator licence is based on the Apache Software Foundation Software Licence, Copyright (c) 2000 The Apache Software Foundation. All rights reserved.

- MD5 checks in the Windows Phone 8 variant of the SDK from Microsoft Corporation

Microsoft Public License (MS-PL)

This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.

1. Definitions

The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law.

A "contribution" is the original software, or any additions or changes to the software.

A "contributor" is any person that distributes its contribution under this license.

"Licensed patents" are a contributor's patent claims that read directly on its contribution.

2. Grant of Rights

(A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

(B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.

3. Conditions and Limitations

(A) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

(B) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(C) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(D) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(E) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

** iOS4Unity - https://github.com/Hitcents/iOS4Unity/

** SQLitePCLRaw - https://github.com/ericsink/SQLitePCL.raw

Apache Software License Version 2.0. Copyright 2001-2004 The Apache Software Foundation. Such Apache Technology is subject to the following terms and conditions: The Apache Software License, Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof. "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution." "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work. 2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.3. Grant of Patent License. Subject to the terms and conditions of this License,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed. 4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions: (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and (b) You must cause any modified files to carry prominent notices stating that You changed the files; and (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License. You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License. 5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions. 6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file. 7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License. 8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages. 9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability. END OF TERMS AND CONDITIONS

(2) BSD-style licenses:

(a) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Jquery Easing v1.3. Such technology is subject to the following terms and conditions:

TERMS OF USE - jQuery Easing

Open source under the BSD License.

Copyright © 2008 George McGinley Smith All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the author nor the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(b) Progress Telerik UI for ASP.NET AJAX 2021 incorporates An O(ND) Difference Algorithm and its Variations v9.0. Such technology is subject to the following terms and conditions:

Software License Agreement (BSD License)

Copyright (c) 2005-2012 by Matthias Hertel, http://www.mathertel.de/

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

•Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

•Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

•Neither the name of the copyright owners nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(c) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Markdown v1.1.1. Such technology is subject to the following terms and conditions:

License

Copyright © 2004, John Gruber

http://daringfireball.net/

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name “Markdown” nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

(d) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Ajax Control Toolkit v4.1.51116. Such technology is subject to the following terms and conditions:

New BSD License (BSD)

Copyright (c) 2012-2018, CodePlex Foundation All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of CodePlex Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(e) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Libtiff.Net and (as part of TDP (for .NET Framework) 2021 R3). Such technology is subject to the following terms and conditions:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

License and Copyright LibTiff.Net, the managed version of the LibTiff library for reading and writing TIFF.

LibTiff.Net

Copyright (c) 2008-2011, Bit Miracle

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

• Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

• Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

• Neither the name of the Bit Miracle nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL BIT MIRACLE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(f) Progress Telerik UI for ASP.NET AJAX 2021 incorporates CMap Resources July 30, 2019 Release and Mapping resources for PDF April 1, 2019 Release (as part of TDP (for .NET Framework) 2021 R3). Such technologies are subject to the following terms and conditions:

Copyright 1990-2019 Adobe. All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are

met:

Redistributions of source code must retain the above copyright notice,

this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright

notice, this list of conditions and the following disclaimer in the

documentation and/or other materials provided with the distribution.

Neither the name of Adobe nor the names of its contributors may be

used to endorse or promote products derived from this software without

specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(g) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Office Binary (doc, xls, ppt) Translator to Open XML v Phase 3 - Milestone 5 (as part of TDP (for .NET Framework) 2021 R3). Such technology is subject to the following terms and conditions:

Copyright (c) 2008-2009, DIaLOGIKa All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of DIaLOGIKa nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY DIaLOGIKa ''AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL DIaLOGIKa BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(3) The Code Project Open License (CPOL) 1.02

Progress Telerik UI for ASP.NET AJAX 2021 incorporates DotNet Color Quantization Code. This third-party technology is licensed to PSC and are hereby distributed to User only under the terms of a third-party license, a copy of which is included herein below:

The Code Project Open License (CPOL) 1.02

Preamble

This License governs Your use of the Work. This License is intended to allow developers to use the Source Code and Executable Files provided as part of the Work in any application in any form.

The main points subject to the terms of the License are:

•Source Code and Executable Files can be used in commercial applications;

•Source Code and Executable Files can be redistributed; and

•Source Code can be modified to create derivative works.

•No claim of suitability, guarantee, or any warranty whatsoever is provided. The software is provided "as-is".

•The Article accompanying the Work may not be distributed or republished without the Author's consent

This License is entered between You, the individual or other entity reading or otherwise making use of the Work licensed pursuant to this License and the individual or other entity which offers the Work under the terms of this License ("Author").

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CODE PROJECT OPEN LICENSE ("LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HEREIN, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE AUTHOR GRANTS YOU THE RIGHTS CONTAINED HEREIN IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO ACCEPT AND BE BOUND BY THE TERMS OF THIS LICENSE, YOU CANNOT MAKE ANY USE OF THE WORK.

1.Definitions. a."Articles" means, collectively, all articles written by Author which describes how the Source Code and Executable Files for the Work may be used by a user.

b."Author" means the individual or entity that offers the Work under the terms of this License.

c."Derivative Work" means a work based upon the Work or upon the Work and other pre-existing works.

d."Executable Files" refer to the executables, binary files, configuration and any required data files included in the Work.

e."Publisher" means the provider of the website, magazine, CD-ROM, DVD or other medium from or by which the Work is obtained by You.

f."Source Code" refers to the collection of source code and configuration files used to create the Executable Files.

g."Standard Version" refers to such a Work if it has not been modified, or has been modified in accordance with the consent of the Author, such consent being in the full discretion of the Author.

h."Work" refers to the collection of files distributed by the Publisher, including the Source Code, Executable Files, binaries, data files, documentation, whitepapers and the Articles.

i."You" is you, an individual or entity wishing to use the Work and exercise your rights under this License.

2.Fair Use/Fair Use Rights. Nothing in this License is intended to reduce, limit, or restrict any rights arising from fair use, fair dealing, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3.License Grant. Subject to the terms and conditions of this License, the Author hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a.You may use the standard version of the Source Code or Executable Files in Your own applications.

b.You may apply bug fixes, portability fixes and other modifications obtained from the Public Domain or from the Author. A Work modified in such a way shall still be considered the standard version and will be subject to this License.

c.You may otherwise modify Your copy of this Work (excluding the Articles) in any way to create a Derivative Work, provided that You insert a prominent notice in each changed file stating how, when and where You changed that file.

d.You may distribute the standard version of the Executable Files and Source Code or Derivative Work in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution.

e.The Articles discussing the Work published in any form by the author may not be distributed or republished without the Author's consent. The author retains copyright to any such Articles. You may use the Executable Files and Source Code pursuant to this License but you may not repost or republish or otherwise distribute or make available the Articles, without the prior written consent of the Author.

Any subroutines or modules supplied by You and linked into the Source Code or Executable Files of this Work shall not be considered part of this Work and will not be subject to the terms of this License.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4.Patent License. Subject to the terms and conditions of this License, each Author hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, import, and otherwise transfer the Work.

5.Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a.You agree not to remove any of the original copyright, patent, trademark, and attribution notices and associated disclaimers that may appear in the Source Code or Executable Files.

b.You agree not to advertise or in any way imply that this Work is a product of Your own.

c.The name of the Author may not be used to endorse or promote products derived from the Work without the prior written consent of the Author.

d.You agree not to sell, lease, or rent any part of the Work. This does not restrict you from including the Work or any part of the Work inside a larger software distribution that itself is being sold. The Work by itself, though, cannot be sold, leased or rented.

e.You may distribute the Executable Files and Source Code only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy of the Executable Files or Source Code You distribute and ensure that anyone receiving such Executable Files and Source Code agrees that the terms of this License apply to such Executable Files and/or Source Code. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute the Executable Files or Source Code with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License.

f.You agree not to use the Work for illegal, immoral or improper purposes, or on pages containing illegal, immoral or improper material. The Work is subject to applicable export laws. You agree to comply with all such laws and regulations that may apply to the Work after Your receipt of the Work.

6.Representations, Warranties and Disclaimer. THIS WORK IS PROVIDED "AS IS", "WHERE IS" AND "AS AVAILABLE", WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES OR CONDITIONS OR GUARANTEES. YOU, THE USER, ASSUME ALL RISK IN ITS USE, INCLUDING COPYRIGHT INFRINGEMENT, PATENT INFRINGEMENT, SUITABILITY, ETC. AUTHOR EXPRESSLY DISCLAIMS ALL EXPRESS, IMPLIED OR STATUTORY WARRANTIES OR CONDITIONS, INCLUDING WITHOUT LIMITATION, WARRANTIES OR CONDITIONS OF MERCHANTABILITY, MERCHANTABLE QUALITY OR FITNESS FOR A PARTICULAR PURPOSE, OR ANY WARRANTY OF TITLE OR NON-INFRINGEMENT, OR THAT THE WORK (OR ANY PORTION THEREOF) IS CORRECT, USEFUL, BUG-FREE OR FREE OF VIRUSES. YOU MUST PASS THIS DISCLAIMER ON WHENEVER YOU DISTRIBUTE THE WORK OR DERIVATIVE WORKS.

7.Indemnity. You agree to defend, indemnify and hold harmless the Author and the Publisher from and against any claims, suits, losses, damages, liabilities, costs, and expenses (including reasonable legal or attorneys’ fees) resulting from or relating to any use of the Work by You.

8.Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL THE AUTHOR OR THE PUBLISHER BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK OR OTHERWISE, EVEN IF THE AUTHOR OR THE PUBLISHER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

9.Termination. a.This License and the rights granted hereunder will terminate automatically upon any breach by You of any term of this License. Individuals or entities who have received Derivative Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 6, 7, 8, 9, 10 and 11 will survive any termination of this License.

b.If You bring a copyright, trademark, patent or any other infringement claim against any contributor over infringements You claim are made by the Work, your License from such contributor to the Work ends automatically.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

c.Subject to the above terms and conditions, this License is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, the Author reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

10.Publisher. The parties hereby confirm that the Publisher shall not, under any circumstances, be responsible for and shall not have any liability in respect of the subject matter of this License. The Publisher makes no warranty whatsoever in connection with the Work and shall not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. The Publisher reserves the right to cease making the Work available to You at any time without notice

11.Miscellaneous a.This License shall be governed by the laws of the location of the head office of the Author or if the Author is an individual, the laws of location of the principal place of residence of the Author.

b.If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this License, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

c.No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

d.This License constitutes the entire agreement between the parties with respect to the Work licensed herein. There are no understandings, agreements or representations with respect to the Work not specified herein. The Author shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Author and You.

(4) MICROSOFT LIMITED PUBLIC LICENSE version 1.1:

Progress Telerik UI for ASP.NET AJAX 2021 incorporates DLINQ. This third-party technology is licensed to PSC and is hereby distributed to User only under the terms of a third-party license, a copy of which is included herein below:

MICROSOFT LIMITED PUBLIC LICENSE version 1.1

This license governs use of code marked as “sample” or “example” available on this web site without a license agreement, as provided under the section above titled “NOTICE SPECIFIC TO SOFTWARE AVAILABLE ON THIS WEB SITE.” If you use such code (the “software”), you accept this license. If you do not accept the license, do not use the software.

1. Definitions

The terms “reproduce,” “reproduction,” “derivative works,” and “distribution” have the same meaning here as under U.S. copyright law.

A “contribution” is the original software, or any additions or changes to the software.

A “contributor” is any person that distributes its contribution under this license.

“Licensed patents” are a contributor’s patent claims that read directly on its contribution.

2. Grant of Rights

(A) Copyright Grant - Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

(B) Patent Grant - Subject to the terms of this license, including the license conditions and limitations in section

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.

3. Conditions and Limitations

(A) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

(B) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(C) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(D) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(E) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

(F) Platform Limitation - The licenses granted in sections 2(A) and 2(B) extend only to the software or derivative works that you create that run directly on a Microsoft Windows operating system product, Microsoft run-time technology (such as the .NET Framework or Silverlight), or Microsoft application platform (such as Microsoft Office or Microsoft Dynamics).

(5) Microsoft Public License (MS-PL):

Progress Telerik UI for ASP.NET AJAX 2021 incorporates MD5 (as part of TDP (for .NET Framework) 2021 R3) and Google Charts for Asp.Net v1.0. These third-party technologies are licensed to PSC and are hereby distributed to User only under the terms of a third-party license, a copy of which is included herein below:

Microsoft Public License (Ms-PL)

This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.

1. Definitions

The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law.

A "contribution" is the original software, or any additions or changes to the software.

A "contributor" is any person that distributes its contribution under this license.

"Licensed patents" are a contributor's patent claims that read directly on its contribution.

2. Grant of Rights

(A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

(B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.

3. Conditions and Limitations

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

(A) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

(B) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(C) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(D) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(E) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

(6) MIT-Style Licenses:

(a) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Cecil v0.9.6.0. Such technology is subject to the following terms and conditions:

Mono.Cecil is licensed under the MIT/X11:

Copyright (c) 2008 - 2011, Jb Evain

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The MIT/X11 is a permissive license, which is GPL compatible, and allows usage within proprietary software as long as this license is distributed along with the software.

(b) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Json.Net v7.0.1.18622. Such technology is subject to the following terms and conditions:

Copyright (c) 2007 James Newton-King

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(c) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Jquery v1.12.4. Such technology is subject to the following terms and conditions:

Copyright jQuery Foundation and other contributors, https://jquery.org/

This software consists of voluntary contributions made by many individuals. For exact contribution history, see the revision history available at https://github.com/jquery/jquery

The following license applies to all parts of this software except as documented below:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

All files located in the node_modules and external directories are externally maintained libraries used by this software which have their own licenses; we recommend you read them, as their terms may differ from the terms above.

(d) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Pako v1.0.1. Such technology is subject to the following terms and conditions:

(The MIT License)

Copyright (C) 2014-2016 by Vitaly Puzrin

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(e) Progress Telerik UI for ASP.NET AJAX 2021 incorporates CsQuery v1.3.5.200 and (as part of TDP (for .NET Framework) 2021 R3). Such technology is subject to the following terms and conditions:

Code use license for CsQuery

Copyright (c) 2012 James Treworgy

LICENSE (MIT License)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(f) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Jquery Throttle v1.1. Such technology is subject to the following terms and conditions:

Copyright © 2010 "Cowboy" Ben Alman

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(g) Progress Telerik UI for ASP.NET AJAX 2021 incorporates jQuery Double Tap Plugin v1.0. Such technology is subject to the following terms and conditions:

Copyright (c) 2010 Raul Sanchez (http://www.sanraul.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(h) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Definitely Typed v0.0.2. Such technology is subject to the following terms and conditions:

This project is licensed under the MIT license.

Copyrights are respective of each contributor listed at the beginning of each definition file.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(i) Progress Telerik UI for ASP.NET AJAX 2021 incorporates SyntaxHighlighter v3.0.83. Such technology is subject to the following terms and conditions:

Copyright 2004-2014 Alex Gorbatchev

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(j) Progress Telerik UI for ASP.NET AJAX 2021 incorporates MarkdownSharp v1.13. Such technology is subject to the following terms and conditions:

Copyright <YEAR> <COPYRIGHT HOLDER>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(k) Progress Telerik UI for ASP.NET AJAX 2021 incorporates PageDown v1.0. Such technology is subject to the following terms and conditions:

Copyright <YEAR> <COPYRIGHT HOLDER>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(l) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Azure SDK for .NET v3.0.2.0. Such technology is subject to the following terms and conditions:

The MIT License (MIT)

Copyright (c) 2015 Microsoft

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(m) Progress Telerik UI for ASP.NET AJAX 2021 incorporates mscorlib v4.6.2 (as part of TDP (for .NET Framework) 2021 R3). Such technology is subject to the following terms and conditions:

The MIT License (MIT)

Copyright (c) Microsoft Corporation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(n) Progress Telerik UI for ASP.NET AJAX 2021 incorporates PDFsharp v1.3.2 (as part of TDP (for .NET Framework) 2021 R3). Such technology is subject to the following terms and conditions:

PDFsharp is published under the MIT License.

Copyright (c) 2005-2014 empira Software GmbH, Troisdorf (Germany)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(o) Progress Telerik UI for ASP.NET AJAX 2021 incorporates JBig2Decoder.NET (as part of TDP (for .NET Framework) 2021 R3). Such technology is subject to the following terms and conditions:

The MIT License (MIT)

Copyright (c) 2014 Daniel Glucksman

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(p) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Open XML SDK v2.5 and (as part of TDP (for .NET Framework) 2021 R3). Such technology is subject to the following terms and conditions:

The MIT License (MIT)

Copyright (c) Microsoft Corporation

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

(q) Progress Telerik UI for ASP.NET AJAX 2021 incorporates HtmlParserSharp v1.4 (as part of TDP (for .NET Framework) 2021 R3). Such technology is subject to the following terms and conditions:

This is for the HTML parser as a whole except the rewindable input stream, the named character classes and the Live DOM Viewer.

For the copyright notices for individual files, please see individual files.

Copyright (c) 2005, 2006, 2007 Henri Sivonen

Copyright (c) 2007-2011 Mozilla Foundation

Portions of comments Copyright 2004-2007 Apple Computer, Inc., Mozilla Foundation, and Opera Software ASA.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The following license is for the WHATWG spec from which the named character data was extracted.

Copyright 2004-2010 Apple Computer, Inc., Mozilla Foundation, and Opera Software ASA.

You are granted a license to use, reproduce and create derivative works of this document.

The following license is for the rewindable input stream.

Copyright (c) 2001-2003 Thai Open Source Software Center Ltd All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Thai Open Source Software Center Ltd nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

(r) Progress Telerik UI for ASP.NET AJAX 2021 incorporates PreMailer.Net v1.2.4.0 (as part of TDP (for .NET Framework) 2021 R3). Such technology is subject to the following terms and conditions:

Copyright (c) 2013 Milkshake Software (http://milkshakesoftware.com)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2. Special Notices Regarding Commercially Licensed Third-Party Components incorporated into the Product: NONE

3. Special Notices Regarding Progress Product Components incorporated into the Product:

(a) Progress Telerik UI for ASP.NET AJAX 2021 incorporates Progress Telerik UI for ASP.NET AJAX Extension v2021.3 and Kendo UI 2017 from Progress Software Corporation. See NOTICE.txt text file within products for third-party license requirements/agreements.

NOTICE FROM PROGRESS SOFTWARE CORPORATION: Additional notices may be included in the release notes or other documentation that accompanies updates received in connection with support of the Product.

Updated 9/9/2021

tensorspace

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Unity.Abstractions

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Unity.Container

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

WindowsBase

The MIT License (MIT)

Copyright (c) 2019 Microsoft

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

WiX Toolset

The WiX toolset is released under the Microsoft Reciprocal License (MS-RL). A reciprocal license is used to ensure that others who build on the effort of the WiX community give back to the WiX community. Specifically the license requires that fixes and improvements to the WiX toolset must be published using the same license.

Sometimes the reciprocal license is incorrectly interpreted to also apply to bundles, packages, and custom actions built using the WiX toolset. The Outercurve Foundation has previously provided this statement below to clarify which now the .NET Foundation reaffirms:

The WiX toolset (WiX) is licensed under the Microsoft Reciprocal License (MS-RL). The MS-RL governs the distribution of the software licensed under it, as well as derivative works, and incorporates the definition of a derivative work provided in U.S. copyright law. OuterCurve Foundation (and the .NET Foundation) does not view the installer packages generated by WiX as falling within the definition of a derivative work, merely because they are produced using WiX. Thus, the installer packages generated by WiX will normally fall outside the scope of the MS-RL, and any of your source code, binaries, libraries, routines or other software components that are incorporated in installer packages generated by WiX can be governed by other licensing terms.

The full text of the MS-RL license is reproduced below. It can also be found in the LICENSE.TXT file included with the source code.

Microsoft Reciprocal License (MS-RL)

This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.

Definitions

The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law.

A "contribution" is the original software, or any additions or changes to the software.

A "contributor" is any person that distributes its contribution under this license.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

"Licensed patents" are a contributor's patent claims that read directly on its contribution.

Grant of Rights

(A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

(B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.

Conditions and Limitations

(A) Reciprocal Grants- For any file you distribute that contains code from the software (in source code or binary format), you must provide recipients the source code to that file along with a copy of this license, which license will govern that file. You may license other files that are entirely your own work and do not contain code from the software under any terms you choose.

(B) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

(C) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(D) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(E) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(F) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

xamarin.essentials

Xamarin.Essentials

The MIT License (MIT)

Copyright (c) Microsoft Corporation

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

xam.plugin.battery

The MIT License (MIT)

Copyright (c) 2016 James Montemagno

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Xml2Json

Copyright 2011-2013 Abdulla Abdurakhmanov

Original sources are available at https://code.google.com/p/x2js/

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ZXing

micjahn/ZXing.Net is licensed under the

Apache License 2.0

A permissive license whose main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Apache License

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

Other Components

Unsigned Components

Below table lists the unsigned third-party components that are shipped with AVEVA Work Tasks. It is advised to verify that the **File names**, **SHA1 Hash**, and **SHA256 Hash** listed in the table matches with those installed on the server.

FileName	FileSize	SHA1 Hash	SHA256 Hash
aaAlarmsAndEventsCommon.dll	39936	293A31EDD95A54EAC817B707DA006187C26220C1	30F9A04CEB7C7AE474274C41976DFC1CD4C2C7D89474DAC716ED130DC393D4B0
aaAlarmsAndEventsLogger.dll	13824	08900E66C85408E203FC8C56855391E3D386548B	CAB4D913AD21376FD5C0DC3C0F91E1A36A49D2C7274D5054FD52FB5D0192A2A1
aaArchestrAEvents.ManageEvent.SDK.dll	13824	A026840374F1F10C43904177DCB00535348A056E	88DEE18D6829DF07B9C70B6F8944A26DA8265E3CC434AC9BC4BDB8CC2C72A712
aaArchestrAEvents.Messages.Contract.dll	24064	072F2A1A1B78979D58AC1763F16A1C97F2B1FE70	C0356708AC4FAAE79CC7C289C405B58F018B1C748E22D027F77913922A4AAA77
aaServicesCommonDataContracts.dll	148480	D294C2018C64B4675D8432EBB1408977A6A6ACA6	FCA96023E3509A7ADB01439FF31CBCB34D844773891B6CEA4FB467674A80570E
aaServicesContractIAuthenticateASB.dll	70144	A08F275499BD4B377AD0AD9613CDED6FFDAA11A6	902E9D2973CE848EF287B1DF666876D400C02110352D38FB853D62551A908E5D
aaServicesContractIData.dll	112640	86D0CB111B6438307B00AFD80158EA734BC37123	B9399F237E5AE2506ECDF25BFF2979AA1AD05E02

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

			511F8CA4F0763060D581A8FB
agsXMPP.dll	315392	40D81446A1D5C141AA6CE23F893D308537081EAD	9DC33D2255FC7B5BD03A7D119C4CF7ECCD08277F89F7A945D55836B589251377
AjaxMin.dll	512000	0A2AFFDED0E2FFB5AFD7F81107EA0D934A6EF9E1	AB782DC0179E35CC7E3AF67F3BD052008EAEBD22DED7C0C3EC106CC92042BF29
Amqp.Net.dll	197120	FA657D438B8054369A3BEE8818D66A0875030593	7A235D6028FFE09FE54AD4CF1403A0CFEFBA8E6F6EFC6E29B27AD47C17C3B5A2
AngleSharp.Css.dll	398336	966C3CD4F991FE599B7B4EC8E847969B27371AF2	3838C7694EDD5DF466936300C411FF944FDF6CB7F6484DDDDF689CFBBEBF3146
AngleSharp.dll	833024	C863CBA483ED9974BF5C0DC9087DE8D9E8886CA2	644EA002CCD0B9734946B2F236E460BED922B2345E846144BFFFF419591D1A99
ArchestrAAlarmsAndEvents.SDK.Common.dll	16896	2293AC49164BE79945B6425339B36C082F286955	CE02A4DD8ECE7AFDAB823F205930545B79F0CF258130A67BD67BAE3495FE06A9
ArchestrAEvents.ManageEvent.Contract.dll	17920	DBDD791A914C76E86022985F5202144B51FCA529	86AC20A8CF6CC215A7602368E71B0D1C4CDF5DDC26D23EECCE370CCCDD7E1C4E
Autofac.dll	336896	71266DB05C832679738FA0FBDABB07E20E42304E	E07219EBC8B4C59A1355E40F439CE65A83A226F3D7A32A7611E6D59698BAE306
Autofac.Extensions.DependencyInjection.dll	16896	BF8847A9C064F081265A588635449C88B5D26824	C44F9BEA7A627C649E59AA51A9093BA962484A35C7A0C47CC8D056383C60FA0E
BouncyCastle.Cryptography.dll	7010304	0315325B4244BB319BCCF1584F0725ED7DAB1A3A	6E4F253CA100B0748229B9943385DB766BA0288A

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

			252F6FD3503BA4A43B2C68AF
CabDotNet.dll	49152	5742D9CBE1E59D6E72A3BC7445F4AD4CC9B7FA80	7396121B0B23E150225B765D80B5A31B368C37E3F0BD976A9BE022E7AABC2D4C
ChilkatDotNet48.dll	10966016	D7186B54672CF9FAEF0ED166E0007E1D88399138	8726EE94F8E6BD2B4C120DBC900E0817DA944F7E6055B1CA729412B72E406D22
ClosedXML.dll	1184768	FA9ACD65077DFB9885378222545B0E096BF30056	D1AFAFDFC411E1A514398FB882FD86849DD66B6E38C1CC321EC7A9CE5F3E3343
ComponentArt.Charting.WebChart.dll	1052672	5D94981410DE597851F932CF92332E13A30B53E3	2348F1646BE1E2011309D290CA45A214350D0EC5A44E62DBAA7A3E531206E42D
EasyNetQ.dll	278528	114F985D92A80A7A8A55DCA5C9D5FEFD7701250F	F89FDCE21E8658443BA0B642B6F659947726B8BA0F1D52F2BABE9EDE040084BE
HtmlSanitizer.dll	42496	6EB42038B6ABD30F933DFA9549D93484106B0C89	646C8974A4B7360E5F0BCF08F6AEDB712C4A61D678CEE15642DB7BA2B6A5ADF5
ISNet.ActiveReports.Exporting.dll	8326144	3DF0D27FF8079363ECA7DB23C2D637C90E1E4E68	CB9DD36905F108AB25D0E55D71BEB81CB4CF54D86717607BFE0F9110BFDF734B
ISNet.dll	102400	0580339F2DFED26DD11F20C50BDECE5F0B04C3AC	9926176B0BC15E454BB1C126C67B6DFC67ADA8181F22B023D8DCB16D506E9F5B
ISNet.WebUI.Design.dll	24576	BD66EAFF1FB43A1594635C62DBF422F62DEF46F5	B334FECAB5A3BCC717E7227C3AF04EDBCE2C349C54BEE24AEB35CD202CA32B40
ISNet.WebUI.dll	495616	2A9B91745275786A9DE91A37C18ABB208491C92E	D774BDB6562CBC42CD43CB82483F22EC3F968751F

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

			B8102B581EE846E0D5C44F9
ISNet.WebUI.WebGrid.dll	1236992	60B0D8726798CF83ADCBB7251FD1DE63EA6125D6	7B51CD2E750310AB1E435FA58016704C40CBF16D398B94A7B19E4DBC0CB176D5
Jint.dll	249856	3087077714C78ED7086AA6A850596FF8EE4C7262	DC48D5C612B0C0607AAE5AB021F55EF2D979B4E3823B7D3534D624DAEFA016C4
LicensingDeviceIDGenerator.dll	11264	2C473321E2401ED5D160A338D4B31694AC12661A	E57D58D6243C15A886ED47579DC650AFC1418F405BC1A73EDC1824E24B5E06D7
MailKit.dll	855552	E123C26A51447F4DEF79F4226EB687093E8116C7	1F559B2A5F98F1E77C3016E807D24AA6A32AA0B8AFFF807ECC887157CEB81D28
Microsoft.Azure.NotificationHubs.dll	408576	5ED8FCA84BC840EF83FA060DEAF8F3245A3CCDE2	9E3DB74E1B55BF404B1D8D2385404401B6312B69773823F080C5B1ED9FA3E1E5
Microsoft.Practices.EnterpriseLibrary.TransientFaultHandling.dll	25600	4EAC6E0D21AF1A67170453BB6C2D5754FCD26556	2B443098988FB5C8B59E29F760CC056C41B7CF3ED7C1533DB875C1411A79901B
Microsoft.Win32.TaskScheduler.dll	294400	F52D2E26530FDFA7CB26D48423B34F8FDFDFA253	C7A3F64DE77521E1A5DF66A05FCAD7D355BC56268A0091935D7095426730E44F
MimeKit.dll	1154048	41FC0D80C85E44C5A4154F2735F40B719B4493C1	4E79D1C9B9492C581133CCFBC70C4116FC6EB2D69A6139AA4B74B790EE37960C
Oracle.ManagedDataAccess.dll	4451328	5C66BF15FE5DFA2BB0DCE29310AEBDB6F317EE89	D3E16D3ABAFDB1E1E857C5C6F4157C54497FB4E39E19233198A49F9D663864C0
RabbitMQ.Client.dll	272896	6BD24630A49EC35970F4D8FC2903AFD7C979F719	060B693905EBE08E27BB304FB0D377F7F6C6E2F28

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

			B90BDD831FA75FC776F49FB
RadAjax.Net2.dll	241664	CFF8D0E527D993EA2DB56B1F8D146CBD8745F0E8	F31CC8264FCB00B9E125CE77B42C5E94C42046187EEF444407D8EAB964CB4783
RadCalendar.Net2.dll	380928	B2612BA0EDC017211AF64C9DF4CF40EC6EFC9090	25EDF476946356C9D7C92D07C9F6D22CEB1F94BC2A77692E4D897CD5049080B6
RadCallback.Net2.dll	159744	95782AD803D91BB923E4006F50FCE158AFA869F5	D6D3EADD2D1992261F6F4C8413877E62FDBACEBA32363538040B565AE792CA1C
RadCombobox.Net2.dll	167936	5674A4E5A3C37820E36E204A448963F298EC5B22	9FB2CDBD8C6523E191609F3CE25BD02AB21DE57624F262D9902CCE0748ABB5AA
RadDock.Net2.dll	180224	0ABF586DC36CD9182838456E4A61E9CA9953C5EC	3EA83D60792B2BF57940E3628C2AC371D2BF58AD8F7FCA98B8BB937648391192
RadEditor.Net2.dll	1310720	770125B0C75172225EFED11911E77CF819493A63	C3A3C810F7A64FBFBACC2906AA80DCEFA367E6F6DFC57BDFE0D7EAFBEE9CDC79
RadGrid.Net2.dll	1589248	37B7096AB192A8303E35F205EAB6AADC262D9162	864C32E1B9F8CD0E80698E99417658E8991F31599E21BF89CF615576280EC3B8
RadInput.Net2.dll	274432	1C00D0DD9E901CF6471C58C679F6D169C9C76BCC	336CB4FF3EF71735ED69DF00B33DFA073B6F3CD4F3838FCBE8C1CAF8A9B945A7
RadMenu.Net2.dll	278528	47195AD5B47454C0F381E6A762BE9D83B91C6170	B82CD3F23646D3D34B8F74FFFF5F9410851D8BB8AB3505A481E28733988B9FEA
RadPanelbar.Net2.dll	241664	34AF8630C90F9B76D0A2413ED08F84EE244C030B	FB51FD8F81D0CECF2166D94C05DC951196467E43

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

			32A0BAD7C0C66BD35195C145
RadSplitter.Net2.dll	274432	DAE8E7CBE286DD7A23E3A199AF1B5F27D45CD06B	2120B07A9AC803274E172D3894A80C34C2B92282C038357DFA6AF08C97153129
RadTabStrip.Net2.dll	253952	DFF06FDB2DE72E59461BEB9D35D8E5305127E6CE	BB42D0A543EB186BC9BF767F1E26A9462794F9FFAC5907FA78F469C99DD174DE
RadToolbar.Net2.dll	147456	8C94B5C40B2AC43FAA54CA68878AE7ABFF2E2787	8D966A7DB83600EC187F13BB56620B2C1189B390E0ED1029F1834BE766DB431B
RadTreeView.Net2.dll	274432	9D43CE06C81180838BD859865BD37C863659F82B	75175BFF44FBABF56432CC20DFD0970E347D73CBE0E9A2367A4E6AD9CBE53BB8
RadWindow.Net2.dll	249856	2F8F5B20F14EB6FABAB739300D561693BC2FFB7E	F33DAC11F42401833C38764490BFB286AD4E93745CA5FC67E87613ABBE5AE78C
Rhino.Mocks.dll	317952	F8926D35C75E7CB7AF90BAD1F24498879928B628	32AE5AFAE10EFD44F04F7F670835C114894D5F410180E87B2406A423540606DC
SendGrid.dll	76288	2D9485D67DA555F3301D633C343CA734A5658A5A	5BB249608319C7324B2D0C34C87DCDBC585F70C06B2605BAE6A9FE79BB3A63F0
Skelta.Mocks.dll	71680	8BF0C1C490F7E7A7A509A3A31C01B53DA656F2C9	7FAE9873FC7B480ED7D1B1410044AA76061F2BC57954189842D5B51ED7F41BE1
StarkbankEcdsa.dll	26112	942C0CC32751B60AE68E5425ED2039F74DAB5607	7A30910E329CF28E7562953BD0E9B5C26BDBF963674298DEBC1F849B20C1984F
System.Web.dll	5292032	B4F7DA55208868B1C40EC85039894B4C09C20DA5	156C6FC82A0379B4C8B2E71D68CB07FD4A4FD590

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

			2654D2C4FD92DF0501170D69
Telerik.Web.Design.dll	2166784	B0BE4D9BB86243D13FD3682423909D212B7FDE6E	3950638EB45786C142795E9B186965D3CA245B1D925D8EE74C086CA288C34530
Telerik.Web.UI.dll	30400512	5C02CA58A668E7515E39526EBDC3AC8E6363A5B7	648336E58F1A7BAD5613B5386783BBA53B268ACC80B3B9FF3E5C32ACC60CC103
WixSharp.dll	461824	F41C40EE9F80137CF8A764413E9DD9205C3B99EB	19212EF056AD8D12A68102D80432D08D86AD9F4E37185250E96391906AE67C0E
WixSharp.Msi.dll	31744	0F8850585898C300FDEC99395EE74548FE12D242	78F7ED40E992E6577BBA5A841BFE7CC0059A56956ED82D53E740CC399C15DB55
zxing.dll	498176	0380D5BCB28BCE39053EFCC25BCFD0D3C2D75E05	EF8B7AD3511FE9658E1BD198FC5FBB83E2A68AC8A567BE9854701A06D3D000F0

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

What's new in AVEVA Work Tasks?

AVEVA Work Tasks 2023 SP1 provides enhanced features and functionalities to further support users in the creation of simple and complex use scenarios and composite applications rapidly.

The following are the new features and enhancements available in AVEVA Work Tasks 2023 SP1:

- Remote Farm Configuration.
- Passphrase migration tool to migrate passphrase and configuration settings from one machine to another.
- Supports editing the details of Scheduler Associations in workflows.
- Supports exporting and deleting list items from Custom List.
- Supports multiple Unique Environment Name (UEN) when the Workflow Gateway Objects are provisioned to different repositories.
- Added localization support for Dutch language in AVEVA Work Tasks Pro.

The following are the other enhancements for **AVEVA Work Tasks 2023 SP1** release:

- Enhanced Token refresh mechanism for workflow engine so that workflows run smoothly.
- Supports column name with spaces in Web API preview.
- Supports Fully Qualified Domain Name (FQDN) for the Active Directory DataSource field while creating Repository for Active Directory provider.
- Supports Alias Name in the domain while creating Active Directory Provider repository.
- Users can log in with the Secondary Authentication Provider when the Primary Authentication Provider is not accessible.
- Binary Serialization used for sending and receiving messages in the Enterprise Server scenario has been changed to Json Serialization.
- Supports AIM authentication to access:
  - AVEVA Work Tasks controls in InTouch HMI
  - Enterprise Console in System Platform IDE
- Supports second-level Object type parameters in the Web API.
- Supports reconfiguring Workflow Gateway Object (WGO) to reset **LoggedInUserID** to the currently logged-in user and allows that user to execute the workflow.
- Compatibility enhancement to support Workflow tab for older AOT objects.
- Safeguard against blank or incorrect object definition in Web API configuration.
- Lookups can be configured using Web APIs without adding dummy data.
- Usage of Microsoft Edge WebView2 control instead of IE Web Browser control in AVEVA Work Tasks 2023 SP1 Connectors for System Platform and AVEVA InTouch.
- Optimized Form page load.
- Enhanced Base Form with **Display Mode** as **Grid** to show the columns in proper format when present inside an Embed Form control and in the main form.
- Security fixes and third-party components updates.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2023 Releases

The key features and enhancements are available for the following release:

- AVEVA Work Tasks 2023 SP1 Release Notes

AVEVA Work Tasks 2023 SP1 Release Notes

This section provides information about important features, resolved issues, and known issues for **AVEVA Work Tasks 2023 SP1**. The product and the documentation are subject to change without notice.

Important:There are unsigned third-party components that are shipped with AVEVA Work Tasks. It is advised to verify that the File names, SHA1 and SHA256 Hash details matches with those installed on the server. For details, refer Other Components under **Acknowledgements** section in the online Help.

System Requirements

The hardware and software requirements to install AVEVA Work Tasks 2023 SP1 are as follows:

- Hardware Requirements
  - Windows Requirements
  - SQL Server Requirements
- Software Requirements
  - Operating System Requirements
  - Database Requirements
  - Browser Requirements
- Other Requirements

Hardware Requirements

Windows operating systems and SQL Server versions may impose hardware requirements that exceed the minimum requirements for AVEVA Work Tasks 2023 SP1. Refer to the following Microsoft Web sites for Windows and SQL Server hardware requirements.

Windows requirements

- Windows Server 2022 System Requirements
- Windows Server 2019 System Requirements
- Windows Server 2016 System Requirements
- Windows 11 System Requirements
- Windows 10 System Requirements

SQL Server requirements

- Hardware and Software Requirements for installing Azure SQL

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Hardware and Software Requirements for Installing SQL Server 2022
- Hardware and Software Requirements for Installing SQL Server 2019
- Hardware and Software Requirements for Installing SQL Server 2016 and 2017

Software Requirements

Operating System Requirements

AVEVA Work Tasks runs on the following 64-bit Windows Operating System versions. We recommend applying all the critical updates pertaining to the version of the server.

Software Requirements Matrix	**Windows 10***	**Windows 11****	**Windows Server (Standard or Data Center)**		
-	-	-	**2016**	**2019**	**2022**
AVEVA Work Tasks 2023 SP1	Y	Y	Y	Y	Y
AVEVA Work Tasks 2023 SP1 Extension for System Platform	Y	Y	Y	Y	Y
AVEVA Work Tasks 2023 SP1 Connector for System Platform	Y	Y	Y	Y	Y
AVEVA Work Tasks 2023 SP1 Extension for OPC UA	Y	Y	Y	Y	Y
AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook	Y	Y	Y	Y	Y

* The supported Windows 10 version includes 1903, 1909, 2004, 20H2, 21H1, 21H2, 22H2.

** The supported Windows 11 version includes 21H2, 22H2.

Database Requirements

AVEVA Work Tasks is compatible with the following SQL Server versions installed on 64-bit Windows Operating System. We recommend applying all the critical updates pertaining to the version of the SQL Server.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Software Requirements Matrix	SQL Server 2016 SP3 CU17-KB5003279 (Standard or Enterprise)	SQL Server 2017 CU 31-KB5016884 (Standard or Enterprise)	SQL Server 2019 CU 21-KB5025808 (Standard or Enterprise)	SQL Server 2022 CU 6-KB5027505 (Standard or Enterprise)	SQL Azure
AVEVA Work Tasks 2023 SP1	Y	Y	Y	Y	Y
AVEVA Work Tasks 2023 SP1 Extension for System Platform	Y	Y	Y	Y	Y
AVEVA Work Tasks 2023 SP1 Connector for System Platform	Y	Y	Y	Y	Y
AVEVA Work Tasks 2023 SP1 Extension for OPC UA	Y	Y	Y	Y	Y

Note:
The recommended collation for the databases is:
SQL_Latin1_General_CP1_CI_AS

The Farm and Repository databases support case sensitive collation:
SQL_Latin1_General_CP1_CS_AS

Browser Requirements

Software Requirements Matrix	Microsoft Edge 115.0.1901.200	Mozilla Firefox Quantum 116.0.2	Google Chrome 115.0.5790.171	Apple Safari for Mac 16.6
AVEVA Work Tasks 2023 SP1	Y	Y	Y	L

Legends:

Supported	Y
Limited Support Only Inbox and Forms in Apple Safari for Mac	L

Other Requirements

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Component	Requirement
Internet Information Services (IIS)	IIS must be installed and configured.
Supported Languages for Operating Systems	• US English • Chinese (Simplified) • French • German • Japanese • Portuguese (Brazilian)

Supported Operating Systems for Work Tasks Pro	• iOS
	  • iOS 14.x, iOS 15.x, iOS 16.x
	• Android
	  • Android 13, 12, 11, 10
	  • Pie
	• Windows
	  • Windows 11 Pro version 21H2 and later
	  • Windows 10 version 20H2 and later

Supported Languages for AVEVA Work Tasks Pro	• US English • Chinese (Simplified) • Dutch • French • German • Greek • Hindi • Italian • Korean • Norwegian Bokmål • Polish • Portuguese (Brazilian) • Russian • Spanish • Turkish • Vietnamese

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Product Compatibility

- **AVEVA Work Tasks 2023 SP1** with SQL Server is compatible with the following virtualization environment:
  - VMware ESXi 6.7: Virtualization with High Availability and Disaster Recovery
- **AVEVA Work Tasks 2023 SP1** is compatible with *SQL Server AlwaysOn Availability Groups* for SQL Server 2016, 2019, and 2022.

AVEVA Work Tasks 2023 SP1 is compatible and can coexist with the following connectors and extensions:

Connectors and Extensions	Compatibility	Coexistence	Can be Applied on
AVEVA Work Tasks 2023 SP1 Extension for OPC UA	Y	Y	• AVEVA Work Tasks 2023 SP1
AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook	Y	Y	• Microsoft Office Outlook 2016 (64-bit) • Microsoft Office Outlook 2019 (64-bit) • Office 365 (64-bit)

AVEVA Work Tasks 2023 SP1 Connectors for System Platform is compatible and can coexist with the following products:

Product	Version	Compatibility	Coexistence
AVEVA Recipe Management	2023	Y	Y
AVEVA Batch Management	2023	Y	Y
AVEVA Application Server	2023 P03	Y	Y
AVEVA InTouch HMI	2023 P03	Y	Y
AVEVA InTouch OMI	2023 P03	Y	Y

ISO Details

The following folders are created after you extract the ISO file of **AVEVA Work Tasks 2023 SP1**:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **AVEVA Enterprise Licensing**:
  - Installer for AVEVA Enterprise License Server and related Licensing documentation
  - Installer for AVEVA Enterprise License Manager and related Licensing documentation
- **AVEVA Work Tasks 2023 SP1**:
  - Installer for AVEVA Work Tasks 2023 SP1
  - **Addins**:
    - Installer for AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook
  - **ISSetupPrerequisites**
    - Installers for prerequisites for installing AVEVA Work Tasks 2023 SP1
  - **Server Extensions**:
    - Installer for AVEVA Work Tasks 2023 SP1 Extension for System Platform
    - Installer for AVEVA Work Tasks 2023 SP1 Extension for OPC UA
- **AVEVA Work Tasks 2023 SP1 - Documentation**:
  - WorkTasksAdminstrativeGuide
  - WorkTasksAPIScenariosGuide
  - WorkTasksConnectorsExtensionsGuide
  - WorkTasksFormsGuide
  - WorkTasksQuickflowGuide
  - WorkTasksUserGuide
  - WorkTasksAPIReferenceGuide
- **AVEVA Work Tasks 2023 SP1 – Connectors**: Installation executable for AVEVA Work Tasks 2023 SP1 Connector for System Platform
- **AVEVA Work Tasks 2023 SP1 Release Notes.htm**: Release notes for the AVEVA Work Tasks 2023 SP1 release.

Installation Instructions

The following prerequisites are installed by the setup program before AVEVA Work Tasks 2023 SP1 gets installed:

- Microsoft .NET Framework 4.8
- Enable IIS Feature
- Microsoft .Net 6.0.7 - Windows Server Hosting
- VS C++ Redistributable2022
- Operations Control Logger
- Platform Common Services 7.1.0
- Common Licensing API Framework

Note:
- AVEVA Work Tasks 2023 SP1 requires a product License version 23.0 or higher.

- If you are planning to install any application on the same machine that installs an earlier version of the above mentioned prerequisites (for example, PCS 7.0), then make sure to install that application before installing AVEVA

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Work Tasks 2023 SP1.

- AVEVA Work Tasks needs .NET Core 6.0.7 or a newer version to be installed on the machine. Although the Work Tasks installer tries to install it as a prerequisite, if the .NET Core installation encounters any issues, the installation process will continue. However, it will be incumbent upon the customer to manually install .NET Core in such instances.

- While .Net 6.0.7 is being installed, if an installer screen with **Repair**, **Uninstall**, or **Close** option appears, select **Repair** option to proceed installation. This will restart the machine after repair and then the installation setup resumes.

- During AVEVA Work Tasks installation, if installation of **VS C++ Redistributable2022**, **Operations Control Logger**, **Platform Common Service**, and **Common Licensing API Framework** prompt to restart the machine, you can ignore and do a restart after completing installation.

Extract the ISO file of AVEVA Work Tasks 2023 SP1 to get the executable files. Run the following executable files to install and use AVEVA Work Tasks 2023 SP1.

1. **To install AVEVA Work Tasks 2023 SP1**

   Run *AVEVA Work Tasks 2023 SP1.exe* located in **AVEVA Work Tasks 2023 SP1** folder.

   For detailed installation instructions, refer to *WorkTasksAdministrativeGuide* located at **AVEVA Work Tasks 2023 SP1 - Documentation** folder.

2. **To install AVEVA Work Tasks 2023 SP1 Connector for System Platform**

   Run *AVEVA Work Tasks 2023 SP1 Connector for System Platform.exe* located in **AVEVA Work Tasks 2023 SP1 - Connectors** folder.

   For detailed installation instructions, refer to *Installing AVEVA Work Tasks Connectors for System Platform* in the *WorkTasksConnectorsExtensionsGuide* located at **AVEVA Work Tasks 2023 SP1 - Documentation** folder.

3. **To install AVEVA Work Tasks 2023 SP1 Extension for System Platform**

   Run *AVEVA Work Tasks 2023 SP1 Extension for System Platform.exe* located in **AVEVA Work Tasks 2023 SP1\Server Extensions** folder.

   For detailed installation instructions, refer to *Installing AVEVA Work Tasks Extension for System Platform* in the *WorkTasksConnectorsExtensionsGuide* located at **AVEVA Work Tasks 2023 SP1 - Documentation** folder.

4. **To install AVEVA Work Tasks 2023 SP1 Extension for OPC UA**

   Run *AVEVA Work Tasks 2023 SP1 Extension for OPC UA.exe* located in **AVEVA Work Tasks 2023 SP1\Server Extensions**.

   For detailed installation instructions, refer to *Installing AVEVA Work Tasks Extension for OPC UA* in the *WorkTasksConnectorsExtensionsGuide* located at **AVEVA Work Tasks 2023 SP1 - Documentation** folder.

5. **To install AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook**

   Run *AVEVA Work Tasks 2023 SP1 Outlook Add-in.exe* located in **AVEVA Work Tasks 2023 SP1\ Addins** folder.

   Note: *AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook* is supported for Microsoft Office 2016, 2019, and Office 365. Based on the installed version of Microsoft Office, the respective Add-in components will be installed on the system.

   For detailed installation instructions, see *Installing AVEVA Work Tasks Add-in for Microsoft Outlook* in the *WorkTasksAdministrativeGuide* located at **AVEVA Work Tasks 2023 SP1 - Documentation** folder.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Upgrading to AVEVA Work Tasks 2023 SP1

Upgrading Scenarios

You can upgrade to AVEVA Work Tasks 2023 SP1 from the AVEVA Work Tasks 2023.

However, if you want to upgrade from versions earlier than AVEVA Work Tasks 2023, you must first upgrade to AVEVA Work Tasks 2023, and then to AVEVA Work Tasks 2023 SP1.

The upgrade will overwrite all files. Therefore, we recommend that you back up the files before upgrading the product.

Make sure that the user has appropriate rights for installing and upgrading software on the machine.

Note:
- Customizations to files in Locales, Web, and WorkflowElements folders need to be applied again to the newly created files after an upgrade.

Steps to Upgrade

1. Run *AVEVA Work Tasks 2023 SP1.exe* located in **AVEVA Work Tasks 2023 SP1** folder.

   The installer splash screen, and then the Prerequisites installation screen appears.
2. Install the Prerequisites.
3. Click **Next** after installing the prerequisites.
4. Click **Finish** on the InstallShield Wizard Completed dialog box.

   The system prompts you for the restart.
5. Click **OK** if you want to restart the machine.
6. The **Upgrade Manager** automatically gets launched and performs the following tasks:
   - Stops the services and Application Pools.
   - Deletes the **WorkTasksTemp** and **Work Tasks Cache** files from the machine.

     For more information on how to run the Upgrade Manager, see Run Upgrade Manager in the *WorkTasksAdministrativeGuide*.

     Notes:
     - During AVEVA Work Tasks upgrade, if **Platform Common Service** prompt to restart the machine, you can ignore and do a restart after completing installation. If you opt to restart the machine, the installation does not resume after a restart. You will have to run the installer to continue the installation.

     - After the upgrade completes, the Work Tasks services and Application Pools start automatically.

     - It is recommended to clear the browser cache on the machines where Enterprise Console site is accessed to get the (js, css, images, etc. files) changes to take effect after the upgrade. For more information on how to clear cache, see the Cache Refresh Settings section in the *WorkTasksAdministrativeGuide*.

Key Features

AVEVA Work Tasks 2023 SP1 provides enhanced features and functionalities to further support users in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

creation of simple and complex use scenarios and composite applications rapidly.

The following are the new features and enhancements available in AVEVA Work Tasks 2023 SP1:

- Remote Farm Configuration.
- Passphrase migration tool to migrate passphrase and configuration settings from one machine to another.
- Supports editing the details of Scheduler Associations in workflows.
- Supports exporting and deleting list items from Custom List.
- Supports multiple Unique Environment Name (UEN) when the Workflow Gateway Objects are provisioned to different repositories.
- Added localization support for Dutch language in AVEVA Work Tasks Pro.

General Enhancements

The following are the other enhancements for **AVEVA Work Tasks 2023 SP1** release:

- Enhanced Token refresh mechanism for workflow engine so that workflows run smoothly.
- Supports column name with spaces in Web API preview.
- Supports Fully Qualified Domain Name (FQDN) for the Active Directory DataSource field while creating Repository for Active Directory provider.
- Supports Alias Name in the domain while creating Active Directory Provider repository.
- Users can log in with the Secondary Authentication Provider when the Primary Authentication Provider is not accessible.
- Binary Serialization used for sending and receiving messages in the Enterprise Server scenario has been changed to Json Serialization.
- Supports AIM authentication to access:
  - AVEVA Work Tasks controls in InTouch HMI
  - Enterprise Console in System Platform IDE
- Supports second-level Object type parameters in the Web API.
- Supports reconfiguring Workflow Gateway Object (WGO) to reset **LoggedInUserID** to the currently logged-in user and allows that user to execute the workflow.
- Compatibility enhancement to support Workflow tab for older AOT objects.
- Safeguard against blank or incorrect object definition in Web API configuration.
- Lookups can be configured using Web APIs without adding dummy data.
- Usage of Microsoft Edge WebView2 control instead of IE Web Browser control in AVEVA Work Tasks 2023 SP1 Connectors for System Platform and AVEVA InTouch.
- Optimized Form page load.
- Enhanced Base Form with **Display Mode** as **Grid** to show the columns in proper format when present inside an Embed Form control and in the main form.
- Security fixes and third-party components updates.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Issues Resolved

The following are improvements and resolved issues in the AVEVA Work Tasks 2023 SP1 release:

- The LDAP Test Connection fails while configuring repository with Active Directory provider when Username contains the hyphen (-) character
- Unable to insert decimal value through Database activity when AVEVA Work Tasks is running on French Operating System.
- The character (\) in the cookie name used for Client Connection License causes issue in the custom application rendering.
- When the **Invoke Workflow** button is clicked several times within a second, then multiple executions of workflows occur rather than single execution.
- If you have a parent folder having two child folders (say A and B) consisting of forms and you create a package that includes child folder A having the form and its associated artifacts. When you try to package the parent folder to the existing package, the child folder B and its child artifacts gets missing in the package.
- The packages created for versions earlier than AVEVA Work Tasks 2023 SP1 has following limitation:

  The standard widgets like Base Grid and Hierarchy Selector get packaged and when package is uninstalled, the widgets also get uninstalled with files associated with them.
- The inherited events from derived template instances in System Platform are not displayed.
- Sometimes import fails for Forms that contain more number of Invoke Workflow controls.
- Auto Logoff does not happen when a timer-based refresh grid script is present in the form.
- Data Grid page information appears arbitrarily.
- When you log in to Work Tasks Pro app with an invalid username, the license is acquired.
- The Inbox Workitems do not reflect the Iterator activity based context values for the local variables in the AVEVA Work Tasks Pro App.

Known Issues

You might come across the following issues when you work with AVEVA Work Tasks:

- When a workflow is scheduled to trigger after a specific time duration and the system experiences downtime, the system may create more workflow instances than expected once it resumes.
- Prolonged use of the Web Logger results in CPU utilization spikes. Immediate closure is crucial to restore CPU to normal levels and maintain application stability.
- Entering GET as instance name is not supported while designing a workflow in Process Designer or while creating Forms and Lists.
- Work Tasks supports OAuth-based authentication for POP3 and SMTP emails. A valid email address must be used for POP3 or SMTP configuration.
- Entering or editing values in **Rich Text** control is not supported in **Mozilla Firefox** browser.
- A colon character (:) is not supported for the Activity Name while designing a workflow in Process Designer.
- Role condition set through Filter Condition of To property in the Human Activity is not supported for CONNECT user provider.
- When installing a package, if Folder A already exists in the Forms List of the target environment and the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

package contains a Forms List with the same folder name (i.e., Folder A), the system logs the following message in the package history:

Folder title is not unique, or it contains special characters like \\,/, #&[]; ...*

- When Quickflow application is integrated with AVEVA Work Tasks, the enabling and disabling persistence of exceptions and warnings into the database requires a restart of the **Quickflow Status Monitor** service to apply the changes.
- RFID scan is not supported for Work Tasks Pro Universal Windows Platform (UWP) app.
- Wonderware Skelta BPM 2017 server version is not compatible with the AVEVA Work Tasks Pro application.

AVEVA Work Tasks Pro

The AVEVA Work Tasks Pro release notes details the version and resolved issues introduced in the release.

Version 8.2.302

The following are the enhancements and resolved issues delivered as part of this release.

Enhancements:

- Support for Modern JavaScript - ES2021 ECMAScript standard.
- Support for a Redundant Single-Sign-On (RSSO) server in Work Tasks enables seamless user authentication, specifically for Active Directory with Token Service.

Resolved Issues:

- Barcode scanning cannot be performed in landscape mode.
- Unable to upload attachments when the Encryption Security Mode is set for the Attachment control.

Version 8.1.214

The below resolved issue is delivered as part of this release.

- Fix for the login to the provider that displays a loading icon on the iPad.

Version 8.1.212

Note: This note is applicable for versions 8.1.212 and onwards.
- Select the latest AVEVA Work Tasks Pro app only if you have either **AVEVA Work Tasks 2020 Update 2** with **Hotfix 7.2.3020.12** or A**VEVA Work Tasks 2023 SP1** with **Hotfix 8.1.2030.4**. Users with other versions should not choose the latest app. This is applicable only for Android.
- To ensure push notifications work properly after applying the hotfix and upgrading the AVEVA Work Tasks Pro app, log out and log back in. If notifications still don't appear, restart your device and/or uninstall and reinstall the app.

The following are the resolved issues delivered as part of this release.

Resolved Issues:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Migrated the legacy FCM APIs to the FCM v1 APIs to ensure continued functionality of AVEVA Work Tasks and maintain notification message delivery despite the deprecation of the legacy FCM APIs.
- Resolved an issue where the AVEVA Work Tasks Pro application would crash when launching the RFID scanner on Android 13 devices. Users can now scan NFC tags seamlessly, allowing them to perform RFID scanner features without interruptions.
- Resolved an issue where users were unable to upload attachments when the Encryption Security Mode was set for the Attachment control. Users can now successfully attach photos in the Attachment control with Encryption Security Mode enabled, ensuring smooth use in their business processes.
- Enhanced overall security measures.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Administrator Guide

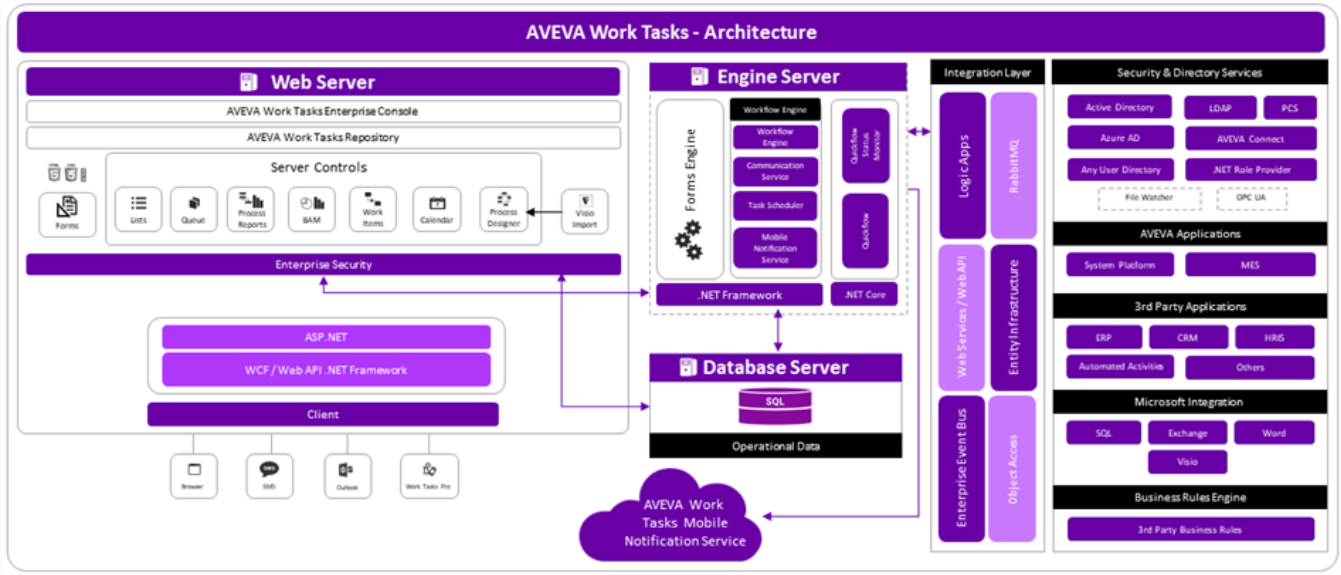
This section is for **AVEVA Work Tasks** users. It contains information you need to manage the domain and security and assumes that you have basic working knowledge of **AVEVA Work Tasks**.

Architecture

The **AVEVA Work Tasks** architecture helps you understand the various features of the product. It helps you to effectively plan and deploy the product in your environment.

Structural Parts

The following image shows the AVEVA Work Tasks Architecture:



The following table describes the different structural components of **AVEVA Work Tasks**:

Architecture

Structural Components	Description
Enterprise Console	This is a Web-based interface that allows authorized users to build workflow driven applications, and to participate in these applications. Organizations can create and access their business applications using Enterprise Console.
Repository	A Repository is a container that holds Business Process Management (BPM) and application-related data. It

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Structural Components	Description
	has a unique name and description. The Repositories are stored within the Database. You can also maintain each Repository on a different Database and isolate the information from each other. AVEVA Work Tasks typically requires that you create one repository for an organization. However, sometimes there may be a need for more than one Repository. For example, you may have separate development and production environments or have a need to completely isolate applications from each other. One more compelling scenario for creating more than one Repository is that of hosted solutions, where each client needs a secure data store and isolation. In such a case, you will need to create multiple Repositories. Repositories are maintained using Enterprise Console.
Server Controls	Server Controls are user interface layers provided by AVEVA Work Tasks. These easy-to-use controls can be reused and embedded in a custom ASP.NET application. You can extend the functionality of AVEVA Work Tasks in your application by simply dragging and dropping these controls onto the ASP.NET page. These are required only for custom ASP.NET development where a context-sensitive workflow UI has to be embedded inside the application. The following controls are available in Server Controls: • **Workflow List**: This lists all workflows defined for a AVEVA Work Tasks Repository. The Workflow List provides options for categorizing, designing, and publishing workflows. • **Process Designer**: The Process Designer is an interface used to model processes. Similar to other workflow designing tools, like Microsoft Visio, it includes popular drag and drop functions that enable you to quickly and easily draw workflows for your business processes. For example, you can use Process Designer to design a process for support request. This process involves creation of workflows and defining business logic. • Queue Management: The Queue Management enables you to create and manage queues for allocating workflow activities. This is done using

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Structural Components	Description
	the Queue control. A queue consists of a pool of resources with shared characteristics. The Queue control offers advanced level resource optimization. Queue Management supports industry standard algorithms for automatic task allocation of resources belonging to a common pool. • **Calendar**: The Calendar interface enables the creation of hierarchical organization and resource calendars. The Calendars are used to determine availability of resources, calculate time outs and perform other scheduling functions. • **Business Activity Monitoring (BAM) Dashboard**: The BAM Dashboard is a key Server Control. It enables you to monitor the execution of a workflow and provides an array of real-time reports. This report includes status and performance of workflows. It is also instrumental in generating alerts and events, providing reports in graphical format, ensuring secured access and providing details for tracking workflows. You can use Process Reports to schedule reports and generate alerts. Business managers and Process administrators can monitor, track, fine tune processes, and make pro-active decisions. • **Work Item List**: The Work Item List is a Web-based interface which lists the various workflow activities in AVEVA Work Tasks. The Work Item List provides detailed information of an activity. You can view and respond to Work items from an interface similar to Microsoft Outlook.
Forms Designer	The Forms Designer is an intermediate layer between a user and Forms Engine. It provides an interface that assists you to design, customize, maintain form templates and define a business logic specific to a form. Forms are associated with workflows. These workflows are responsible for driving end-to-end business applications. A form triggers a business process. For example, you can use Forms Designer to design a Leave Application form for your organization. The

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Structural Components	Description
	designed form can be used as a template by all the employees while submitting leave requests.
Lists	A List is a collection of items such as Task lists or Item lists captured using Forms. It acts as a logical table (similar to a database table), but involves more complicated structure to represent an entire entity with links to child tables. The data entered in the Lists are stored in a Repository. It stores application-level data thus facilitating generation of business reports. AVEVA Work Tasks lists group related information. A Form is used as an interface to enter data in the Lists. For example, you can define a List to store all the leave requests submitted by employees in your organization. Your list can include parameters (columns or fields) such as From Date, To Date, Leave Type, and Reason.
Delivery Channel	A Delivery Channel is an interface for users to interact with workflows. AVEVA Work Tasks provides multiple delivery channels which include: • **Browsers**: AVEVA Work Tasks provides controls that can be embedded in Web-based applications to view or respond to Work items. • **Messenger Services**: AVEVA Work Tasks uses this channel to send tasks or information to messenger applications. • **Mobile Phones**: AVEVA Work Tasks allows both SMS and e-mail communication. You can send and receive messages using AVEVA Work Tasks. The two-way e-mail channel within AVEVA Work Tasks is used to deliver and act on tasks directly using mobile phones. • **E-mail**: AVEVA Work Tasks also supports two-way e-mail communication. A Work Item can be delivered to both email and Web channel based on a user's preference. The user can act on the work item from either of the channels. AVEVA Work Tasks synchronizes this information. • **Microsoft Outlook Applications**: AVEVA Work Tasks provides a Microsoft Outlook plug-in to respond to tasks. It enables work items to be delivered to the Inbox of your mail client such as

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Structural Components	Description
	Microsoft Office Outlook 2016 or Microsoft Office Outlook 2019. For more information, see Microsoft Outlook Integration. • **Desktop Applications**: AVEVA Work Tasks provides extensive APIs and Web services that can be called from various desktop applications to work with Work item.
Forms Engine	The Forms Engine is responsible for every activity starting from the time a form is opened till a form is closed. It captures all the real-time user inputs, processes the inputs, and displays this information in terms of controls, business logic, and usability. The Forms Engine is triggered when a form is submitted. The submitted form will follow a workflow based on the business logic defined for the form. For example, the Forms Engine makes the Leave Application form available to an employee when the employee wants to apply for a leave.
Services	The Server controls various service requests using Services. The Services comprises the following: • **Workflow Engine Service**: This service needs to be running for the execution and control of workflows. • **Task Scheduler Service**: This service handles the execution of all tasks that have not taken place as scheduled (escalations). • **Communication Service**: This service handles incoming e-mail and SMS communication. • **Advance Server Service**: This service provides load balancing and failover support. This service is available only if Load Balanced Server is installed. • **Client Service**: This is used internally by AVEVA Work Tasks for provisioning and deployment. • **Mobile Notification Service**: This service is used to send notifications to the Work Tasks Pro mobile application. The Workflow Engine Service can exist on a single machine or on multiple machines. For Load Balancing and Failover support to function,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Structural Components	Description
	Workflow Engine Services must be installed on multiple machines and Advance Server Service must also be enabled on these machines. For more information on Load Balancing and Failover support, see Load Balanced Server.
Operational Data	The data created while working with AVEVA Work Tasks are stored in SQL databases. These databases are defined while installing the solution. The Server Controls, Delivery Channel, Services, and Forms Engine connect to the database to store or retrieve data.
Integration Layer	The integration layer acts as an interface between AVEVA Work Tasks and third-party applications. The various objects within the integration layer include: • Entity Infrastructure: Entity Infrastructure is used to manage users across different platforms. An Entity has two categories - User Entity and Role Entity. A User Entity manages all the information for the users across different data sources. A data source can be an active directory, an SQL provider, or more. A Role entity manages users in roles. • Web Services: AVEVA Work Tasks provides out-of-the-box activities that can be used to call a third-party Web Service from a workflow. • AVEVA Work Tasks Object Access (SOA): This object allows reuse of third-party .NET objects in AVEVA Work Tasks, thus avoiding custom coding. AVEVA Work Tasks allows registration of third-party object assemblies. Example: An organization has a Human Resources (HR) system developed using Microsoft.NET technology. This HR system has information related to employee, such as salary, leave, promotions. To strengthen the process, the organization integrates its existing process with AVEVA Work Tasks. While designing a workflow for the appraisal process using AVEVA Work Tasks, there is a need to get employee information and accordingly route the workflow based on the experience level and the grade of an employee. The need is to reuse the existing employee object

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Structural Components	Description
	inside the AVEVA Work Tasks process and avoid custom coding. This objective is achieved by using AVEVA Work Tasks Object Access. • Enterprise Event Bus: This is a mechanism that handles third-party events and these events are used to trigger a workflow. The third-party events can include updating files and receiving e-mails. Example: An organization has implemented AVEVA Work Tasks to automate their invoicing process. The organization wants to monitor changes made to a folder that stores scanned copies of invoices as files. As soon as any modification is made to the folder, the workflow triggers the accounts payable process and a notification is sent to the respective person. Modifications to a folder can include activities, such as adding an invoice, updating an invoice, renaming an invoice, or deleting an invoice. All these objects work collectively to ensure smooth information exchange.
Security and Directory Services	AVEVA Work Tasks does not force you to create new user accounts and passwords. Instead, it assumes that you already have some means to authenticate a user name and password. This allows AVEVA Work Tasks to use the existing user name and password, without creating replicas of those user accounts in specific tables. The solution thus supports external Security and Directory Services. This is a service that allows AVEVA Work Tasks to use and manage a user's identity stored in an external database. AVEVA Work Tasks supports the following Security and Directory Services: • Active Directory • Lightweight Directory Access Protocol (LDAP) • Any User Directory • .NET Role Provider **Note**: You can create user accounts using AVEVA Work Tasks. These user accounts are stored in User List.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Structural Components	Description
File Watcher	A File Watcher constantly monitors activities in a folder. Whenever there is an activity, such as a file creation or a file update, File Watcher raises events that are used to initiate a workflow. Example: An organization has implemented AVEVA Work Tasks solution to automate their invoicing process. The organization wants to monitor changes made to a folder that stores scanned copies of invoices as files. The File Watcher feature monitors all changes made to the folder. As soon as any modification is made to the folder the workflow triggers the accounts payable process and a notification is sent to the respective person. Modifications to a folder can include activities like creating or adding a new invoice, updating an existing invoice, renaming an invoice, or deleting an invoice.
Third-party Applications	AVEVA Work Tasks can integrate with different third-party applications, such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), and Human Resources Information System (HRIS) using the integration components, such as Web service, custom activities, and API.
Microsoft Integration	The open architecture of AVEVA Work Tasks assists you to derive maximum benefits by integrating it with various Microsoft applications. The Microsoft applications that are integrated comprise: • **Microsoft Exchange**: AVEVA Work Tasks integrates with Microsoft Exchange to provide alerts and notifications. • **Microsoft Word**: AVEVA Work Tasks integrates with Microsoft Office applications like Microsoft Word to participate in content-driven business processes. • **SQL**: SQL integrates with AVEVA Work Tasks to store records. • **Microsoft Visio**: AVEVA Work Tasks can integrate with Microsoft Visio and import diagrams created using Microsoft Visio.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Structural Components	Description
Business Rules Engine	For common process related scenarios, AVEVA Work Tasks provides basic rules capabilities using rule activity, decision activity, and expression builder. You can also integrate AVEVA Work Tasks with other third-party business rules engine using the integration components, such as Web services, Custom activities, and Event API.

Core Components

The core components of AVEVA Work Tasks are as follows:

Server

A Server is a machine on which the Client Service, Advance Server Service, Communication Service, Task Scheduler, Mobile Notification Service, and Workflow Engine are installed. If the load balancing feature is enabled during the Farm configuration, the Load Balanced Server License is acquired and the Advance Server service is installed and enabled. These services can be installed on a single server or on multiple servers. In the case of Developer and Professional Editions, all services are installed on the same server, whereas in the case of Load Balanced Server, the services are installed on multiple servers.

The primary role of the Server is to control workflow execution and schedule tasks. The Workflow Engine and the Task Scheduler are services that control workflow execution. The Communication service manages communication and notifications. The Advance Server service enables load balancing.

Client

A Client is a machine in which the Central Configuration and the Enterprise Console are enabled. If the load balancing feature is enabled during the Farm configuration, the Load Balanced Server License is acquired and the Advance Server service is installed and enabled.

Farm Database

A Farm Database is a machine that contains information about the Enterprise Console, location of the Central Configuration, and configuration settings at a global and server (machine) level. These settings are specified in the Farm Configuration Wizard database.

Central Configuration

The Central Configuration enables you to define, create and manage repositories for workflows and execute them. A Repository refers to a database that contains workflow definitions, queues, calendars, settings, and workflow execution data for an application.

The location and settings for the Central Configuration are specified from the Farm Configuration Wizard. The

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Repository settings are stored in the database specified for the repository.

Enterprise Console

This is a Web-based interface that allows authorized users to build workflow driven applications and also allows users to participate in these applications. The console enables central administration of all workflows associated with a repository. Activities that can be performed from the Enterprise Console include workflow categorization, workflow design, associating workflows with lists, events, initiating workflows for new workflow creation, publishing workflows, creation, management and association of Forms with workflows and viewing reports. The Enterprise Console also enables administrators to set up and manage queues and calendars for use in workflows.

- One Enterprise Console can be mapped to one or more Repositories; however users can login only to their respective Repositories.
- One or more Enterprise Consoles can be configured for the same repository.
- The Enterprise Console can be configured for more than one repository.
- The Enterprise Console is defined and managed using the Farm Configuration Wizard.

Repository

A Repository is a database that contains workflow definitions, queues, calendars, settings, workflow execution data and Business Activity Monitoring (BAM) data. Data Sources for Repositories are defined in the Central Configuration.

A Repository can also be created for each client in a hosted environment (Application Service Provider environment).

Note: Workflows existing in two different Repositories cannot interact with each other. User Sources (also known as User Providers) are specific to a repository although multiple user providers are possible per repository. A User Provider is a user data source such as Active Directory or an application's custom data source.

Example: If you are defining workflows for your organization, the organization must be defined as a AVEVA Work Tasks Repository. Repositories are created from the Central Configuration.

Repository Data Source

A Repository Data Source is connection information specific to a repository database. AVEVA Work Tasks services use a Repository Data Source to connect to the database.

Forms

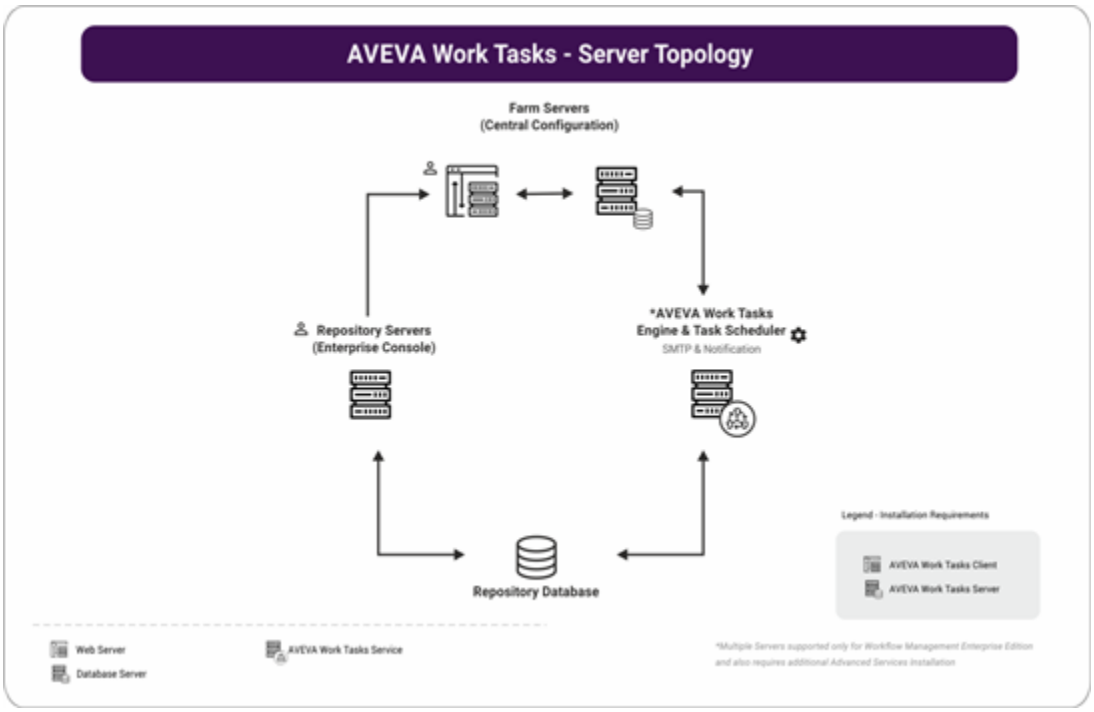
The Forms is an easy to use drag and drop interface used for designing forms. You can use forms designer to quickly design forms, maintain form templates, and define business logic. The Forms are used to trigger Workflows.

Farm

A Farm is a collection of Servers and Client services working as a single logical entity to provide the AVEVA Work Tasks solution. The Farm configuration is achieved through the Farm configuration wizard.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The following image displays the various components of AVEVA Work Tasks and how these components interface with each other.



Microsoft Outlook Integration

AVEVA Work Tasks supports two-way communication via e-mail. It provides a Delivery Channel for users to interact with workflows. Also provides multiple delivery channels which include:

Browsers: AVEVA Work Tasks provides controls that can be embedded in Web-based applications to view or respond to the work items.

Messenger Services: AVEVA Work Tasks uses this channel to send tasks or information to messenger applications.

Mobile Phones: AVEVA Work Tasks allows both SMS and e-mail communication. You can send and receive message s using AVEVA Work Tasks. The two-way e-mail channel within AVEVA Work Tasks is used to deliver and act on tasks directly using mobile phones.

E-mail: AVEVA Work Tasks also supports two-way e-mail communication. A Workitem can be delivered to both e-mail and Web channel based on a user's preference. The user can act on the Workitem from either of the channels. AVEVA Work Tasks synchronizes this information.

Microsoft Outlook applications: AVEVA Work Tasks provides a Microsoft Outlook plug-in to respond to tasks. It enables work item to be delivered to the Inbox of your mail client, such as Microsoft Office Outlook 2016 and Microsoft Outlook 2019.

Desktop Applications: AVEVA Work Tasks provides extensive APIs and Web services that can be called from various desktop applications to work with Workitem.

You can act on a workflow task from any mail client. AVEVA Work Tasks Outlook Integration is required for providing any advanced options through Microsoft Outlook.

The AVEVA Work Tasks Outlook Integration utility is a plug-in for AVEVA Work Tasks. It enables work item to be delivered to the Inbox of your mail client, such as Microsoft Office Outlook 2016 and Microsoft Outlook 2019.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

This is an alternative option provided to reply to e-mails related to your activity. This integration thus allows you to respond to work item from within Microsoft Outlook. The Outlook follow up functionality helps you to mark a mail as follow up. This functionality is used to set reminders and perform tasks at the scheduled time.

AVEVA Work Tasks Outlook integration utility automatically detects the installed Microsoft Office version and accordingly adds the plug-in. The tasks appear on the Outlook Ribbon bar. The plug-in menu options include:

- Approval
- CheckList
- Choice
- Task

Note:
- Currently rendering of AVEVA Work Tasks Forms is not supported.

System Requirements to Install Outlook Integration

The system requirements to install AVEVA Work Tasks Microsoft Outlook Integration is same as AVEVA Work Tasks.

Refer to any of the installation scenarios in the section AVEVA Work Tasks Products.

The system requirements to install AVEVA Work Tasks Outlook Integration utility are:

- Windows operating system
- Microsoft Visual Studio 2015 Tools for Office Runtime
- Microsoft Framework 3.5 and above
- Microsoft office Outlook 2016 or Microsoft Office Outlook 2019

Outlook Integration Best Practices

- If you want to add comments to **Human Activities** through **Mail Channel**, then you can manually reply to the AVEVA Work Tasks Mail Channel mails without using add-ins.
- It is a best practice to not use Outlook Add-ins if you want to use localized mail channel templates.

Outlook Integration Licensing Information

AVEVA Work Tasks Microsoft Outlook integration plug-in is distributed free with the core product license.

Scalability and Availability

This section provides information on the high availability and load on the AVEVA Work Tasks Web servers and services.

High Availability and Load on AVEVA Work Tasks Web Servers

High Availability and Load on the Services

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

High Availability and Load on the Services and Web Servers
High Availability of the Services and High Load on Web Servers

High Availability and Load on AVEVA Work Tasks Web Servers

In this scenario, installation includes a AVEVA Work Tasks Web Farm besides AVEVA Work Tasks Server, and AVEVA Work Tasks Databases. All these components are installed on different machines and the Web Farm on multiple web servers.

A Web Farm is a group of two or more Web Servers. All the web servers in the Web Farm will have the Central Configuration and the Enterprise Console installed. The Services and the Databases are installed on two different machines.

This scenario is recommended when you require a quick response from web servers under heavy load situations.

See Also: AVEVA Work Tasks Web Server Farm, Work Tasks Server, and AVEVA Work Tasks Database Server on Different Machines

High Availability and Load on the Services

In this scenario, you have a Web Server and Services installed on the same machine. You can host this set up on multiple computers to form a Web Farm with clustered the Services. The Databases are installed on a machine by itself. While scaling out, ensure that the machine has both the Web Server and the Services installed.

See: Web Server Farm with Clustered Work Tasks Services and Databases on Different Machines

In this scenario, you can maintain a Web Farm with the Web Servers and Services cluster on separate machines while the Databases are installed on a different machine. This deployment provides scalability and high availability of the Workflow Engine and the Web Servers. This is achieved by distributing the load equally among all the machines in the cluster and the Web Farm.

Multiple servers in the Web Farm or the cluster run the same services. Failure of a server does not impact other servers. You can take the affected server offline while the other servers maintain the service. This type of installation provides Failover. Thus allowing you to easily add or remove machines from the Web Farm or from the cluster without affecting the execution of workflows.

This is the optimal scenario for heavy load conditions and can be considered if you anticipate heavy load on the Workflow Engine or the Web Server.

See: Web Farm Connecting to Separate Services Cluster with Database on a Different Machine

High Availability and Load on the Services and Web Servers

In this scenario, you can maintain a Web Farm with Web Servers and a AVEVA Work Tasks Services cluster on separate machines while the Databases are installed on a different machine. This deployment provides scalability and high availability of the Workflow Engine and Web Servers. This is achieved by distributing the load equally among all the machines in the cluster and the Web Farm.

Multiple servers in the Web Farm or the cluster run the same services. Failure of a server does not impact other servers. You can take the affected server offline while the other servers maintain the service. This type of installation provides Failover. Thus allowing you to easily add or remove machines from the Web Farm or from the cluster without affecting the execution of workflows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

This is the optimal scenario for heavy load conditions and can be considered if you anticipate heavy load on the Workflow Engine or the Web Server.

See: Web Farm Connecting to Separate Services Cluster with Database on a Different Machine

In this scenario, you have a Web Server and the Services installed on the same machine. You can host this set up on multiple computers to form a Web Farm with clustered Services. The Databases are installed on a machine by itself. While scaling out, ensure that the machine has both the Web Server and the Services installed.

See: Web Server Farm with Clustered Work Tasks Services and Databases on Different Machines

High Availability of the Services and High Load on Web Servers

In this scenario, you can maintain a Web Farm with Web Servers and a AVEVA Work Tasks Services cluster on separate machines while the Databases are installed on a different machine. This deployment provides scalability and high availability of the Workflow Engine and Web Servers. This is achieved by distributing the load equally among all the machines in the cluster and the Web Farm.

Multiple servers in the Web Farm or the cluster run the same services. Failure of a server does not impact other servers. You can take the affected server offline while the other servers maintain the service. This type of installation provides Failover. Thus allowing you to easily add or remove machines from the Web Farm or from the cluster without affecting the execution of workflows.

This is the optimal scenario for heavy load conditions and can be considered if you anticipate heavy load on the Workflow Engine or the Web Server.

See: Web Farm Connecting to Separate Services Cluster with Database on a Different Machine

Setting up Intranet and Extranet Web Fronts

Usage with the Single Server

This scenario can be deployed in both the Single Server and the Load Balanced Server. In this scenario, one Web Server will be set up in a private network whereas the other Web Server will be set up in a public environment. AVEVA Work Tasks Services and the Databases are installed on two different machines.

For License related information, see About AVEVA Enterprise Licensing.

You might consider this deployment to configure multiple servers to host the same content accessible to a private network of your organization as well as to a secured, external Web Site.

Role	Components Installed	Remarks

AVEVA Work Tasks Server	• Services
	  • Client Service
	  • Advance Server Service (Disabled)
	  • Mobile Notification Service
	  • Communication Service
	  • Task Scheduler
		-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

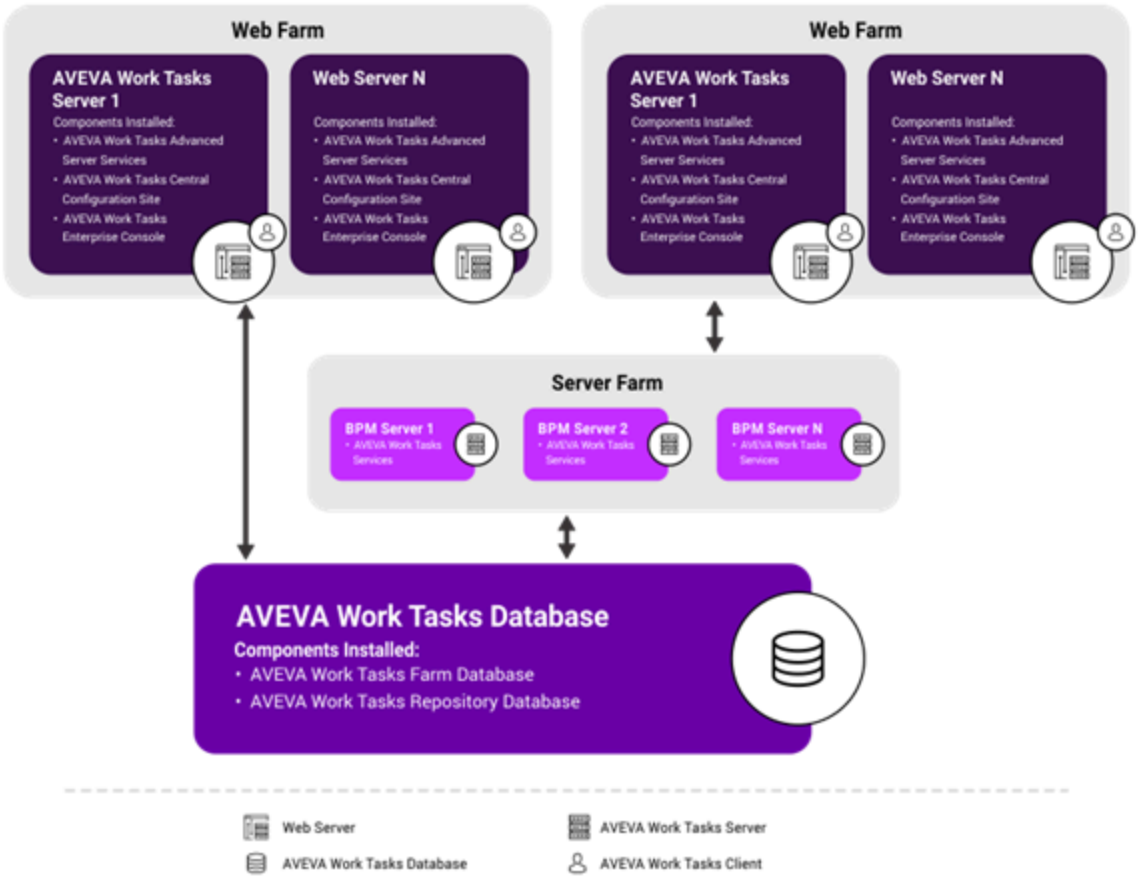
Role	Components Installed	Remarks
	• Workflow Engine	

Database Server	• Databases
	  • Farm Database
	  • Repository Database
		-

Web Farm	• Central Configuration • Enterprise Console	Web Server 1 is setup in a private network.
Web Farm	• Central Configuration • Enterprise Console	Web Server 2 is setup in a public network.

System Landscape

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



For more information about installation instructions for this scenario, see Intranet and Extranet Web Servers on Different Machines.

Usage with the Enterprise Editions

In this scenario, one Web Server will be set up in a private network whereas the other Web Server will be set up in a public environment. The AVEVA Work Tasks Server Components and Databases will be on two different machines.

For License related information, see About AVEVA Enterprise Licensing.

You might consider this deployment to configure multiple servers to host the same content accessible to a private network of your organization as well as to a secured, external Web Site.

Role	Components Installed	Remarks

AVEVA Work Tasks Server	• Services
	  • Client Service
	  • Advance Server Service
		-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

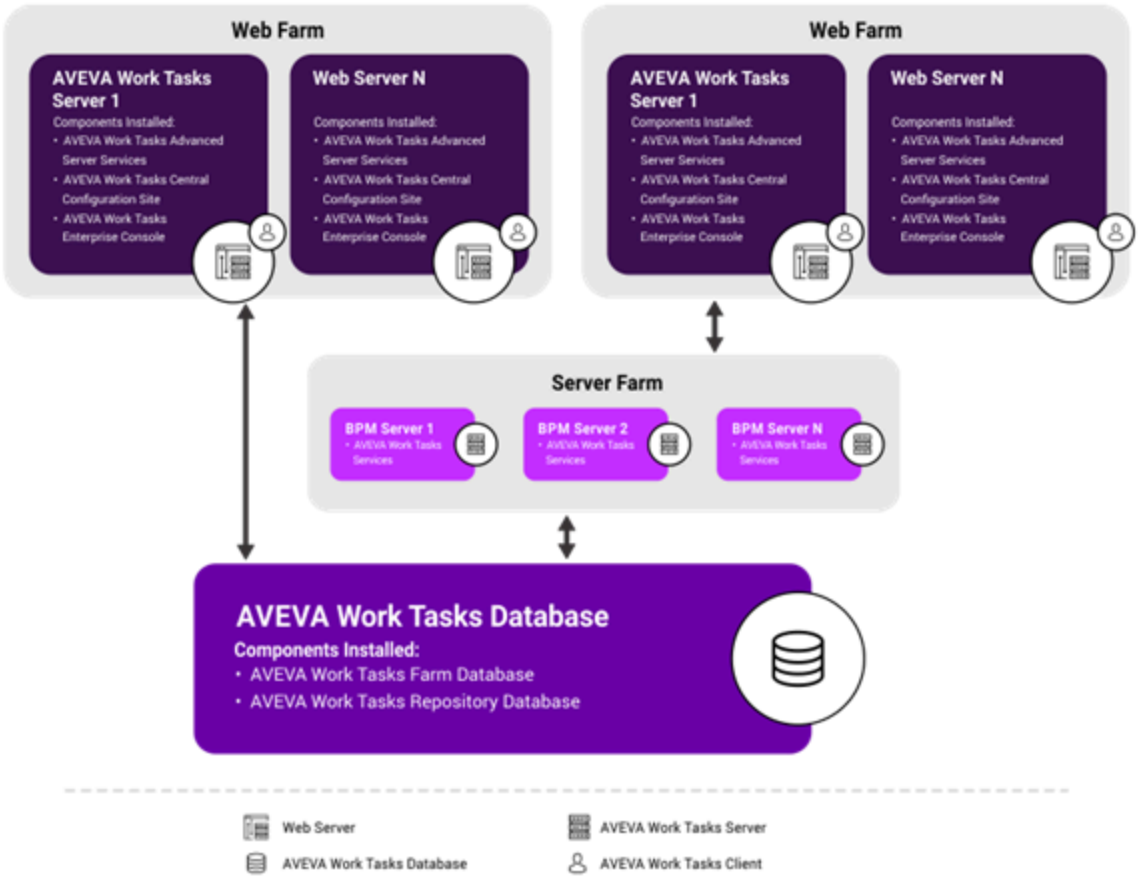
Role	Components Installed	Remarks
	• Mobile Notification Service • Communication Service • Task Scheduler • Workflow Engine	

Database Server	• Databases
	  • Farm Database
	  • Repository Database
		-

Web Server	• Central Configuration • Enterprise Console • Advance Server Service	Web servers will be setup in a private and public network.

System Landscape

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



For more information about installation instructions for this scenario, see Intranet and Extranet Web Servers on Different Machines.

Partitioning Database

The database partition feature can be enabled while creating or upgrading a SQL Server data source. Therefore, for multiple repositories based on the same data source, all the repositories will be partitioned. Ideally, each repository should be in a separate data source to avoid partitioning of the same.

Overview

AVEVA Work Tasks stores application data in Microsoft SQL Server. This data is very important as it is required for analytical or legal purposes. However, keeping all data in transactional database increases the size of the database enormously. Although, the latest relational database management systems and allied hardware are optimized to handle such data, it still might affect the overall performance of the business application.

AVEVA Work Tasks can be used across different business domains. Business requirements may require workflows to run continuously, and thus increases the size of some of the database tables. These database tables are critical for optimal performance of the workflow engine, and further to the performance of the business application.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Microsoft SQL Server supports table partitions to move completed transactions to a different partition. The active transactions can be kept in one partition and the completed transactions can be moved to a different partition. This optimizes workflow engine performance as queries are run on the lean partition. Database tables that grow based on workflow instances are considered for partition on completion of workflow instances, the data is moved to the secondary database partition.

Partition Feature

The partition feature can be enabled while creating or upgrading a SQL Server data source. The partition feature is at the database level. Therefore, for multiple repositories based on the same data source, all the repositories will be partitioned. Ideally, each repository should be in a separate data source to avoid partitioning of the same.

Note:
- The partition feature is available only on Microsoft SQL server versions and editions that support table partition feature. Refer to the Microsoft documentation to check the availability of the partition feature.
- The partitioning of the database is recommended only for those business applications that run large number of workflow instances.
- Enabling and disabling partition is a maintenance activity which requires closing the application. So, plan it in advance before you enable or disable partition.
- Enabling and disabling partition must be done with precaution, considering the following points:
- Enable or disable partition under the supervision of database expert, preferably a Database Administrator (DBA)
- Take the Database backup to avoid any eventuality
- Stop the Workflow services and Web application before enabling or disabling partition

For a data source, partition can be enabled either from the **DataSource** settings in the **Central Configuration** or from the **AVEVAWorkTasksPartitionManager.exe** (Partition Manager tool).

Partitioning with Central Configuration

Enabling Partition

For a data source, partition can be enabled from the **DataSource** settings in the **Central Configuration as follows:**

1. Enable partitioning by selecting a data source, and then selecting **Enable Table Partitioning**.
2. Enter appropriate path for **Database Secondary Data File Path**.

Database Secondary Data File Path

Ideally, the secondary database file should be on a separate file group on a different drive for optimal performance. That benefits in keeping the least used partition in a different path.

The database secondary data file path should be on a Microsoft SQL Server machine with sufficient rights to create files on the specified path. Ensure, you provide a valid path as it is not validated by the business application. If the path is not specified, then the secondary database file will be created at the default path, where the primary database file exists.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Enabling Partition on an Existing Data Source

For an existing data source, partition can be enabled from the **DataSource** settings in the **Central Configuration** as follows:

1. Enable partitioning by selecting an existing data source, and then selecting **Enable Table Partitioning.**
2. Enter appropriate path for **Database Secondary Data File Path**.

For more information on **Database Secondary Data File Path**, see *Database Secondary Data File Path*.

Partitioning with Partition Manager

Partition Manager (**AVEVAWorkTasksPartitionManager.exe**) is a Windows application available in the installation path. This application allows you to enable partition on an existing non-partitioned SQL Server data source, and to move data to the secondary database partition.

We recommend enabling or disabling partition only when no users are logged on to the system. That is perform partition activities during a scheduled maintenance lock out period. Also, ensure to restart the system after performing partition activities.

Enabling Partition

For an existing data source, partition can be enabled from the **Partition Manager** tool as follows:

1. Select an existing data source.
2. Enter appropriate path for **Database Secondary Data File Path**.
3. Click **Enable Table Partitioning**.
4. Click **Yes** to move data automatically or click **No** to move data later manually.

For more information on **Database Secondary Data File Path**, see *Database Secondary Data File Path*.

Moving Data

For an existing data source, after partition is enabled, data can be moved using the **AVEVAWorkTasksPartitionManager.exe** as follows:

1. Select an existing data source.
2. Right-click on the data source, and then click **Move Data**.

Note: Moving data to the secondary database partition consumes time and resources on the server. Therefore, ensure you move data only when there is no load on the server.

Disabling Partition

For an existing data source, after partition is enabled, the partition can be removed using the **AVEVAWorkTasksPartitionManager.exe** as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Select an existing data source.
2. Click **Disable Table Partitioning**.

Note:
- Disabling database partition may take some time based on the size of the database.
- Disabling removes the archived file from SQL Server, but by default SQL Server does not remove the associated file until a log backup is performed. Therefore, ensure to perform a transaction log backup.

If **AVEVAWorkTasksPartitionManager.exe** fails to disable the partition, then follow these steps:

1. Open **Microsoft SQL Server Management Studio**.
2. Create a new query for the data source.
3. Run `[RemoveSkeltaPartition]` stored procedure using `Exec [RemoveSkeltaPartition]`SQL script.
4. Use the Farm database and update the **IsPartitioned** column of the **SKDataSource** table. Use the following SQL query by providing Data source name.

   ```
   Update [SKDataSource]
   set IsPartitioned = 0
   Where Name = '< Data Source Name>'
   ```

5. Check the **ARCHIVED files** and **ARCHIVED scheme** of **Files** and **Filegroups** under Database Properties to verify that the partition is completely removed. The system should not show the **ARCHIVED file** and **ARCHIVED scheme** under **Filegroups**.
6. If the above items are shown, manually delete them using SSMS:
   - Select `[<Database Name>_ARC.NDF]` from **Files** and then click **Remove**.
   - Select **ARCHIVED scheme** from the **Filegroups** and then click **Remove**.

   OR

   Execute the below script:

   ```
   ALTER DATABASE [<Database Name>] MODIFY FILEGROUP [PRIMARY] DEFAULT
   ALTER DATABASE [<Database Name>] REMOVE FILE [<Database Name>_ARC.NDF]
   ALTER DATABASE [<Database Name>] REMOVE FILEGROUP ARCHIVED
   ```

7. Take the transaction log backup.

Verifying Partition

The following steps will help you to verify the status of Enabling/Disabling of partition.

To verify the enabled partition status

1. Open **SQL Server Management Studio**.
2. Select the **Database** of **AVEVA Data source**.
3. Right-click the **Menu** and select **Properties** option.

   When the partition is enabled, you will see the following items in the **Database Properties** of the system:
   - The **ARCHIVED file** in **Files** page
   - The **ARCHIVED scheme** in **Filegroups** page

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- File with *<DataSource Name _ARC.ndf>* under the secondary selected folder path

To verify the disabled or removed partition status

1. Open **SQL Server Management Studio**.
2. Select the **Database** of **Work Tasks Data source**.
3. Right-click the **Menu** and select **Properties** option.

   When the partition is disabled or removed, you will not see the following items in the **Database Properties** of the system:

   - The **ARCHIVED file** in **Files** page
   - The **ARCHIVED scheme** in **Filegroups** page
4. After disabling partition, run the following query and see if the file still exist in the database:

   ```
   SELECT * FROM sys.master_files WHERE name = '<DatabaseName>_ARC'
   ```

   If the file still exists, take a transaction log backup before you proceed to enable partition.

Maintenance Activities

The following maintenance activities are recommended before and after partitioning the table. These activities may take some time (30 minutes to a few hours) based on the size of the database.

Before partitioning

- Backup the database.

After Partitioning

- Improve overall performance by optimizing the database as follows:

1. Shrink the primary database file.
2. Shrink the log file.
3. Rebuild indexes.

Backup and Restoring Partitioned Database

Backup and restoring steps for partitioned database is slightly different from the standard database backup and restoration. Please follow the Microsoft documentation for details.

Following references may help for backup and restoration of Database with Partition:

https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/restore-and-recovery-overview-sql-server?view=sql-server-2017

https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/example-piecemeal-restore-of-only-some-filegroups-simple-recovery-model?view=sql-server-2017

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Editions

AVEVA Work Tasks provides you the following editions:

- Single Server
- Load Balanced Server

Single Server

A single server is used to host applications in a live environment. You can install single server in one of the following scenarios:

- Deployment of AVEVA Work Tasks Web Server, AVEVA Work Tasks Server, and AVEVA Work Tasks Database Server on a Single Machine
- AVEVA Work Tasks Web Server and AVEVA Work Tasks Server on Different Machines
- AVEVA Work Tasks Web Server, AVEVA Work Tasks Server, and AVEVA Work Tasks Database Server on Different Machines
- AVEVA Work Tasks Web Server Farm, Work Tasks Server, and AVEVA Work Tasks Database Server on Different Machines

Deployment of AVEVA Work Tasks Web Server, AVEVA Work Tasks Server, and AVEVA Work Tasks Database Server on a Single Machine

In this scenario, all AVEVA Work Tasks components are installed on the same machine. This scenario is preferred in a User Acceptance Test (UAT) environment. However, for UAT load testing and for better performance in a production environment, it is recommended to host the Databases on a different machine. In such scenarios, you must consider AVEVA Work Tasks Web Server and AVEVA Work Tasks Server on Different Machines.

For License related information, see AVEVA Enterprise Licensing.

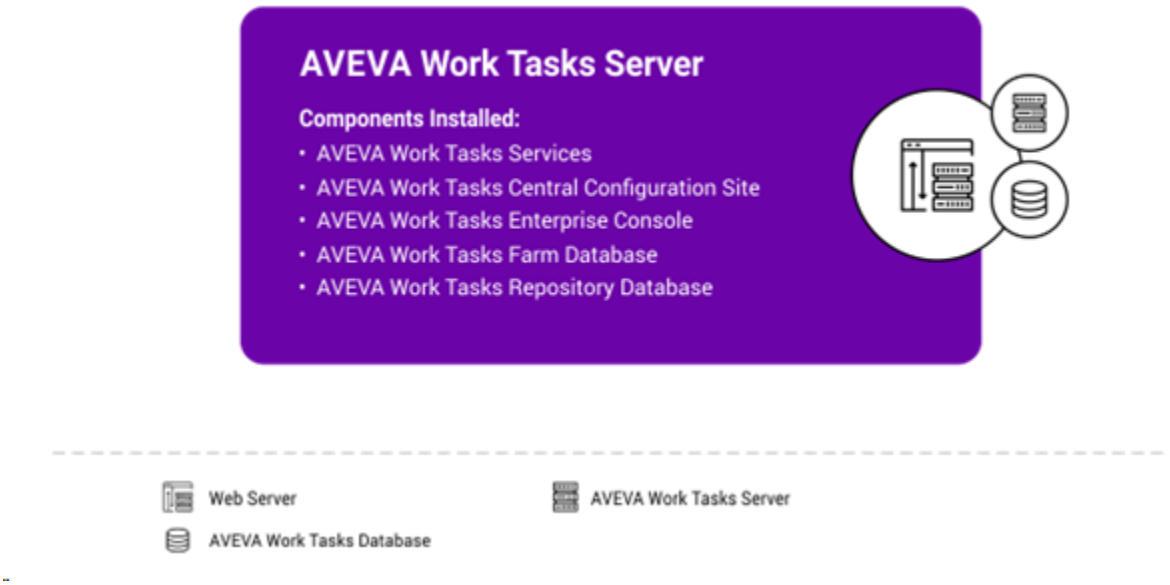
Role	**Components Installed**

AVEVA Work Tasks Server	• Services
	  • Client Service
	  • Advance Server Service (Disabled)
	  • Mobile Notification Service
	  • Communication Service
	  • Task Scheduler
	  • Workflow Engine
	• Databases
	  • Farm Database
	  • Repository Database

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Role	Components Installed
	• Central Configuration • Enterprise Console

System Landscape



Advantages

- Cost-effective deployment as machine requirement is minimal
- Easy to configure

Disadvantages

- May not scale well
- Failover and Clustering support not provided

System Requirements

The hardware and software requirements apply to both AVEVA Work Tasks Client and AVEVA Work Tasks Server systems. The system requirements are classified based on the requirements of the physical server.

Hardware Requirements

For information on AVEVA Work Tasks Hardware Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Software Requirements

For information on AVEVA Work Tasks Software Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

AVEVA Work Tasks Web Server and AVEVA Work Tasks Server on Different Machines

In this scenario, the Services and the Database Server will be installed on the same machine while the Central Configuration and the Enterprise Console will exist on a different machine. This deployment is recommended for a Production environment.

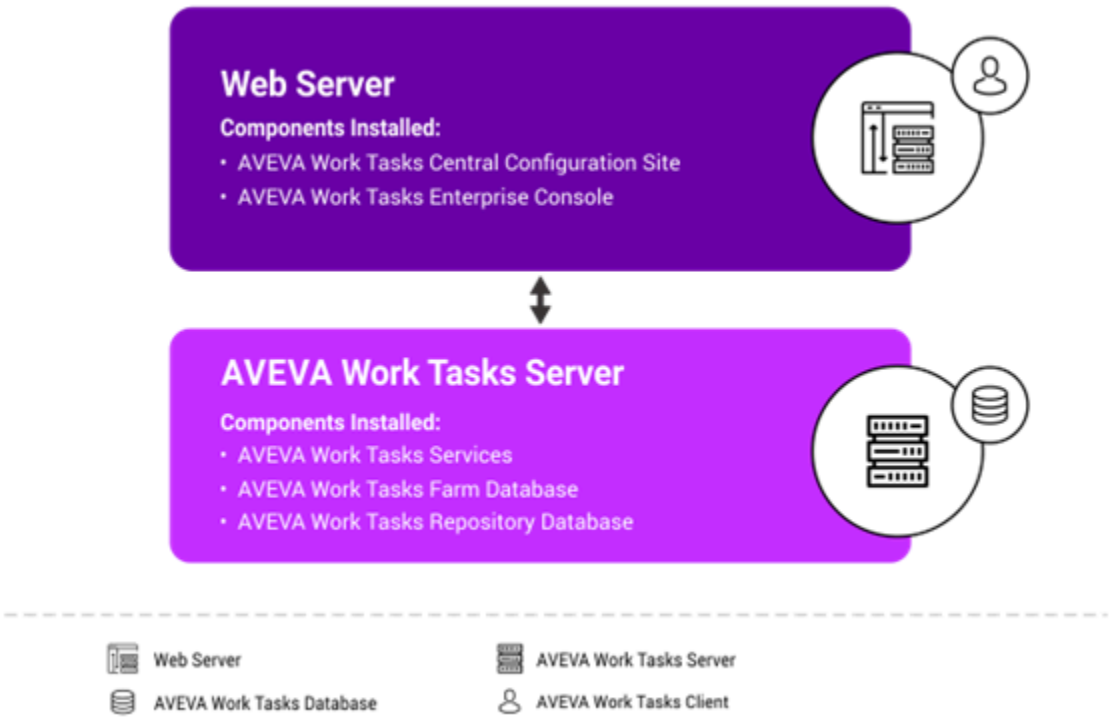
For License related information, see AVEVA Enterprise Licensing.

Role	**Components Installed**
AVEVA Work Tasks Web Server	• Central Configuration • Enterprise Console

AVEVA Work Tasks Server	• Services
	  • Client Service
	  • Advance Server Service (Disabled)
	  • Mobile Notification Service
	  • Communication Service
	  • Task Scheduler
	  • Workflow Engine
	• Databases
	  • Farm Database
	  • Repository Database

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

System Landscape



Advantages

- It is possible to secure AVEVA Work Tasks Databases behind a firewall
- Better performance is obtained from the database-intensive AVEVA Work Tasks Solution by hosting AVEVA Work Tasks Databases on a different machine

Disadvantages

- Load Balancing is not supported
- High CPU utilization due to load on Web applications affects the performance of AVEVA Work Tasks Workflow Engine
- Scale-Out is not possible

System Requirements

The hardware and software requirements apply to AVEVA Work Tasks Client and AVEVA Work Tasks Server systems. The system requirements are classified based on the requirements of the physical server.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Hardware Requirements

For information on AVEVA Work Tasks Hardware Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

Software Requirements

For information on AVEVA Work Tasks Software Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

AVEVA Work Tasks Web Server, AVEVA Work Tasks Server, and AVEVA Work Tasks Database Server on Different Machines

In this scenario, AVEVA Work Tasks Web Server, AVEVA Work Tasks Server, and the Database Server are all installed on different machines. This scenario is recommended for a Production environment.

For License related information, see AVEVA Enterprise Licensing.

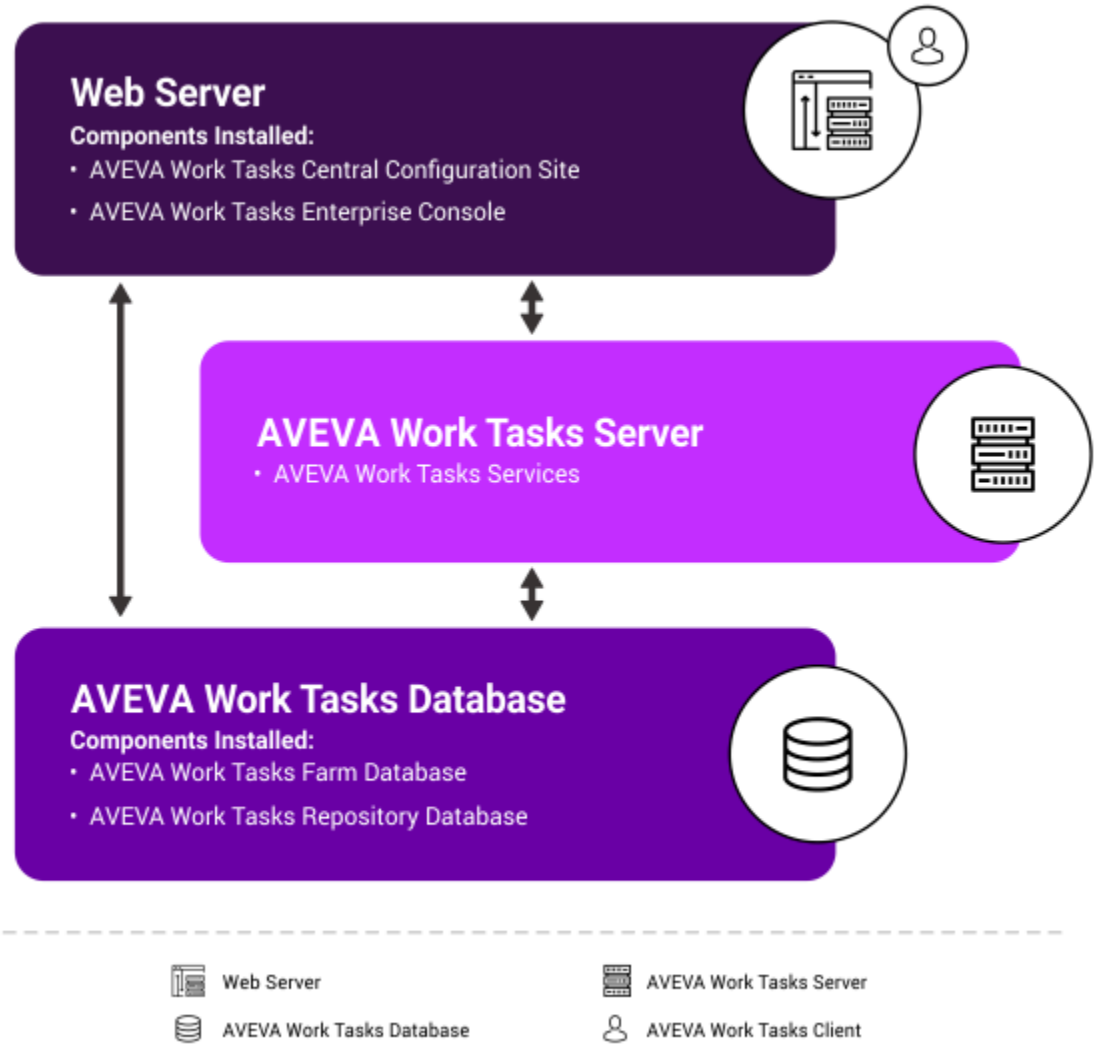
Role	Components Installed
AVEVA Work Tasks Web Server	• Central Configuration • Enterprise Console

AVEVA Work Tasks Server	• Services
	  • Client Service
	  • Advance Server Service (Disabled)
	  • Mobile Notification Service
	  • Communication Service
	  • Task Scheduler
	  • Workflow Engine

AVEVA Work Tasks Database Server	• Databases
	  • Farm Database
	  • Repository Database

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

System Landscape



Advantages

- It is possible to secure AVEVA Work Tasks Databases behind a firewall
- Better performance is obtained from the database-intensive AVEVA Work Tasks Solution by hosting the Databases on a different machine
- High CPU utilization due to load on Web applications does not affect the performance of AVEVA Work Tasks Workflow Engine

Disadvantages

- Load Balancing is not possible

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Scale out of AVEVA Work Tasks Workflow Engine service is not possible

System Requirements

The hardware and software requirements apply to AVEVA Work Tasks Client and AVEVA Work Tasks Server systems. The system requirements are classified based on the requirements of the physical server.

Hardware Requirements

For information on AVEVA Work Tasks Hardware Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

Software Requirements

For information on AVEVA Work Tasks Software Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

AVEVA Work Tasks Web Server Farm, Work Tasks Server, and AVEVA Work Tasks Database Server on Different Machines

In this scenario, the installation includes a AVEVA Work Tasks Web Server Farm besides AVEVA Work Tasks Server and Database. All these components are installed on different machines and the Web Farm on multiple Web servers.

A Web Server Farm is a group of two or more Web Servers. All the Web Servers in the Web Farm will have the Central Configuration and the Enterprise Console installed. AVEVA Work Tasks Services and the Database are installed on two different machines.

For License related information, see AVEVA Enterprise Licensing.

This scenario is recommended when you require a quick response from Web Servers under heavy load situations.

Role	**Components Installed**
Multiple AVEVA Work Tasks Web Server	• Central Configuration • Enterprise Console

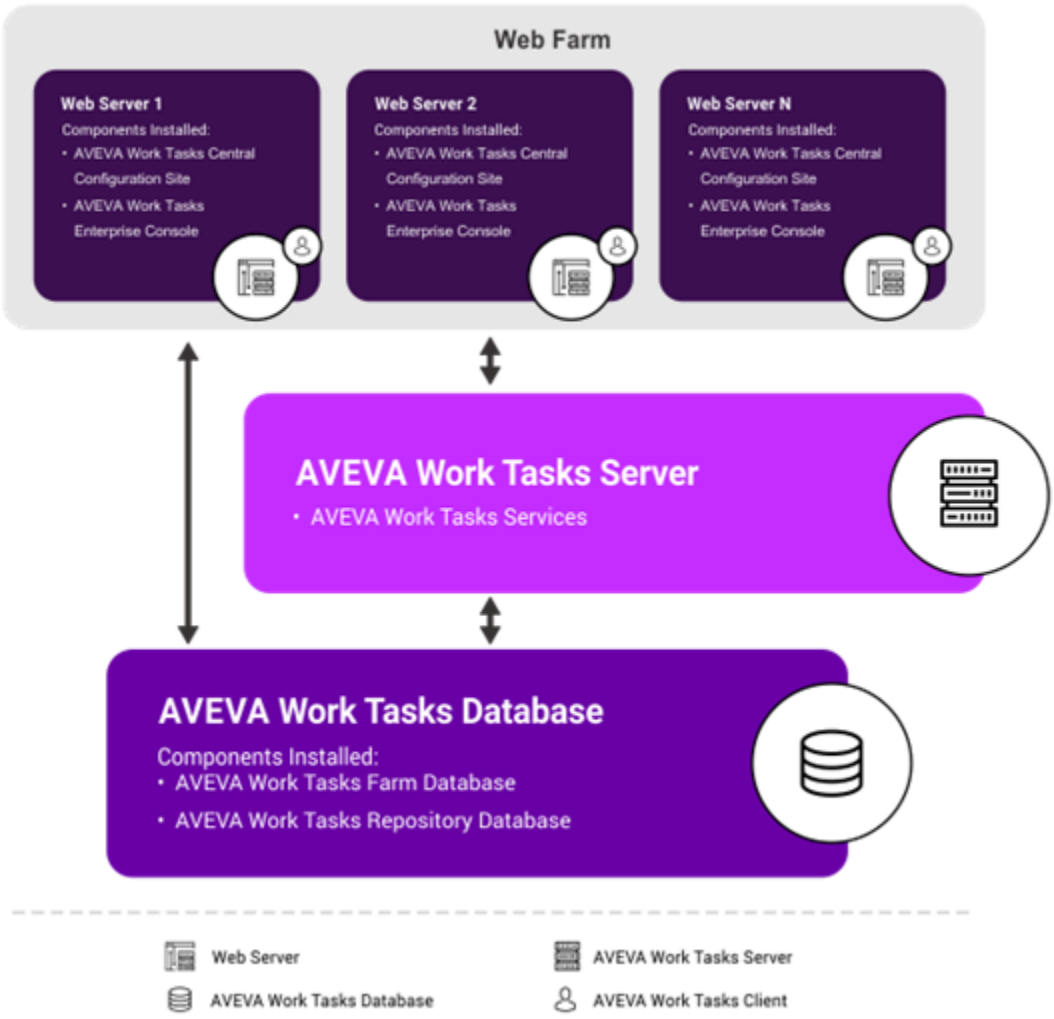
AVEVA Work Tasks Server	• Services
	  • Client Service
	  • Advance Server Service (Disabled)
	  • Mobile Notification Service
	  • Communication Service
	  • Task Scheduler
	  • Workflow Engine

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Role	Components Installed

Database Server	• Database
	  • Farm Database
	  • Repository Database

System Landscape



Advantages

- Response time is low and application throughput is high as processing requests received by a Web Farm are directed to different hosts in the Web Farm

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- It is possible to secure AVEVA Work Tasks Databases behind a firewall
- Better performance is obtained from the database-intensive AVEVA Work Tasks Solution by hosting the database on a different machine
- High CPU utilization due to load on Web applications does not affect the performance of the workflow engine

Disadvantages

- Load Balancing is not possible
- Scale-Out of AVEVA Work Tasks Engine service is not possible

System Requirements

The hardware and software requirements apply to AVEVA Work Tasks Client and AVEVA Work Tasks Server systems. The system requirements are classified based on the requirements of the physical server.

Hardware Requirements

For information on AVEVA Work Tasks Hardware Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

Software Requirements

For information on AVEVA Work Tasks Software Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

Single Server Licensing Information

There is no count limit for the Concurrent workflow execution for a Single Server license.

Load Balanced Server

The Load Balanced Server provides Failover and Clustering support. It provides the following features:

Active Load Balancing

This feature of the product causes the Workflow Engine and the Task Scheduler services on different servers to form a cluster and share the load. The workflow client always requests the least used server in the cluster for workflow execution. During the execution of a workflow, the Workflow Engine identifies a least used server and forwards the execution of that workflow to this server. However, this is true only for workflows executed in asynchronous mode. The load is shared uniformly by all the servers in the cluster leading to a true Active-Active load balancing environment.

Scalability

The number of servers running the Workflow Engine in a cluster can be increased or decreased depending on

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the number of workflow execution requests. After the configuration is done and the Workflow Engine is started, it automatically joins the cluster and starts sharing the load immediately.

Failover Capabilities

Any server running the Workflow Engine in the cluster can be safely shutdown or restarted without interrupting the execution of any workflows that may have been using that server. When a service is stopped or the server goes out of the network (following a shutdown or restart), all requests to that server are stopped. The other Workflow Engine services in the cluster will resume all pending tasks for the shutdown server.

Load Balanced Server Architecture

AVEVA Work Tasks uses its own clustering technology which allows real-time load balancing based on load on the server. It is managed either using the Multicast mode or the Database mode depending on the configuration done at the Farm level.

- Multicast mode: This uses multicasting UDP protocol to share the server status with other servers over a multicast IP and port at a regular interval.
- Database mode: This uses a database to store the server status of all the Enterprise Servers at a regular interval. All the servers keeps polling the data from the database to get the latest status of each server.

The Enterprise Server installation has a service called "Advance Server service" on each machine running the Load Balanced Server. This service acts as a Load Balancing coordinator between all the servers in the cluster, and continuously keeps communicating with the other Advance Server service running on other machines in the cluster. Services such as Workflow Engine, Task Scheduler, ArchestrA Mobile Notification Service, and Communication Service register themselves internally with this service to get health information about the cluster and provides their own health information to this service. This service performs the following operations:

- Monitoring CPU utilization of the server
- Monitoring health of each service such as Workflow Engine and Task scheduler in the given machine
- Broadcasts the health of each service to the entire cluster over a Multicast based UDP group using the Multicast mode or stores the information in the database using the Database mode. This broadcasting happens at a regular interval which is referred as the “Heart Beat”. So essentially the heart beat is important to determine how fast the cluster can respond to failures or respond to load pattern changes on the server (for example, high CPU utilization on the server).
- Whenever a new workflow execution is requested, one of the servers in the cluster decides which machine will execute it; based on the load on each of the servers. This server is called the “Authority server”. The Authority server role is not bound to any specific machine in the cluster, and any machine can become an Authority server in the cluster. This helps in scenarios where the Authority server itself goes down, since in such cases some other machine can play the role of an Authority server dynamically. The Authority server also makes some more decisions related to coordination of load balancing and failover in the entire cluster.
- If you select the Multicast option, it is recommended that a dedicated Network Interface Card (NIC) be present in the system as AVEVA Work Tasks uses Multicast-based UDP protocol for the cluster coordination. To have a dedicated Network Interface Card, UDP multicasting in the group and all the machines should be part of the same subnet. Network Switch used to connect all the servers for UDP multicasting must be configured to allow multicast packets.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Use of multicast or database to share the server status with other servers for such frequent messages greatly reduces load on the network and also allows the cluster to scale over multiple machines. The cluster can easily scale to even 4-5 machines and still allow true active-active load balancing.
- Each machine in the cluster communicates with the other machine in two modes. One mode of communication is the regular TCP/IP and the other mode is either by sending Multicast messages or polling data from the database at an appropriate point in time to exchange more specific data as and when required. In case a firewall exists between the servers it should allow broadcasting of both TCP/IP as well as multicast messages.

  Note: We do not recommend to have a firewall between machines in a cluster. The cluster can be setup behind a firewall.

- On AVEVA Work Tasks client machines (such as AVEVA Work Tasks Web Servers which are not having AVEVA Work Tasks Engine services installed, and are going to consume Workflow Engine services in a cluster), the Advance Server service plays the role of an assistant. The Advance Server service assists client APIs to determine which machine to connect in the cluster for the request. This allows fast identification of machine, rather than checking the health of cluster for each call, and also it has the most current state of the cluster. But this means in addition to machines which are participating in the cluster as Workflow Engine Role, even the client machine needs to have additional NIC and supporting network infrastructure to allow them to be part of the same Multicast group for consuming services from the cluster, when using the Multicast mode.

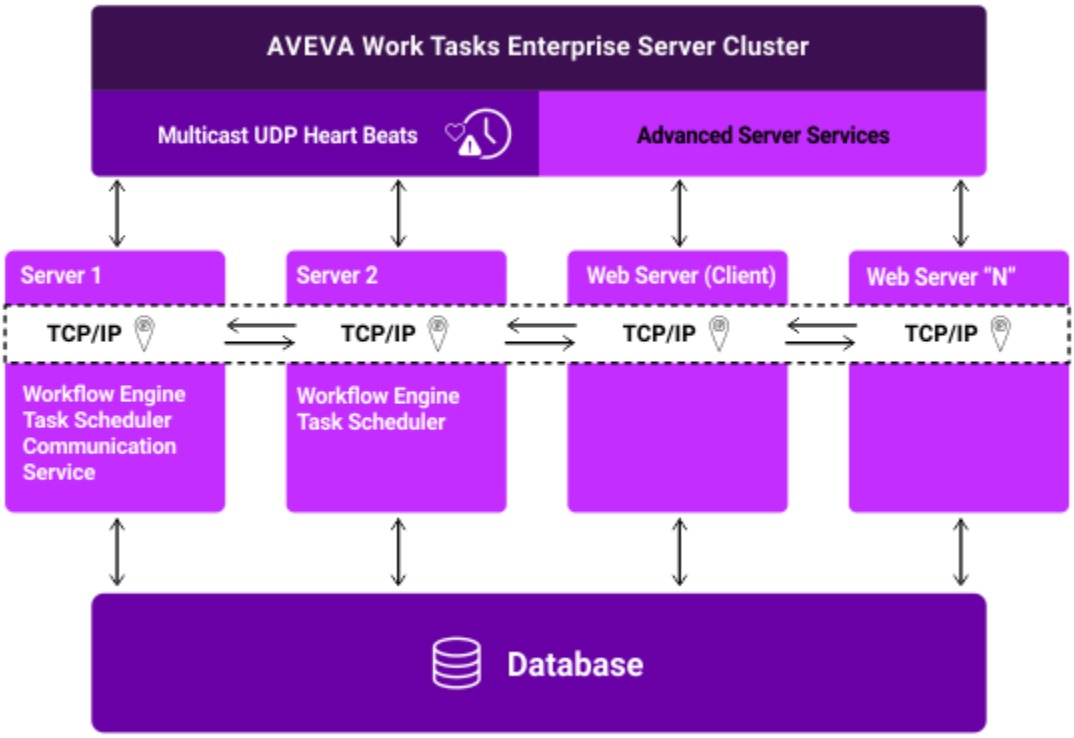
AVEVA Work Tasks Client Notification service which is responsible for monitoring incoming emails and SMS messages and File system events.

Database server can be a point of failure. It is recommended that the Database server is available with optimum performance.

Note: For Enterprise Server deployment, it is mandatory to have the date and time synchronization in all the Enterprise Servers running.

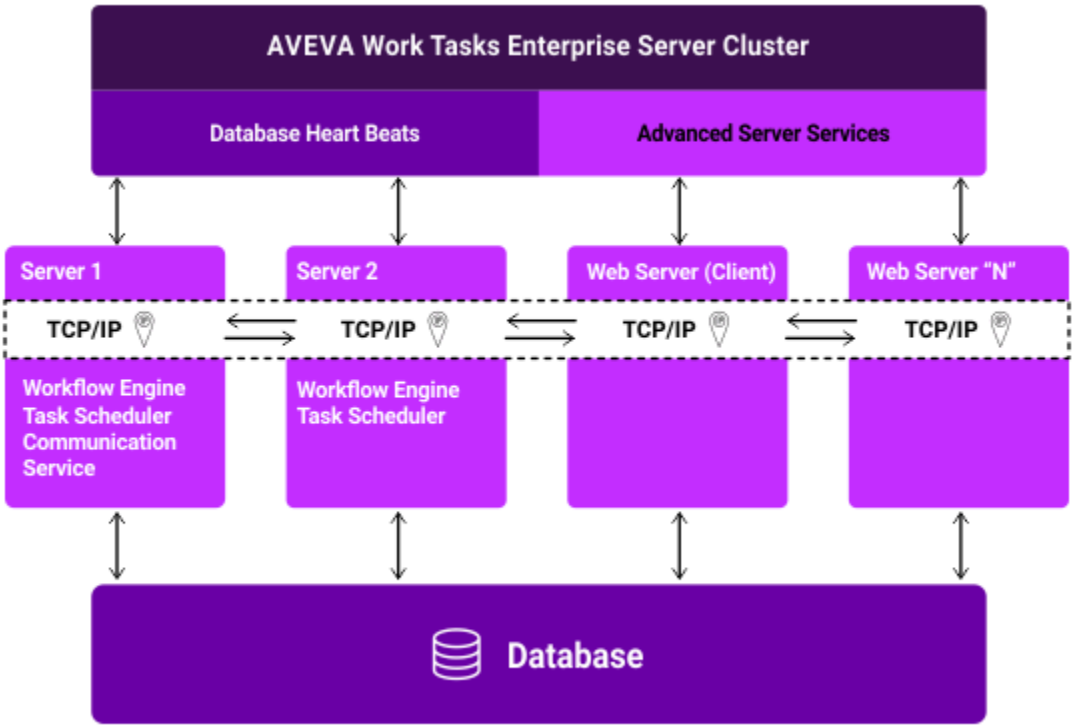
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Multicast Mode



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Database Mode



Note:

- In a virtualization environment, the following conditions hold good for the Load Balanced Server:

- While performing the disaster recovery scenarios, you must configure the recovery plan for the virtual machines in the following order, the first being with the highest priority:
a. Server1
b. Server2
c. Client

Load Balanced Server Installation Scenarios

This topic lists the scenarios in which the Load Balanced Server can be installed. See one of the following scenarios to understand its description, the system landscape, advantages and disadvantages, and system requirements.

The scenarios are listed below:

- Web Server Farm with Clustered Work Tasks Services and Databases on Different Machines
- AVEVA Work Tasks Web Server Farm Connecting to AVEVA Work Tasks Services with AVEVA Work Tasks Databases on a Different Machine
- Single Server Capable of Adding Machines at Any Time

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Single Server Capable of Adding Machines at Any Time

In this scenario all the AVEVA Work Tasks components will be installed on the same machine. This setup supports scale-out and you can add more machines depending on the number of workflows that need to be processed.

Use this scenario if you want to scale your deployment gradually as and when the load on the AVEVA Work Tasks Server increases. This deployment also allows you to provide Failover and Load Balancing support after additional servers are added in the cluster.

For License related information, see About AVEVA Enterprise Licensing.

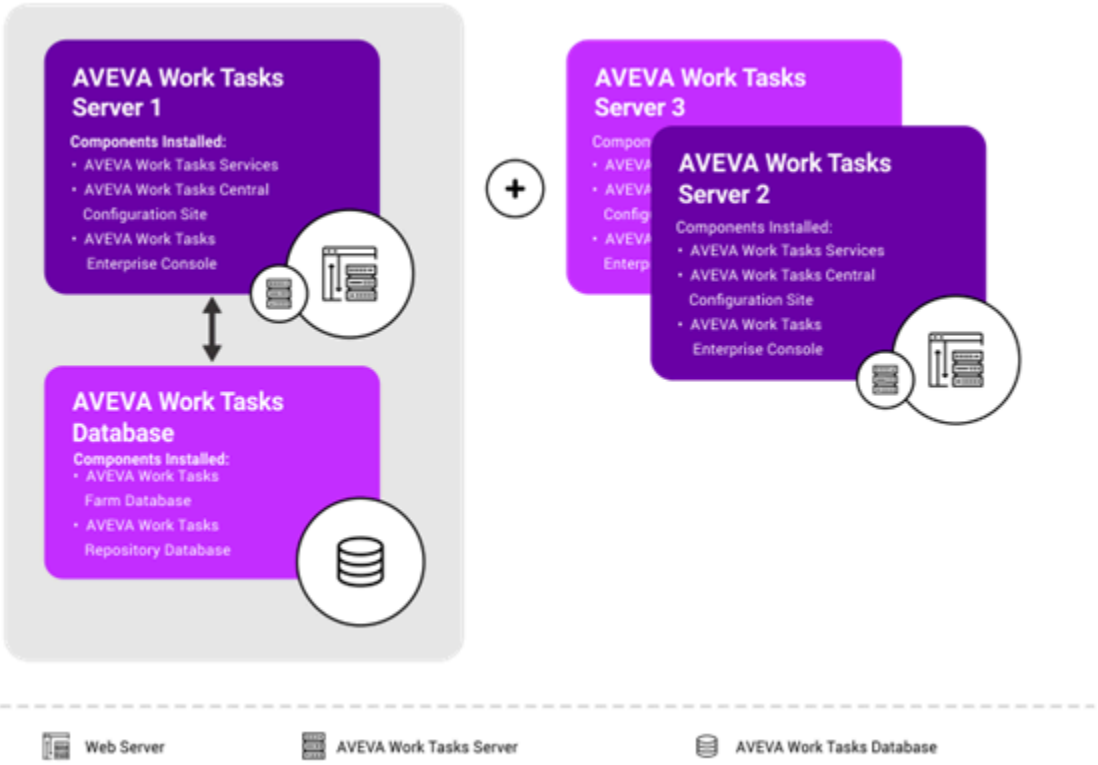
Role	Components Installed	Remarks

AVEVA Work Tasks Server	• Services
	  • Client Service
	  • Advance Server Service
	  • Mobile Notification Service
	  • Communication Service
	  • Task Scheduler
	  • Workflow Engine
	• Central Configuration
	• Enterprise Console
		Scalable setup

Database Server	• Databases
	  • Farm Database
	  • Repository Database
		-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

System Landscape



Advantages

- Cost-effective as machines with AVEVA Work Tasks Workflow Engine are added gradually depending on the load in a Production environment
- Supports Load Balancing if more than one machine is added

System Requirements

The hardware and software requirements apply to AVEVA Work Tasks Client and AVEVA Work Tasks Server systems. The system requirements are classified based on the requirements of the physical server.

Hardware Requirements

For information on AVEVA Work Tasks hardware requirement, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

Software Requirements

For information on AVEVA Work Tasks software requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Web Server Farm with Clustered Work Tasks Services and Databases on Different Machines

In this scenario, you have a AVEVA Work Tasks Web Server and AVEVA Work Tasks Services installed on the same machine. You can host this set up on multiple computers to form a AVEVA Work Tasks Web Server Farm with clustered AVEVA Work Tasks Services. The Databases are installed on a machine by itself. While scaling out, ensure that the machine has both the AVEVA Work Tasks Web Server and the AVEVA Work Tasks Services installed.

A AVEVA Work Tasks Cluster is a group of servers running AVEVA Work Tasks Services as a single logical entity. In a AVEVA Work Tasks Cluster setup only a single Communication service must be running.

For License related information, see About AVEVA Enterprise Licensing.

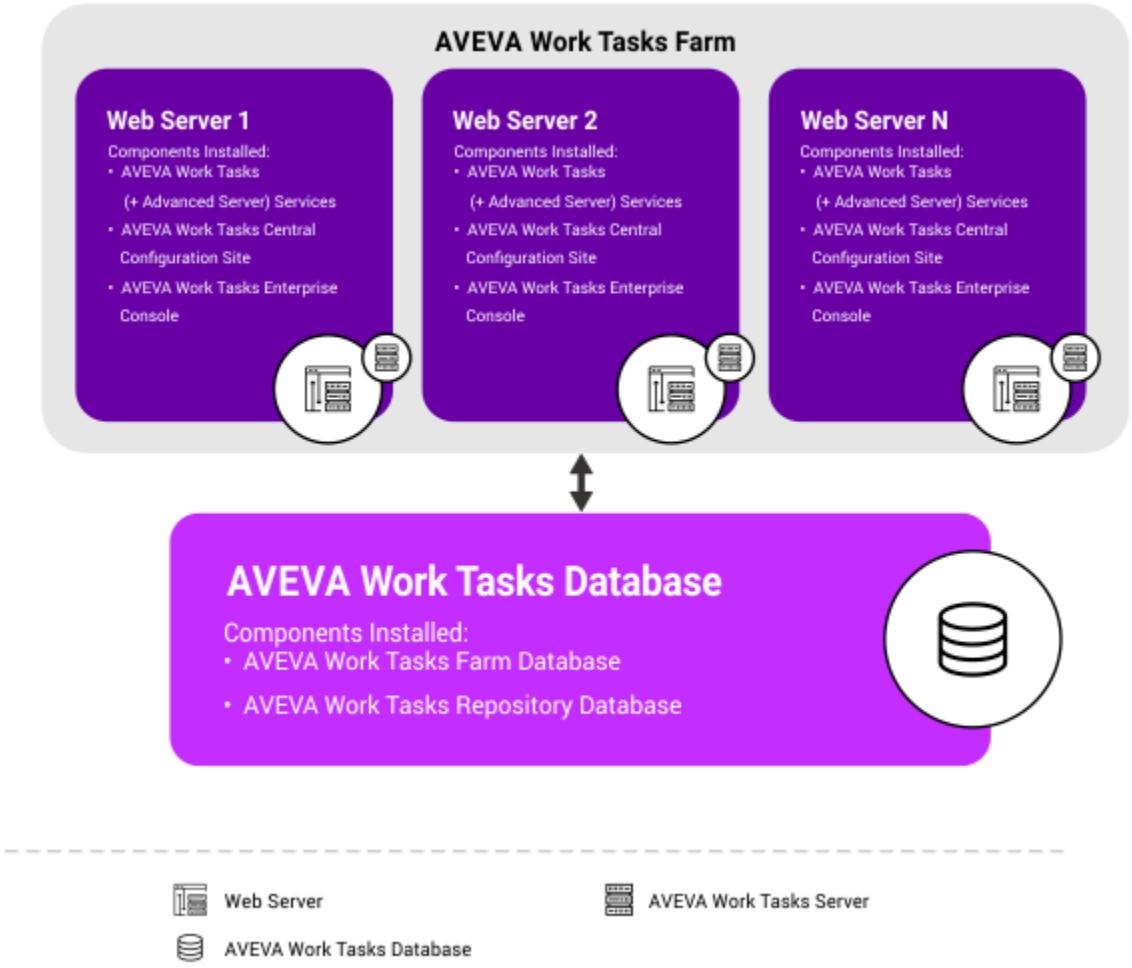
Role	Components Installed

AVEVA Work Tasks Web Servers - Web Server Farm with AVEVA Work Tasks Services	• Services
	  • Client Service
	  • Advance Server Service
	  • Mobile Notification Service
	  • Communication Service
	  • Task Scheduler
	  • Workflow Engine
	• Central Configuration
	• Enterprise Console

AVEVA Work Tasks Database Server	• Databases
	  • Farm Database
	  • Repository Database

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

System Landscape



System Requirements

The hardware and software requirements apply to AVEVA Work Tasks Client and AVEVA Work Tasks Server systems. The system requirements are classified based on the requirements of the physical server.

Hardware Requirements

For information on AVEVA Work Tasks Hardware Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

Software Requirements

For information on AVEVA Work Tasks Software Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

AVEVA Work Tasks Web Server Farm Connecting to AVEVA Work Tasks Services with AVEVA Work Tasks Databases on a Different Machine

In this scenario, you can maintain a AVEVA Work Tasks Web Server Farm with AVEVA Work Tasks Web Servers and a AVEVA Work Tasks Services cluster on separate machines while the the Databases are installed on a different machine. This deployment provides scalability and high availability of AVEVA Work Tasks Server and AVEVA Work Tasks Web Servers. This is achieved by distributing the load equally among all the machines in the cluster and the AVEVA Work Tasks Web Server Farm.

Multiple servers in the AVEVA Work Tasks Web Server Farm or the cluster run the same AVEVA Work Tasks services. Failure of a server does not impact other servers. You can take the affected server offline while the other servers maintain the service. This type of installation provides Failover. Thus allowing you to easily add or remove machines from the AVEVA Work Tasks Web Server Farm or from the cluster without affecting the execution of workflows.

This is the optimal scenario for heavy load conditions and can be considered if you anticipate heavy load on the Workflow Engine or the AVEVA Work Tasks Web Server.

For License related information, see About AVEVA Enterprise Licensing.

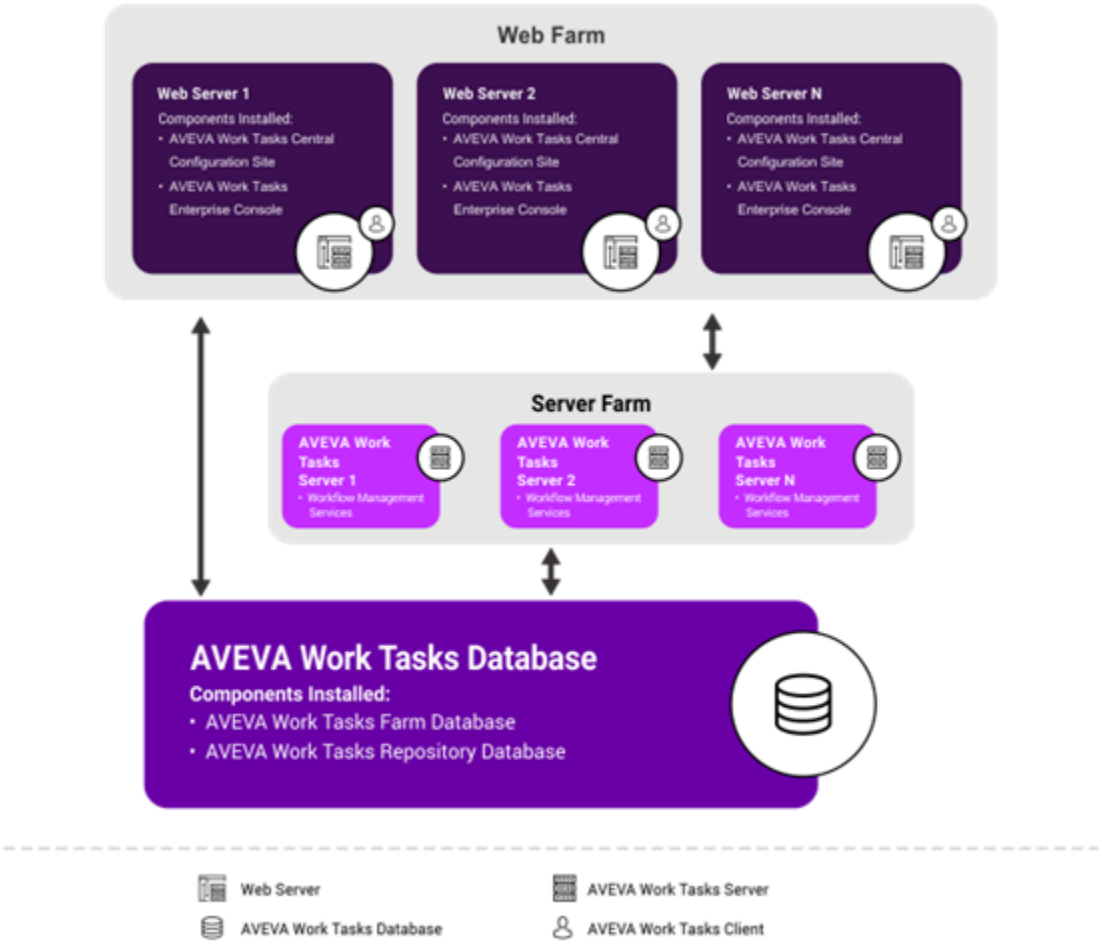
Role	Components Installed	Remarks
Front-End Servers	• Central Configuration • Enterprise Console • Advance Server Service • Client Service	Multiple AVEVA Work Tasks Web Servers

AVEVA Work Tasks Server	• Services
	  • Client Service
	  • Advance Server Service
	  • Mobile Notification Service
	  • Communication Service
	  • Task Scheduler
	  • Workflow Engine
		Multiple AVEVA Work Tasks Servers

Database Server	• Databases
	  • Farm Database
	  • Repository Database	

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

System Landscape



Advantages

- Provides Load Balancing support
- Dedicated machines are available for AVEVA Work Tasks Web Servers and AVEVA Work Tasks Workflow Engine
- Supports very heavy load
- Failover support is provided
- Scale out is possible

System Requirements

The hardware and software requirements apply to AVEVA Work Tasks Client and AVEVA Work Tasks Server systems. The system requirements are classified based on the requirements of the physical server.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Hardware Requirements

For information on AVEVA Work Tasks Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

Software Requirements

For information on AVEVA Work Tasks Software Requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

Behavior of AVEVA Work Tasks Services on Enterprise Server

This section provides information on the activity of AVEVA Work Tasks Services when installed on the Load Balanced Server.

Workflow Engine and client

Whenever a workflow execution is requested by a client, the Advance Server service running on the client machine determines the least used AVEVA Work Tasks Engine machine and routes the request to that machine.

For workflow dehydrating from a sleep state, the execution request is actually sent to the machine where the last execution for that workflow had happened, if that machine is not active in the cluster at that point the event is routed to the least used machine using the Advance server service.

If a machine running a Workflow engine goes down in the cluster, the machine acting as an Authority Server takes over the pending workflows of that machine. Thus the Authority server provides high availability and maintains uninterrupted workflow execution.

Task Scheduler

Task scheduler engine has two main components, one which is responsible for monitoring the task for next possible execution and other being actual execution of these tasks.

Task scheduler service running on acting Authority server is the only instance of task scheduler service which is responsible for fetching tasks. Fetching tasks does not require any load balancing. However, failover is provided whenever the active Authority server goes down in the cluster. Another machine in the cluster takes the role of the Authority server and starts monitoring the task again.

For execution of tasks real load balancing is used, and execution is scheduled on the least used machine. This machine provides active-active load balancing with failover support.

Communication Service

AVEVA Work Tasks Service makes use of communication service settings for the following activities:

Wait for E-mail: The Workflow processes can be designed to be initiated or processed on receipt of email matching specified parameters. The Communication service monitors the specified port and IP for incoming emails that could be used in the Workflows. If default option is selected, email should be forwarded to all the Enterprise servers. Authority server will process the message and other servers will ignore it.

Note: In the Enterprise Server environment, Wait for Email activity can read file from the network shared path or from the Map network drive. To read the files, you must provide the relevant access rights for the Communication Service on network shared path or Map Network drive.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Wait for SMS: The communication service can be setup to monitor incoming SMS.

Note: Device must be connected to the authority server.
Sending SMS: The Workflows can send SMS through a Global System for Mobile communication (GSM) device. The Communication Service uses settings and configuration specified in this screen for sending SMS.

Communication service running as an acting Authority server is the only instance of communication service which is responsible for processing the email and SMS notifications. Failover is provided whenever the active Authority server goes down in the cluster. Another machine in the cluster takes the role of the Authority server and starts processing the notifications.

If you want to run the Communication Service under the Enterprise Server mode and the services are installed on different paths in the servers, you must do the following modifications at the database level:

1. Query the Communication Service Configuration property from the database:

```
SELECT Properties FROM dbo.SKServiceMapping with (nolock)
WHERE Name = 'SMTPNotificationEngine2' AND Id = 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX'
```

Update the properties of the XML returned as shown in the following query, removing the directory path.

Actual XML:

```
<properties><property key="listnerip"><![CDATA[ServerName]]></property><property
key="listnerport"><![CDATA[8770]]></property><property
key="taskhandlerpath"><![CDATA[C:\Program Files\AVEVA\Work Tasks\Bin\
Workflow.NET.EmailReceiver.NET2.dll]]></property><property
key="taskhandlerclass"><![CDATA[Workflow.NET.EmailReceiver.CDecisionHandler]]></property
><property key="mailserver"><![CDATA[Default]]></property><property
key="blockedmailaddresses"><![CDATA[]]></property><property
key="smtpserverip"><![CDATA[ServerName]]></property><property
key="smtpserverport"><![CDATA[25]]></property><property
key="pophost"><![CDATA[ServerName]]></property><property
key="popmailport"><![CDATA[110]]></property><property
key="popusers"><![CDATA[user1@domain.com|pwd]]></property><property
key="provideencryptedpwd"><![CDATA[false]]></property><property
key="poptimeinterval"><![CDATA[0.00:00:15]]></property><property
key="smsinterval"><![CDATA[0.00:01:00]]></property><property
key="smsreciever"><![CDATA[false]]></property><property
key="smstaskpath"><![CDATA[C:\Program Files\AVEVA\Work Tasks\Bin\
Workflow.NET.SMSReceiver.NET2.dll]]></property><property
key="smstaskclass"><![CDATA[Workflow.NET.SMSReceiver.SMSTimeoutTaskHandler]]></property>
<property key="smsprovider"><![CDATA[GSM|C:\Program Files\AVEVA\Work Tasks\Bin\
Workflow.NET.SMS.Messenger.NET2.dll|Workflow.NET.SMS.Messenger.SMSReceiver|]]></property
><property key="isPOP3passwordencrypted"><![CDATA[false]]></property></properties>
```

Modified XML:

```
<properties><property key="listnerip"><![CDATA[ServerName]]></property><property
key="listnerport"><![CDATA[8770]]></property><property
key="taskhandlerpath"><![CDATA[Workflow.NET.EmailReceiver.NET2.dll]]></property><propert
y
key="taskhandlerclass"><![CDATA[Workflow.NET.EmailReceiver.CDecisionHandler]]></property
><property key="mailserver"><![CDATA[Default]]></property><property
key="blockedmailaddresses"><![CDATA[]]></property><property
key="smtpserverip"><![CDATA[ServerName]]></property><property
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
key="smtpserverport"><![CDATA[25]]></property><property
key="pophost"><![CDATA[ServerName]]></property><property
key="popmailport"><![CDATA[110]]></property><property
key="popusers"><![CDATA[user1@domain.com|pwd]]></property><property
key="provideencryptedpwd"><![CDATA[false]]></property><property
key="poptimeinterval"><![CDATA[0.00:00:15]]></property><property
key="smsinterval"><![CDATA[0.00:01:00]]></property><property
key="smsreciever"><![CDATA[false]]></property><property
key="smstaskpath"><![CDATA[Workflow.NET.SMSReceiver.NET2.dll]]></property><property
key="smstaskclass"><![CDATA[Workflow.NET.SMSReceiver.SMSTimeoutTaskHandler]]></property>
<property key="smsprovider"><![CDATA[GSM|
Workflow.NET.SMS.Messenger.NET2.dll|Workflow.NET.SMS.Messenger.SMSReceiver|]]></property
><property key="isPOP3passwordencrypted"><![CDATA[false]]></property></properties>
```

2. Update the database entry with the modified XML.

```
Update dbo.SKServiceMapping
SET Properties = '<properties><property
key="listnerip"><![CDATA[ServerName]]></property><property
key="listnerport"><![CDATA[8770]]></property><property
key="taskhandlerpath"><![CDATA[Workflow.NET.EmailReceiver.NET2.dll]]></property><property
key="taskhandlerclass"><![CDATA[Workflow.NET.EmailReceiver.CDecisionHandler]]></property><prope
key="mailserver"><![CDATA[Default]]></property><property
key="blockedmailaddresses"><![CDATA[]]></property><property
key="smtpserverip"><![CDATA[ServerName]]></property><property
key="smtpserverport"><![CDATA[25]]></property><property
key="pophost"><![CDATA[ServerName]]></property><property
key="popmailport"><![CDATA[110]]></property><property
key="popusers"><![CDATA[user1@domain.com|pwd]]></property><property
key="provideencryptedpwd"><![CDATA[false]]></property><property
key="poptimeinterval"><![CDATA[0.00:00:15]]></property><property
key="smsinterval"><![CDATA[0.00:01:00]]></property><property
key="smsreciever"><![CDATA[false]]></property><property
key="smstaskpath"><![CDATA[Workflow.NET.SMSReceiver.NET2.dll]]></property><property
key="smstaskclass"><![CDATA[Workflow.NET.SMSReceiver.SMSTimeoutTaskHandler]]></property><proper
key="smsprovider"><![CDATA[GSM|
Workflow.NET.SMS.Messenger.NET2.dll|Workflow.NET.SMS.Messenger.SMSReceiver|]]></property><prope
key="isPOP3passwordencrypted"><![CDATA[false]]></property></properties>'
WHERE Name = 'SMTPNotificationEngine2' AND Id = 'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX'
```

3. Restart the Communication service on all the servers.

Note: It is recommended that the Communication services for all the Enterprise Servers should have similar configuration.

Tools Used in Enterprise Server

In the Load Balanced Server, it is mandatory to check if all the machines (AVEVA Work Tasks Server and the Client) are able to communicate with each other before installation using **Multicast** protocol.

AVEVA Work Tasks provides a tool called **Multicast tool** which you can use to verify the multicast communication between machines on the same subnet. The Multicast tool ensures that all machines used for AVEVA Work Tasks Server installation are able to communicate with each other. This test ensures that all the Server machines involved can actively support Load Balancing and Failover support feature of the Enterprise Server.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Multicast tool can be copied on your local hard disk. To check communication between machines, the Multicast tool must be run simultaneously on all the machines.

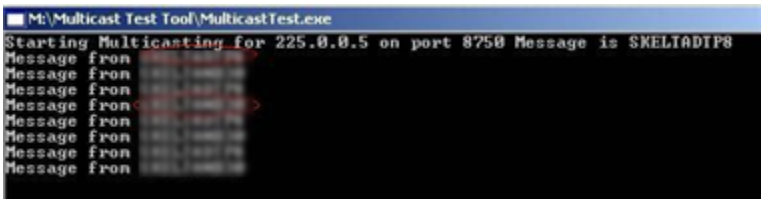
Messages received from all the machines are displayed by the Multicast tool indicating that the machines are able to communicate with each other. A machine not able to receive or display messages from other machines indicates that the machine is not able to communicate with other machines.

More details are provided in specific installation scenarios on how to use this tool.

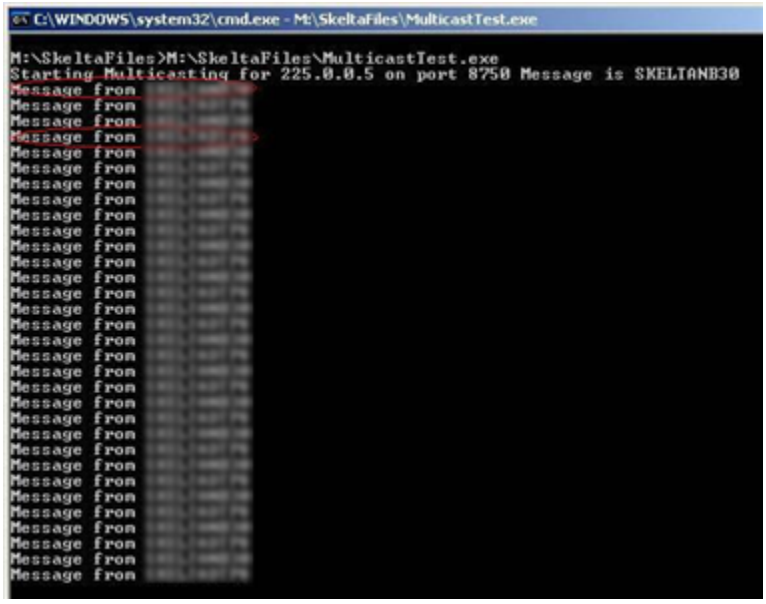
Example

The following example illustrates after running the Multicast tool from machines <ServerName1> and <ServerName2>.

The following image shows message from <ServerName1>:



The following image shows message from <ServerName2>:



The above figures indicate that there is a successful communication between <ServerName1> and <ServerName2> machines.

Load Balanced Server Limitations

The Load Balanced Server has the following limitations:

- File transfer action should not be used with the local path. Universal Naming Convention (UNC) path should be used in the load balanced environment.
- For Create Office - Document action, destination path should be UNC, FTP, or http. These destination paths ensure that all the Workflow Engines can perform file transfers without any issues.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Load Balanced Server Licensing Information

- All machines running the AVEVA Work Tasks Engine Service in a cluster will require a valid Load Balanced Server license. In a cluster environment, the AVEVA Work Tasks Client installation does not require a license.
- There is no count limit for the Concurrent workflow execution for a Load Balanced Server license.

Installing Different Editions

The following table lists the types of installation scenarios that you can use to identify your installation requirements and map them to the best-suited deployment.

Requirement	**Recommended Deployment**
Need to implement AVEVA Work Tasks solution in a production environment and at the same time secure the database.	For this requirement, you can consider any of the following scenarios: • Web server and AVEVA Work Tasks Server on computer1 and database on computer2. • Web server, AVEVA Work Tasks Server, and database on different machines For more information on these scenarios, refer to the scenarios section of Single Server.
Need fast response from Web servers under heavy load situations.	Web Farm (Multiple Web servers), AVEVA Work Tasks Server, and database on different machines is the scenario used for heavy load situations. For more information on this scenario, refer to the scenarios section of Single Server.
Heavy load anticipated on the AVEVA Work Tasks Server or on Web server - there is also a need to provide Load balancing and fail-over support.	Web Farm connecting to separate Services cluster with database on a different machine is the scenario best-suited for heavy load with load balancing and fail-over support. For more information on this scenario, refer to the scenarios section of Load Balanced Server Installation Scenarios.

Layout Editing Modes

Before you install AVEVA Work Tasks, ensure the following conditions are met

- Identify the roles to be performed by each server in your network. For more information, see Identifying Server Roles.
- Identify the authentication requirements for each server.
- List servers that are behind a firewall. Access to such servers would require the relevant ports to be opened.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

See Prerequisites before continuing with the installation.

Identifying Server Roles

A Server Farm comprises physical servers that run logical services. Physical servers are tangible machines. Logical services are installed on physical servers. Some logical services run on Web servers and others run on AVEVA Work Tasks Servers. Logical services are installed on one or more physical servers. The logical services running on a physical server determines the role of the server in the server farm.

Example:

The server running a database is called the database server.

The following table lists the server roles based on the application/services running on a server:

Applications/Services running on a Server	**Server Role**
Database applications	Farm Database Server and Repository Database Server
Workflow Engine, Task Scheduler, and Communication services	AVEVA Work Tasks Server
Advance Server service,Workflow Engine, Task Scheduler, and Communication services	Enterprise Server or Clustering Server
Central Configuration and Enterprise Console	Web Server

The number and kind of servers required to perform a particular role depend on many factors, such as the number of users, volume of data, and performance of the solution required.

Identifying User Source

A User Source is an external user database used to manage a user's identity. AVEVA Work Tasks supports the following User Source, which is an out of the box solution for user authentication:

- **Windows Active Directory Server**: This user source allows you to retrieve user related information from the active directory.
- **ASP.NET Membership**: This user source allows you to retrieve user related information from an ASP.NET membership provider. The ASP.NET membership provides a built-in option to validate and store user accounts. This user source thus helps you manage user authentication.
- **Forms**: Forms make it possible to retrieve user related information from another application database.
- **List**: These are out-of-the-box user sources provided by AVEVA Work Tasks. The Lists use the database for storing user information.

Apart from the above listed user providers, you can use a custom User Source too. User authentication enables users to participate in human activities that form a part of the AVEVA Work Tasks Workflow.

See Also

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Architecture

Making Database Selection

For information on Database requirements, see *AVEVA Work Tasks 2023 SP1 Release Notes*.

Setting up Communication

AVEVA Work Tasks uses a variety of communication protocols such as e-mails to communicate with actors. An actor is a user who is responsible for a task or an activity.

AVEVA Work Tasks workflows can be initiated on receipt of e-mails. The workflows can also trigger e-mails, thus allowing actors to operate on processes from an email interface.

Integrating Email

The mail delivery channel in AVEVA Work Tasks allows the action items of human activities like Approval and Checklist to be sent by email to the intended recipient(s). The recipient of these emails can perform the required action from the Inbox of email clients like Microsoft Outlook.

See Also

Microsoft Outlook Integration

Setting up Ports

When AVEVA Work Tasks is configured with default settings, it installs Enterprise Console and Central Configuration on a default port. You can configure the solution to use other port numbers through the custom settings option using the the Farm Configuration Wizard. Ports configured for AVEVA Work Tasks must be opened for access through firewalls.

For more information, refer to Enable and Verify MSDTC.

You must configure the ports manually for Central Configuration and Enterprise Console to work in Mozilla Firefox. To configure the ports, add the line prefix ("`network.security.ports.banned.override`", "`port1,port2`") to "`C:\Program Files\Mozilla Firefox\defaults\pref\firefox.js`", where `port1` and `port2` represent the ports you need to browse.

The following table lists the default ports used by various services while running AVEVA Work Tasks:

Services	Port Type	Default Port number	Description
Advance Server Services with Multicast IP address 225.0.0.10	multicast port	8850	This port is used to check the connectivity between all the machines while running the Load Balanced Server.
	tcp port	8853	Communication between the Workflow Engine and the Advance Server

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Services	Port Type	Default Port number	Description
			service is established through this port. This port is used to get the status of the least used server, and also analyze server performance.
Workflow Engine	load balance server port	8853	This is the port on which the Advance Server service listens to the registration from other services.
	load balance client port	8859	This is the port on which the Advance Server client services listen to the events.
	port	8855	This is the port on which the Workflow Engine listens to client requests.
SMTP services	smtp port	25	This is the default port for sending e-mails.
Communication services	listener port	8770	-
	pop-mail port	110	This is the default port for receiving emails.
Task Scheduler services	load balance server port	8853	This is the default port for the Load Balanced Server.
	load balance client port	8860	This is the default port for the Load Balanced Client.
	port	8856	-
Service Monitor EXE	port	8866	This is the default port used by the AVEVA Work Tasks Service Monitor EXE.
License Server Port	port	55555	This is the default port used by the License Server. Ensure that you enter a valid port number.
Client service	listener port	8863	Default listener port for

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Services	Port Type	Default Port number	Description
			the Client service. This can be changed in **CentralConfig.xml** file located at the root of the installation folder.

The following table lists the ports used by AVEVA Work Tasks services, when the AVEVA Work Tasks Pro application is used outside Corporate Network.

Services	Port Type	Default Port number	Description
Workflow Engine	SB Over TCP	9350, 9351, 9352, 9353	This is the port on which the Workflow Engine communicates with Service Bus for the AVEVA Work Tasks Pro application.
Communication services	SB Over TCP	9350, 9351, 9352, 9353	This is the port on which the Workflow Engine communicates with Service Bus for the AVEVA Work Tasks Pro application.
Push Notification service	http port	80	This is the port on which AVEVA Work Tasks communicates with the Push Notification servers.
	https port	443	

Cookies Created by Web Application

The cookies created by Aveva Work Tasks Web application are listed in the table:

Name	Description
_instanceName	The cookie having the provider name information with encrypted value to which the user is logged in to the web application
_prov	The cookie having the provider internal name information with encrypted value to which the user is logged in to the web application
_provtext	The cookie having the provider display name information with encrypted value to which the user is logged in to the web application

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name	Description
_repo	The cookie having the repository or application name with encrypted value to which the user is logged in to the web application
_userinfo	The cookie having user information with encrypted value
skdatecorrectionfactor	The cookie created on the client-side to list the grid data based on the client's offset time
skpointerucontextinfo	The cookie created on server side with a dummy GUID. Used by the client-side for handling the closing of windows when a user logs-out of the application. This cookie is a pointer to the dynamic cookie skucontextinfo2token (in the current context), which can be referred to check whether the user is logged out of the application.
skucontextinfo2	The dynamic cookie created on server side for user context information with encrypted value
skucontextinfo2token	The dynamic cookie created on server side with a dummy GUID. Used by the client-side for handling the closing of windows when a user logs out of the application

Notes:

- All the cookies are HTTP Only, except for skdatecorrectionfactor, skpointerucontextinfo, and skucontextinfo2 (cookie name might vary as it is dynamic), as they are needed on the client-side and do not have any sensitive information.

- All cookies are secured if the web application is in a secure state.

Prerequisites

Before installing AVEVA Work Tasks, the installer installs the following system-specific prerequisites:

- Microsoft .NET Framework 4.8
- Enables IIS Feature
- Microsoft .Net 6.0.7 - Windows Server Hosting
- VS C++ Redistributable2022
- Operations Control Logger
- Platform Common Services 7.1.0
- Common Licensing API Framework

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- AVEVA Enterprise Licensing 3.7.002

AVEVA Work Tasks utilizes a server-based licensing subsystem that provides centralized license management. Licenses must be activated before use.

- License Manager and License Server should be installed separately from the **License Manager** and **License Server** folders respectively. These folders are available in the **AVEVA Enterprise Licensing** folder in the ISO file.
- The Configuration utility, which will be prompted after the 'AVEVA Enterprise Licensing API Framework' is installed, can be used for connecting to the AVEVA License Server. For more information, see AVEVA Enterprise License documentation.

Note:
- Copy the Installation EXE and ISSetupPrerequisites folder to the machine where you want to install AVEVA Work Tasks 2023 SP1 and then start the installation.

- If any of the above mentioned system-specific prerequisites are already installed, the installer skips installing that particular prerequisite component and proceeds with other prerequisites.

- If you install AVEVA Work Tasks on a system where prerequisite Microsoft .NET Framework 4.8 is not already installed, then the system will restart automatically after the installation of this prerequisite. To avoid system restart, install each of these prerequisites separately using the respective installers from the **ISSetupPrerequisites** folder.

- AVEVA Work Tasks needs .NET Core 6.0.7 or a newer version to be installed on the machine. Although the Work Tasks installer tries to install it as a prerequisite, if the .NET Core installation encounters any issues, the installation process will continue. However, it will be incumbent upon the customer to manually install .NET Core in such instances.

- If you are planning to install any application on the same machine that installs earlier version of the above prerequisites, then ensure to install that application before installing AVEVA Work Tasks.

- While installing AVEVA Work Tasks, if you opt to restart the machine after **Operations Control Logger**, **Platform Common Service**, or **Common Licensing API Framework** prerequisite installation,
-- Installation does not resume after a restart. Run the installer to continue the installation.
-- Common Licensing API Framework prerequisite installation might get skipped if you restart the machine after Platform Common Service prerequisite installation.
It is recommended not to restart the machine while installing Operations Control Logger, Platform Common Service, or Common Licensing API Framework prerequisites.

- PCS 4.7 is dependent on the following components in case PCS is getting installed. Regardless of which method you choose for integration, you need to ensure that the following components are installed before attempting to install the PCS Framework.
-- .NET Framework 4.8 Full or later
-- Visual C++ 2015 - 2022 Redistributable (x86 and x64). Use Universal C Runtime. (https://docs.microsoft.com/en-US/cpp/windows/latest-supported-vc-redist?view=msvc-170)
-- windowsdesktop-runtime-6.0.x-win-x64.exe (https://dotnet.microsoft.com/en-us/download/dotnet/6.0)
-- aspnetcore-runtime-6.0.x-win-x64.exe (https://dotnet.microsoft.com/en-us/download/dotnet/6.0)

Before you install AVEVA Work Tasks, ensure the following conditions are met:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- System requirements are met as mentioned in the respective installation scenarios of Editions.
- AVEVA License Server is available.
- A user profile with a dbcreator account has been created for configuring the product. A Database Connection Errors while Installing Farm is required for the server on which the Workflow Engine service is installed.
- Microsoft Distributed Transaction Coordinator (MSDTC) service and configurations are enabled on your system. This is a mandatory process before installing AVEVA Work Tasks. For enabling this service, see Enable and Verify MSDTC.
- The various scenarios for installing AVEVA Work Tasks Products have been read and understood. For information about the various scenarios, see Editions.
- The computer name does not contain underscore(_). See http://support.microsoft.com/kb/909264 to view the list of characters that are not allowed for naming conventions of the computer.
- The **ISSetupPrerequisites** folder is copied along with the Installation EXE.

Additional Prerequisite for Enterprise Server

Ensure that the multicast tool is running on your computer. This tool is required to test your computer before installing the Load Balanced Server. For more information about the multicast tool, see Tools Used in Enterprise Server.

Prerequisite for Connecting to Oracle using Database Connection

You can connect to Oracle Database 12c, 18c, or 19c using the Database Connection option. However, ensure to install Oracle Client on your machine.

Note: When you upgrade the product from earlier versions to AVEVA Work Tasks 2023 SP1, copy **TNSNAME.ORA** file from <Oracle Installation path>\NETWORK\ADMIN folder to the folders:
- *C:\Program files\AVEVA\Work Tasks\Bin*
- *C:\Program files\AVEVA\Work Tasks\Web\EnterpriseConsole*
- *C:\Program Files\AVEVA\Work Tasks\Web\EnterpriseConsole\Bin*

Configure Internet Information Services

Before you install AVEVA Work Tasks, you must install and configure Internet Information Service (IIS).

IIS must have the following features enabled:

- .NET Extensibility
- ASP.NET
- ISAPI Extensions
- ISAPI Filters
- Request Filtering
- Windows Authentication
- Static Content
- HTTP Activation

Note: HTTP Activation option is enabled automatically upon installation of Microsoft .NET Framework 4.0

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

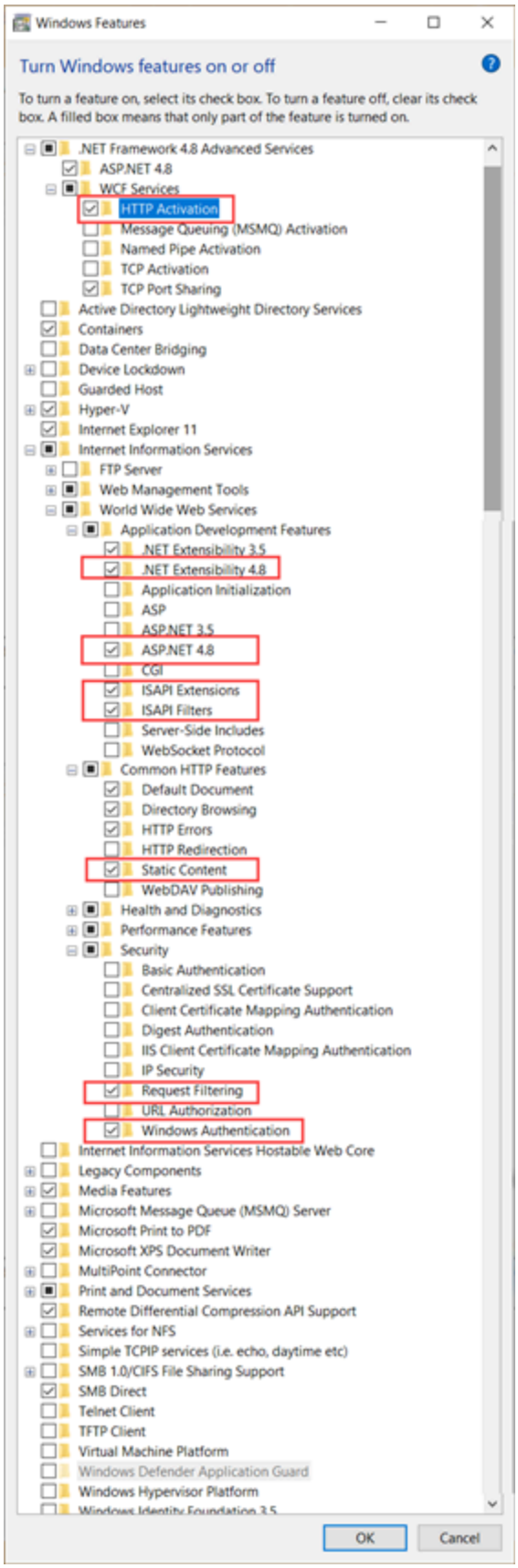
and above versions. If it is disabled, then you must enable this option manually.

Enabling features to configure IIS

Note: The above listed features are enabled when you run the Farm Configuration Wizard to configure the server/client machines for AVEVA Work Tasks. However, if those features are not enabled, you need to enable manually in the **Windows Features** dialog box.

- To enable features, open the **Control Panel**, click **Programs and Features**, and then click **Turn Windows Features on or off**. Select the features in **Windows Features** dialog box to configure IIS.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Enable and Verify MSDTC

AVEVA Work Tasks internally uses System Transactions APIs of Microsoft to perform transactions. These transactions require the MSDTC service to be configured between the database server machine and all other machines in the farm. The MSDTC service ensures that a transaction succeeds or fails as a complete unit. This is a mandatory step. AVEVA Work Tasks must be installed only after enabling the MSDTC service on your computer. Enabling MSDTC is a two-step process. You need to enable the MSDTC service before enabling MSDTC settings.

Steps to Enable MSDTC Service

1. On the Windows **Start** menu, click **Control Panel.**

   The Control Panel Items window appears.
2. Click **Administrative Tools**.
3. Click **Component Services**.

   The Component Services dialog box appears.
4. Select **Services**.

   The right pane lists all the services on the computer.
5. In the **Name** column, select **Distributed Transaction Coordinator**.
6. Check the status of the service.
7. If the **Status** column shows blank, right-click the selected service, and then click **Start**.

   The service gets started.

Note: Enable MSDTC service and MSDTC settings on the Database Server computer and on all the computers used to connect to the database server.

DTCTester Tool

The MSDTC connection settings can also be checked using the DTCTester tool. This tool is provided by Microsoft. Check for the availability and download from Microsoft download site.

The DTCTester tool does the following:

Establishes a connection to SQL Server using a data source name (DSN), User Name, and Password. You can establish this connection via a command line, using the default network library.

1. Creates a temporary table.

1. Enlists the connection in a transaction.
2. Does an insert on the temporary table.
3. Commits the distributed transaction.
4. Selects inserted value to make sure it was committed.
5. Closes the connection.

Steps to Use the DTCTester Tool:

1. Create an ODBC data source for your SQL Server through the ODBC utility in Control Panel.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Run the following from the command line:

   ```
   dtctester <dsn name><user name><password>
   ```

3. Replace the values in brackets as appropriate for your environment.

   The following table lists the ports to be opened for DTC:

Port Number	Purpose
135	RPC EPM (end point mapper)
1433	TDS SQL traffic when using TCP/IP
1434	SQL 2000 Integrated Security
5100 - 5200	MSDTC (Dynamically assigned a port by the EPM)

See the Microsoft website for updated list of suggested ports.

MSDTC Troubleshooting Articles

1. How to troubleshoot connectivity issues in MSDTC using the DTCPing tool?

   http://support.microsoft.com/kb/918331

   This article describes how to troubleshoot connectivity issues in Microsoft Distributed Transaction Coordinator (MSDTC) by using the DTCPing tool.

2. How to use DTCTester tool?

   DTCTester tests a distributed transaction against a specified Microsoft SQL Server. This tool helps to test distributed transactions across firewalls or against networks. DTCTester performs distributed transactions by using ODBC API against a SQL Server database.

3. Distributed Transactions Operations Guide

   This operations guide provides administering, configuring, managing, and monitoring information for distributed transactions in Microsoft Windows operating systems.

   See the following pages in the *Distributed Transactions Operations Guide*:

   Administering Distributed Transactions

   Troubleshooting and Resolving Transactions

4. Configuring Microsoft Distributed Transaction Coordinator (DTC) to Work through a Firewall

   http://support.microsoft.com/kb/250367

   This article describes how to configure Microsoft Distributed Transaction Coordinator (DTC) to work through firewalls. You can configure DTC to communicate through firewalls, including network address translation firewalls.

For more information on MSDTC troubleshooting, refer https://stackoverflow.com/questions/673806/msdtc-how-many-ports-are-needed.

Configure Oracle Data Handler

Before connecting to Oracle Database 12c, 18c, or 19c using the Database Connection option, you should install Oracle Client on your machine and configure AVEVA Work Tasks Oracle data handler as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Copy **TNSNAME.ORA** from **<Oracle Installation path>\NETWORK\ADMIN** folder to
   - **<AVEVA Work Tasks Installation Path>\Bin** folder
   - **<AVEVA Work Tasks Installation Path>\Web\EnterpriseConsole** folder
   - **<AVEVA Work Tasks Installation Path>\Web\EnterpriseConsole\Bin** folder
2. Include **TNSNAME** configuration in the Enterprise Console **web.config** file (this is applicable for all Enterprise Consoles where Oracle data access is required).

```
</system.web.extensions>
<oracle.manageddataaccess.client>
<version number="*">
<dataSources>
<dataSource alias= "<Connection name/SID> descriptor="(DESCRIPTION=(ADDRESS_LIST
=(ADDRESS = (PROTOCOL = TCP)(HOST = < Fully qualified host name as in TNSNAME.ORA>)(PORT
= 1521)))(CONNECT_DATA =(SERVICE_NAME = < Fully qualified service name as in
TNSNAME.ORA>)))" />
</dataSources>
<edmMappings>
<edmMapping dataType="number">
<add name="bool" precision="1"/>
<add name="byte" precision="2" />
<add name="int16" precision="5" />
</edmMapping>
</edmMappings>
</version>
</oracle.manageddataaccess.client>
</configuration>
```

If the above configuration gives error while loading Enterprise Console, then include the following configuration in the Enterprise Console **web.config** file:

```
<configSections>
    <section name="oracle.manageddataaccess.client"
     type="OracleInternal.Common.ODPMSectionHandler, Oracle.ManagedDataAccess,
Version=4.122.18.3, Culture=neutral, PublicKeyToken=89b483f429c47342"/>
  </configSections>
```

Note:
- In case of multiple Enterprise Consoles or the Enterprise Server, ensure you configure for each Enterprise Console and Server.

- When you upgrade the product from earlier versions to AVEVA Work Tasks 2023 SP1, copy **TNSNAME.ORA** file to the folder:
- *C:\Program files\AVEVA\Work Tasks\Bin*
- *C:\Program files\AVEVA\Work Tasks\Web\EnterpriseConsole*
- *C:\Program Files\AVEVA\Work Tasks\Web\EnterpriseConsole\Bin*

Install Single Server

Complete the installation tasks for the scenario that matches your configuration.

The Single Server supports the following scenarios:

- Single Machine Installation

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Web Server and AVEVA Work Tasks Server on Different Machines
- Web Server, AVEVA Work Tasks Server, and Database Server on Different Machines
- Web Farm (Multiple Web servers), AVEVA Work Tasks Server, and Database on Different Machines
- Intranet and Extranet Web Servers on Different Machines

Single Machine Installation

This section describes the activities involved while installing the Single Server for the Single Machine scenario.

Before installing ensure that all the Prerequisites are taken care of.

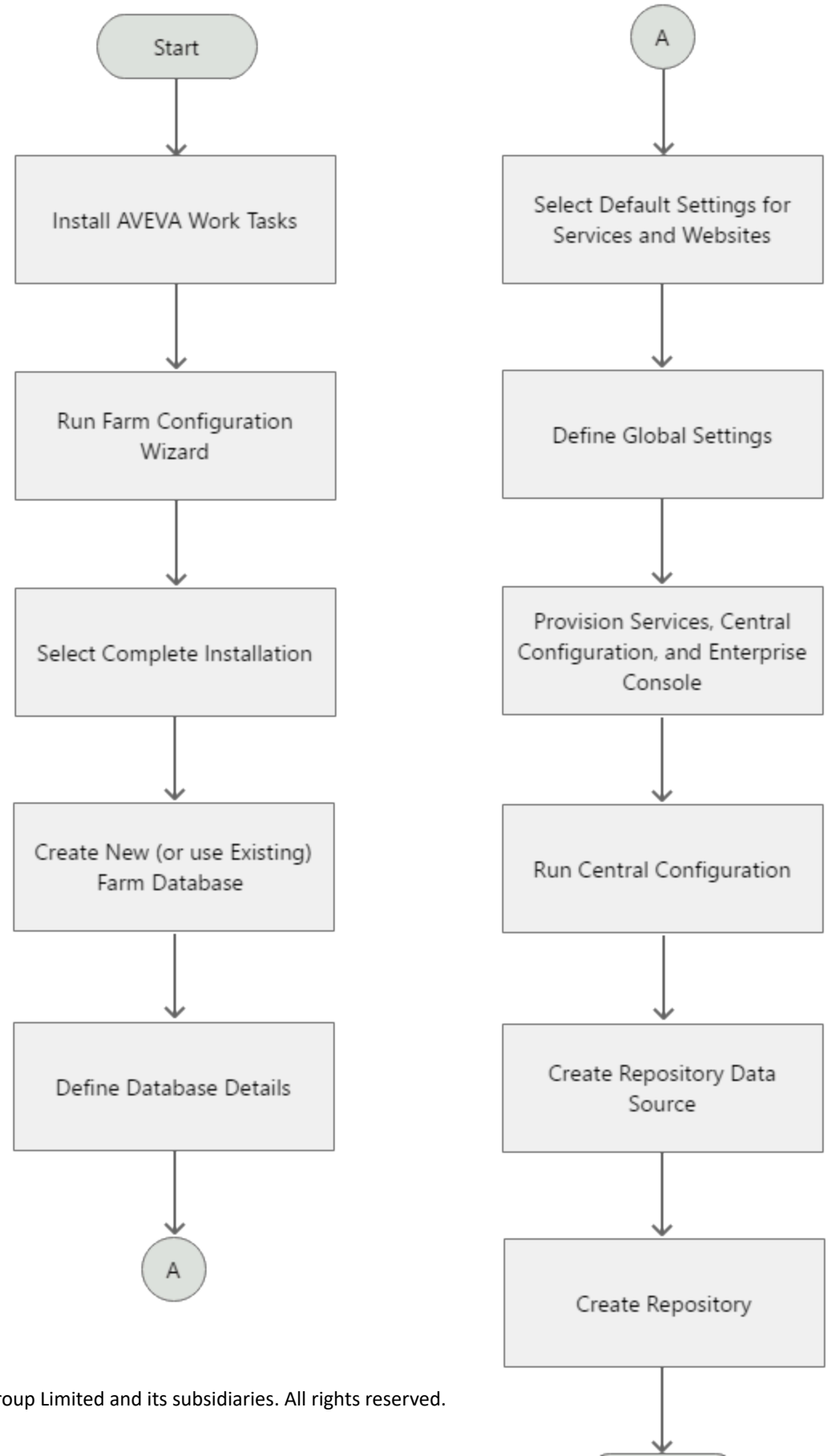
See Also: Deployment of AVEVA Work Tasks Web Server, AVEVA Work Tasks Server, and AVEVA Work Tasks Database Server on a Single Machine

High-Level View of Installation

Before continuing with the installation it is presumed, that you have read and understood the Prerequisites section.

The following flowchart illustrates a high-level view of the various installation steps with an overview of the various activities involved while installing AVEVA Work Tasks . Using this flowchart you can visually keep track of your activities while following the sequence of tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation

This section provides installation steps for the Single Machine scenario.

In this scenario, AVEVA Work Tasks Server is installed on a single machine.

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
4. Click **Next**.

   The License Server details are configured successfully.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.
4. Click **Next**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Edit Farm Settings** window appears.

5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.

6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.

9. Click **Next**.

   The License Server details are configured successfully.

Select the type of installation

To select the type of installation:

1. In the Select the type of installation page, select **Complete Installation (Server)**.

   A Complete Standalone Deployment does not require a client installation.

2. Under **Create AVEVA Work Tasks Farm**:
   - Click **Create AVEVA Work Tasks Farm database** to create a new Farm database.
   - Click **Use existing AVEVA Work Tasks Farm database (Creates AVEVA Work Tasks Farm tables only if they do not exist)** to use an existing database as the Farm database.
3. Click **Next**.

Create Farm Database

Use the Create AVEVA Work Tasks Farm Database page to provide **New Database** details. Select the **Connection String** option to establish a connection with the database. This connection information is encrypted and stored.

You can enter either Windows or Microsoft SQL Server user settings for authentication.

Note:
- The Farm database password does not support **&** and **;** special characters.
- While setting up **Farm Database** with **Windows Authentication**, do not enter **FQDN** in the **Domain Name** field.

To Create Farm Database

1. Enter **Database Title**.
2. Select **Connection String**.

   See *Connection Provider* later in this topic, if you have selected the **Connection Provider** option.

   - Enter the following details:

     Server Name:

     Provide information about the database server on which the Farm database can be installed.

     If the **Database Type** is **SQL Server**, then you can specify an IPv6 address for the **Server Name**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If SQLExpress Instance need to be used as a Database instance of the Farm database, then **Server Name** should be <server name>\SQLExpress.

To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

- **Authentication Type**

  This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports Windows and SQL Server authentication process.

  Example: Windows
  If Windows Authentication is selected, the user configured for the Farm configuration will be used to access the database. This is the authentication recommended by AVEVA Work Tasks.

- **Login**

  This field is used to enter the SQL login credential for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

- **Password**

  This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

- **Advanced Settings**

  This is an optional field. It is used to enter any additional connection information, such as connection Timeout. Example: Connection Timeout = 20.

  While creating a data source for repository, use the following custom settings to set properties for connection pool. These settings are case sensitive and should be mentioned only in lower case.

  ```
  encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
  ```

  encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

  enableltm: AVEVA Work Tasks internally implements light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive. This property can be used to turn off this feature. The default value is set as True.

  connectionretrywaittime: While opening a new connection, if the data base server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

  maxconnectionretries: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

  sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

3. Click **Test Connection** to ensure database connectivity.
4. Click **Next**.

   The message *AVEVA Work Tasks Farm is Created* appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Click **OK**.

AlwaysOn Availability Groups feature of Microsoft SQL Server

The **AlwaysOn Availability Groups** feature of Microsoft SQL Server is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in Microsoft SQL Server 2012, AlwaysOn Availability Groups maximizes the availability of a set of user databases for an enterprise.

After you configure the AVEVA Work Tasks, you must add the created Farm Database to the SQL Server AlwaysOn Availability Groups. Contact your Database Administrator to add the database to SQL Server AlwaysOn Availability Groups.

Notes:

- The authentication user needs different rights assigned for different scenarios. If a user wants to create database (Farm or Repository Data Source Database), Authentication User needs to have minimum 'dbcreator' access rights on Database Server Instance.

- If database is already created, user needs to have following permissions on Database Server:

- SELECT, INSERT, UPDATE, and DELETE permissions.

- EXECUTE permission for Stored Procedures.

To choose Connection Provider

Complete the following steps to choose a connection provider:

1. Enter **Database Title**.
2. In the **Database Type** box, select a database type.
3. Select **Connection Provider**.

   This option is used to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.
4. Enter the following details:
   - **Assembly Path**:

     Click the icon to browse and choose the required dll file.

     If custom properties (enableltm, connectionretrywaittime, maxconnectionretries and sqlcommandtimeout) are required to be used in Assembly connection string, then user should select **Use existing Farm database** while creating a the Farm. See Creating or Using an Existing Database.

     See the topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub chapter in the *Build Business Process* section for creating the assembly. Make sure that, there should be only one class implementing *IDBConnectionString* interface within this assembly.
5. Click **Test Connection** to ensure database connectivity.
6. Click **Next**.
7. The message *AVEVA Work Tasks Farm is created* appears.
8. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Define Security Settings

You can define the Security Settings using the Farm Configuration Wizard.

For more information on Security settings, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client.

To define the Security Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Security Settings** and **Service Settings** and then, click **Next**.
4. Select **Secure Communication** to establish secured communication between the AVEVA Work Tasks services through HTTPS.
5. Select **Secure Web Sites** check box to configure secured web sites. Uncheck the **Secure Web Sites** check box to use HTTP sites. By default the sites are created in HTTPS mode.
6. Enter the Port number. The default port for the HTTPS site is 8001, and for the HTTP site is 8000. You can enter any port number that IIS accepts. But if any other services or applications are already using the entered port number, AVEVA Work Tasks will not start.
7. Select the available certificate from the **Certificate** list. If your certificate is not listed, click the **Refresh** icon to repopulate the list.

   Note: If the required certificate is not listed, generate one using the System Management Server (Configurator tool). Do not use a self-signed certificate without a proper root chain.
8. Click the **Validate** button to validate the certificate before configuring the security settings.

   Note: An Internet connection is needed to validate the certificate.
9. Click **Configure**, then **Next**, and then **Finish** to apply the new security settings.

Note:
- To update the port number, certificate, and websites from unsecure to secure or vice-versa, select **Security Settings** and **Service Settings** check box in **Edit AVEVA Work Task Farm Settings** window before updating the values. After the required details are provided, these settings get updated only when you click **Configure**, then **Next**, and then **Finish** as you navigate through the Farm Configuration Wizard..
- If the Quickflow is already deployed on a machine, you need to deploy it again after you update the security settings. If secured web site is configured, then you cannot change the server configuration for Quickflow in Central Configuration. However, if the **Secure Web Sites** checkbox is not selected, you can make the changes to the server configuration.
- Advance Server communication cannot be configured on HTTP/HTTPS.
- For resources of Load Balanced Server, ensure to configure all services with the same protocol (HTTP or HTTPS).
- It is recommended to use Standard SSL certificate instead of self-signed SSL certificate.

Using ASB Certificate for Secured Communication

ASB Certificate is a PCS-managed SSL certificate and renews automatically every 30 days when the certificate is still valid. The certificate is valid for two years. The PCS watchdog service on the runtime node checks the validity

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

of the SSL certificate within 5 minutes from when it starts and once a day. PCS also renews the SSL certificate if it is expired.

If the runtime node is offline and later back online, the PCS watchdog service regularly checks the certificate validity and renews it if it is required. Consider restarting the PCS watchdog service to enforce the SSL certificate validity check within 5 minutes from the restarting time.

Furthermore, when the SSL certificate is renewed, PCS binds it to port 443 (by default) and changes the SslThumbprint PCS registry value. If AVEVA Work Tasks sites are hosted on port 443, certificate binding will be auto-renewed when the SSL certificate is renewed. It is recommended to check the bindings in IIS with every certificate renewal to ensure the application is running seamlessly.

To disable the certificate renewal, consider disabling the **ArchestrA Certificate Renewal Service** windows service.

Note: Client Service will try to automatically bind the renewed certificate. If there are any issues in browsing the website or starting the services after certificate renewal, users need to reconfigure the secure communication in Farm Configuration. To reconfigure, see the steps mentioned in above section *To define the Security Settings*.

Securing Communication between Services

To have a secure communication between the services, apply the below changes to the Windows registry and reboot the server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
```

For more information on securing communication, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client and

https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2015/2960358.

Define Services and Global Settings

You can define the Services and Global Settings using the Farm Configuration Wizard or through Configuration XMLs.

To define Services

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: You can also customize the settings for these services and websites using the **Custom Settings** option. For more information about Service-related changes and Website-related changes, see Advanced Settings and Administering Web Servers respectively.

2. Click **Next**.

   The Global Settings dialog box appears.

3. In the **Communication** tab, enter key fields in **Email Settings** and **Exception Email settings** to configure exception e-mails.

   The following are the details:

   - **Host**: Enter information related to mail server.
     Example: Enter details related to Microsoft Exchange Server (IP address of the exchange server).
   - **Port**: Enter the port number for the mail server. The default port number is 25.
   - **From email**: Enter the from e-mail address.
     Example: admin@telematics.com
   - **To email**: Enter the to e-mail address.
     Example: alex@telematics.com

4. Click **OK**.

Note: To confirm the exception e-mail settings, click **Test SMTP**. An e-mail will be generated and sent to the **To email** address confirming successful configuration of the e-mail. The advanced settings like **Secure Socket Layer (SSL), Email Timeout, User Name,** and **Password** for the e-mail account holder, **Password Encrypted** flag will differ based on the mail server settings.

Provision Services

While the installation is in progress, you can view the services that will be installed. This provides information on the progress of installation. At the end of the installation, a status indicating the completion of installation appears.

- Click **OK** to complete the installation.

Create a Repository DataSource

A **Repository DataSource** is used to store all repository related data. The **Repository DataSource** must be created on a machine where the **Central Configuration** site is installed.

Example

A TelematicsDataSource is created for TelematicsHR Repository.

A Repository contains Workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating the repository.

To create a Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**.

   The Central Configuration web page appears.

2. Click **DataSource Settings**.

   The Datasource List page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Click **Add**.

   The Manage Datasource page appears.

4. Enter the required details.

   a. **Name**: Enter a valid data source name. Example: *TelematicsDataSource*.

   b. **Description**: Enter a description for the data source. Example: *This data source is used to store connection details*.

   c. **Connection String**: Select this option to establish a connection with the database. The connection string contains fields that that defines the connection. The connection information is encrypted and stored.

   Server Name

   To use a database on the local machine, enter "localhost", ".", or the local computer's name for the **Server Name**.

   Note: Server Name can be machine name/IP address/SQL Server AlwaysON/SQL Server instance name. Instead of entering "local host", it is preferable to give the local machine name.

   If the **Database Type** is **SQL Server**, then specify an IPV6/IPV4 address for the **Server Name**.
   To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: *localhost,1455*
   To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be *<server name>\SQLExpress*.

   Authentication Type

   Use this field to determine the process to validate a user who is trying to connect to the database. AVEVA Work Tasks uses Windows and SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. The **User ID** and **Password** field is disabled if Windows Authentication is selected. Example: *Windows*

   Advanced Settings

   Use this field to define advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection.
   Example: *Connection Timeout = 20*
   or
   For defining maximum connections that can be opened in a connection pool.
   Example: *max pool size = 250*
   Default connection pool is set as 250. We recommend to enter a higher value on heavily loaded environment.
   While creating a data source for a repository, use the following custom settings to set properties for the connection pool.
   encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
   encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.
   enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature, AVEVA Work Tasks controls the transaction from escalating to an expensive distributed transaction.
   The above property can be used to turn off this feature. The default value is set as *True*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is 30 seconds. The value should be set in seconds.
maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is infinite.
sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.
These settings are case sensitive and should be mentioned only in lower case.

Database:

Specify or select a database that can be used as a repository database. This database stores all repository related data. The repository database is created if it does not exist on the database server instance.
Example: *TelematicsDB*

Enable Table Partitioning:

Select to enable database table partitioning. For more information, see Partitioning Database.

d. **Connection Provider**: Use this to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.

The selected dll must be copied to all the machines, all servers and all clients, that are connected to the current Farm.

Assembly Path

Click to browse and choose the required dll file.
To use custom properties (enableltm, connectionretrywaittime, maxconnectionretries, and sqlcommandtimeout) in the Assembly connection string, the database must be created before.
See *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Developer Guide* for creating the assembly. Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.

   A message about successful test connection appears.

6. Click **OK**.

   A message about successful creation of the data source appears.The message also states that the required tables will be created in the data source after a repository is associated with the data source.

When a data source is saved, an entry is created for the data source in the **Farm Database**. When a first repository is created for the data source, AVEVA Work Tasks tables are added to the data source. While creating a data source, the archival database is created with the same database name suffixed by *_BI*. This database is used for archival in future use.

Note: After you configure the AVEVA Work Tasks, you must add the created **Farm Database** to the **SQL Server AlwaysOn Availability Groups**. Contact your Database Administrator to add the database to **SQL Server AlwaysOn Availability Groups**.

To view the data source created, click **DataSource Settings** link.

Create a Repository

A **Repository** contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

To create a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The Central Configuration page appears.
2. Click **Repository Settings**.
   The Repository List page appears.
3. Click **Add**.

   Note: If the number of repositories exceeds the permitted limit for the license, then the message "You have exceeded the maximum number of repositories" appears, and the repository creation is stopped. Click **Close**.

   The **Manage Repository** page appears.
4. Enter the required details.
5. Click **OK**.
   The **Config Administrator** page appears.
6. Click to configure an administrator.
   The User Lookup dialog box appears.
7. Select the provider from the provider drop-down list.
8. Enter the details to search the administrator for the repository that is created.

   Note: You can also search a user based on *ObjectSID* and *objectGUID* (applicable only for Active Directory provider).
9. Click **Search**.
   The user is listed under **Available Resources**.
10. Select and click the arrow to list the user under **Selected Resource**.
11. Click **Update**.
    The selected user is displayed in **User Lookup**.
12. Click **OK**.
    A message confirming the successful creation of an administrator appears.
13. Click **OK**.
    The **Repository List** page appears.
    For more information about configuring the system user, see Configure System User.

Manage Repository Window Fields

The following table explains the fields of the Manage Repository window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Repository DataSource	Lists all the data sources created in a Farm. The selected data source will be used to store data for the current repository.	TelematicsDatasource
Attach DataSource	Select Attach DataSource checkbox and then select a repository from the selected DataSource .Data associated with the selected repository will be copied to the new repository. The version of the installed product and destination DataSource should be the same as that of the DataSource from which the repository data is to be copied. For example, in an upgraded environment, from build Work Tasks version 1 to Work Tasks version 2, you cannot attach a DataSource created in build Work Tasks version 1 to build Work Tasks version 2. To match the build version of the installed product to the version of the DataSource: 1. If the DataSource was created in build Work Tasks version 1, then the entire build Work Tasks version 1 must be upgraded to Work Tasks version 2 by running the product installer in the upgrade mode. 2. All the Farm, DataSources, Repositories, and Lists of build Work Tasks version 1 must be upgraded to build Work Tasks version 2 by running the AVEVAWorkTasksUpgradeManager.exe located at the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder. 3. After upgrading all the AVEVA	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	Work Tasks components to build Work Tasks version 2 build, that DataSource can be used in any machine that has same version by using this feature.	
Repository Name	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the repository name.	Telematics Repository
Description	Provide a description for the repository.	This stores Telematics Employee Leave Details.
From mail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might occur in the Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be sent to the mail ID defined in **Exception to mail**. The From address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To avoid this, restart the Workflow Engine Service and Communication Service.	Provide the valid email id that is specified in the **User Name** field while configuring **SMTP Settings** in the **Mail Server Settings** page. For more details, see Mail Server Setting.
Exception to mail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails is sent to the email ID specified in **Exception to mail**.	Provide the valid email id.
Alert initiator of the Workflow	Determines if the status of a	A process with a rule to check the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	workflow should be sent as a notification mail to the initiator of the workflow.	pending leave available for the employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
Servers	Lists the servers available in the Farm. Repository will be available only to the mapped servers.	Work Tasks Server Name
Enterprise Consoles	Lists all the Enterprise Consoles for a selected AVEVA Work Tasks Server. Repository will be available only to the mapped Enterprise Console.	http://worktaskservername:8001/EnterpriseConsole
Enable Theme	Select this checkbox to enable the defined theme.	-
Theme Name	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. It provides **Default** and **Lite** themes.	-
Resource Provider	Lists the various default resource providers supported by AVEVA Work Tasks. Resource providers are also termed as User Provider. A User Provider is a data store containing information about the user. AVEVA Work Tasks also supports custom resource provider. For more information about Resource Provider, see User Providers. AVEVA Connect and Azure Active Directory resource providers are also available after you configure them. To configure these providers, see AVEVA Connect Settings and Azure Active Directory Settings. AVEVA Work Tasks allows secured LDAP configuration. The default	AVEVA Work Tasks Lists or Active Directory

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	port number for LDAPS is 636. You can map the repository in IDE and use the Secured LDAP. The logged-in Active Directory is displayed by default, and **Active Directory Datasource** expects **LDAP://** or **Fully Qualified Domain Name** as value.	
Custom Application Folder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with a semicolon.	-
Hide Repository	Select this option to make the repository unavailable to the Farm.	-
Source path	Enter the path to the Source Elements Folder.	-
Attachment Storage Path	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an absolute path. - The default path is the **Attachments** folder in *C:\ProgramData\AVEVA Work Tasks*. - The default path is used when no path is specified. - Ensure appropriate rights are set to the folder, where attachments are stored, for the following identities: -- The application pool identity for **Central Configuration** and **Enterprise Console(s)**. -- The account configured for the **Workflow Engine** service. - In case of mapped drives, ensure they are available to the identities. - In multiple-server environments, the path specified must be in a shared location.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Auto Log Off Settings for Enterprise Console	Automatically logs out the logged in user after the designated period of inactivity. Enter the time in minutes. This field accepts only valid integer values between 0 to 360. 0 minute means the user will not get automatically logged off. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. The default value for automatic log off is 30 minutes. You use this field to define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML.	0 to 360 (minutes)
Auto Log Off Settings for AVEVA Work Tasks Pro	Request for user credentials for AVEVA Work Tasks Pro app after the set time is elapsed. Enter the time in minutes. This field accepts only valid integer values between 0 to 2880. 0 minute means the user will not get automatically logged off. The default value is 30 minutes. If AVEVA Work Tasks Pro app is offline, the Auto Log Off will not get active, and the user will not be automatically logged off as per the defined settings.	0 to 2880 (in minutes)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Offline Support	Select this option to enable offline submission of items in AVEVA Work Tasks Pro app.	-
Refresh on Work Item submission	Select this option to enable Inbox refresh on Work Item submission in AVEVA Work Tasks Pro . Refresh option is enabled by default.	-
Timeout	Enter the time in seconds to set the timeout. If the value is set as 0, then there is no timeout. Default value is 60 seconds.	-
Label for Fill Form	Enter the label for Fill Form. The customized label will be displayed for the Fill Form in AVEVA Work Tasks Pro.	Work Tasks
Date Configuration Flag for Alternate Actors	Select the **Enable Date Settings** option to set the **From** date and **To** date as mandatory while creating alternate actors. For more information about creating alternate actors, see *Alternate Actors* topic in the *User Guide*.	-
Display Domain name with User Name	Select this checkbox to display the domain name along with the user name in the Enterprise Console and the List. The domain name will be displayed only: • If this checkbox is selected. • If the **Resource Provider** is selected as **Active Directory**.	-

User Providers

AVEVA Work Tasks supports the following User Providers by default:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Active Directory

When you select **Active Directory** as the **Resource Provider**, details such as **Active Directory datasource** and the **Active Directory search base** are populated by default.

Authentication User, **Authentication Password**, **User Log-on ID**, and **User Log-on Display Name** are optional fields. Access to the active directory might not be available for all users. In such cases, provide the active directory authentication user name and the active directory authentication password with which you can connect to the Active Directory.

AVEVA Work Tasks considers *SAM-Account-Name* as **User Log-on ID** by default. If you want to use active directory attributes other than the *SAM-Account-Name* user, then enter the desired active directory attribute details in the **User Log-on ID** field and **User Log-on Display Name** field. **User Log-on Display Name** is the user friendly display name of the active directory attribute. This **User Log-on Display Name** is referred in the User Lookup dialog box against the **User Log-on ID**.

Steps

1. When you select **Active Directory** as the **Resource Provider**, then the **Use Token Service** field appears. In this field, select one of the following:
   - If you do not want to use AIM authentication, select **Without Token Service**.
   - If you want to use AIM authentication, then select a server from the list of alias server names.
2. Enter **Active Directory datasource**.
3. Enter **Active Directory search base**.
4. Enter **Domain Name**.

   The domain name is case-sensitive.
5. Enter **Authentication User**.
6. Enter **Authentication Password**.
7. Check **Is Cross Domain Role Provider** checkbox, if cross domain users and roles mapping is required.

   Example: Users *John* and *Jim* of domain *TelematicsU* and users *Mike* and *Bob* from domain *TelematicsAU* are added to role Team HR of domain *TelematicsMain*.
8. Enter **User Log-on ID** if you want to use another Active Directory attributes instead of *SAM-Account-Name*.

   We recommend to use attributes of type *string* as input for **User Log-on ID.** However, *ObjectSID* and *objectGUID* are also supported.

   After you set the **User Log-on ID**, you cannot change it later.
9. Enter the **User Log-on Display Name**.

   After you set the **User Log-on Display Name**, you cannot change it later.

   This is the default display name of the user account of the active directory user used. **User Log-on Display Name** is used in user lookups.
10. Click **Test LDAP**.

    A message about successful LDAP test appears.

Note:
- Some of the active directory attributes or properties are not indexed. If you use an active directory attribute that is not indexed, it might impact the log-on validation performance based on the active directory entries. We recommend to use indexed attributes as **User Log-on ID** or index the attributes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- For an upgraded repository, you cannot change both **User Log-on ID** and **User Log-on Display Name**.
- You cannot localize **User Log-on ID** and **User log-on Display Name** fields.
- Disabled users will not be shown in user lookup search.

Using Forms

For Forms Resource Provider:

- If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.

1. In the **Resource Provider** drop-down list**,** select **Forms.**
2. In the **User Entity properties** field, enter XML data.

   Click **Sample Code** to get the XML Schema Template.

   Example

   ```
   <properties>
   <property name="userid" type="string" purpose="identifier" />
   <property name="username" type="string" purpose="name" />
   <property name="pwd" type="string" purpose="password" />
   </properties>
   ```
3. In the **User Provider Properties** field, enter the same XML Schema.

   There must be at least one identifier in the XML Schema. For example, UserID
4. In the **Datasource** field, enter a valid data source or specify a dll.

   The value for the data source can be specified as **Physical path of the dll** or **Physical path of the dll::param. Param** is not mandatory.

   See topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Build Business Process* section for creating the assembly.

   Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly, whereas one class can return multiple connection strings.

   Examples of data source parameter:

   - c:\DLLStore\GetConnectionString.dll
   - c:\GetConnectionString.dll::Leave

   If there are two data sources, *Leave* and *Licensing*, and only one DLL is used to return a connection string based on the data source selected. The parameter can be specified as **c:\GetConnectionString.dll::Leave** or **c:\ GetConnectionString.dll::Licensing**. The connection string can be returned from the connection dll based on this parameter.
5. Enter a proper table name or view name which holds the User Name and Password for the Authentication purpose.

   Example: Passmast (the table which is available in IMS_NEW Database)
6. Click **Test SQL Connection**.

   A message about successful test connection appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using ASP.NET Membership

This resource provider lists users defined in the asp.net membership database.

Lists

This resource provider lists users created using AVEVA Work Tasks Users Creation. If you are unable to determine a User Provider to create a repository, use AVEVA Work Tasks Lists.

To verify successful creation of the Repository:

1. Click **Enterprise Console** URL from the site path column.
   The repository created is listed.
   Add the created Enterprise Console link into "Add to Favorites" to easily navigate in future.
2. Click **Repository User Provider** link to logon to the repository.
   The Enterprise Console log-on page appears.
3. Enter the logon details, and then click **Login**.
   The Enterprise Console home page appears. This indicates successful creation of a Repository.
   All features can only be available for a user with administrative privileges.

Note: Consider you created the Repository and then deleted it. If you create the same Repository again immediately and try to log into it through Enterprise Console, the Provider does not list the Repository. You need to access IIS and recycle AVEVA Work Tasks from the Application Pool to get the Repository listed in the Provider list..

Verify the Installation

After you install AVEVA Work Tasks, verify that the product is installed successfully. To verify successful installation, create a simple Workflow using Process Designer. The tasks includes the following:

- Creating a Workflow
- Publishing the Workflow
- Testing the Workflow

Creating a Workflow

1. Log on to the **Enterprise Console**.
2. If the Repository is configured for Active Directory provider, go to step 4.
3. Enter the **User Name** and **Password**.
   The **User Name** and **Password** will be validated with the user provider selected while creating the repository.
   The default authentication provider is the Repository User Provider List.
4. In the **Enterprise Console**, click menu icon.
   All features can only be available for a user with administrative privileges.
5. From the menu, select **Workflows**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Workflow window appears.

6. On the Action bar, click **New**, and then click **Workflow**.
   The New Workflow window appears.
7. Enter **Title** and **Description**.
8. Click **Save & Continue**.
   The successful message appears.
9. Select **Design the Workflow now**.
   This is the default option.
10. Click **Finish**.
    The Process Designer window appears.
11. From the activities pane on the left, drag the Information activity to the Process Designer window.
12. The activity *Information1* is created.
    Click to create a link between two activities.
    The Link Action window appears.
    a. Select from the **From Action** drop-down box.
    b. Select from the **To Action** drop-down box.
    c. Click **OK**.
       The link is established.
13. Right-click *Information1*.
14. Select **Activity Properties** from the context menu.
    The Activity Properties pane appears.
    a. Enter **Name** and **Description**.
    b. Click to **Assign Actor(s)**.
       The Assign Actor(s) window appears.
    c. Select the required provider from the **Providers** drop-down list.
    d. Select values from the drop-down list.
    e. Click **Add**.
    f. Click **Update**.
15. Click to enter the **Subject** for Notification Contents.
    The window to enter subject for notification appears.
16. Click **Update**.
17. Click to enter the **Body** for Notification Contents.
    The window to enter the Body content appears.
18. Click **Update**.
19. From the Activity Properties pane, click **Save**.

Publishing the Workflow

1. On the **Publish** menu, click **Publish**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Published successful message appears.

2. Close the dialog box.

   The Enterprise Console screen lists the workflow created.

Testing the Workflow

1. Select the listed workflow.
2. Right-click the selected workflow.
3. Click **Publish and Test Run**.

   The **Status Finished successfully** appears.

   If Status is **Execution Pending** in the Process Execution window, then click on the **Refresh** button to get the status.
4. In the **Enterprise Console** menu, select **Inbox**.

   The Inbox lists the work item.
5. Click to view the work item listed.

   This indicates that the installation is successful.

Web Server and AVEVA Work Tasks Server on Different Machines

This section describes the activities involved while installing the Single Server for the scenario Web Server and AVEVA Work Tasks Server on Machine1 and Database on Machine2.

In this scenario, AVEVA Work Tasks Server is installed on Machine 1 and the Database is installed on Machine 2.

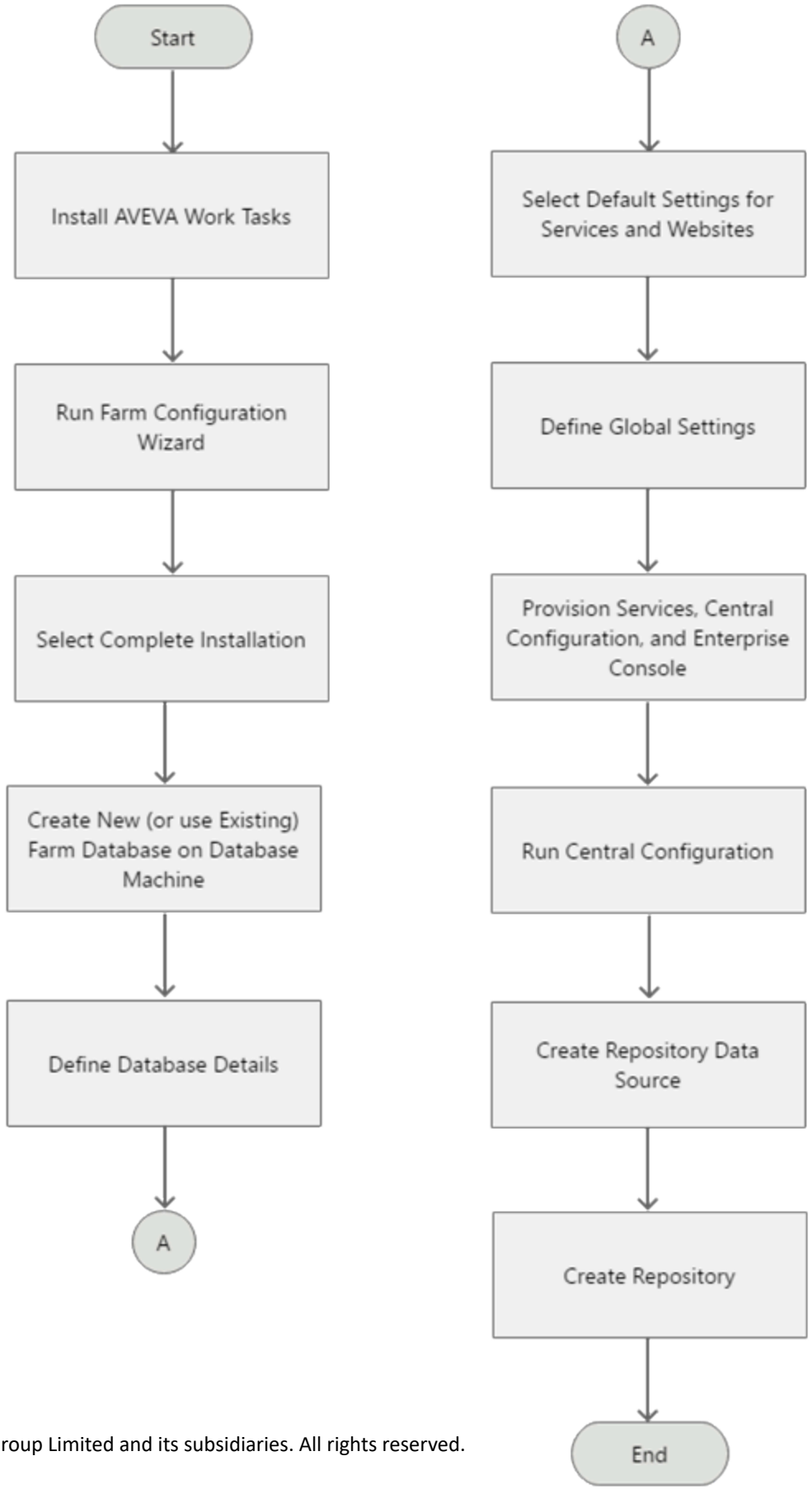
See Also: AVEVA Work Tasks Web Server and AVEVA Work Tasks Server on Different Machines

High-Level View of the Installation

Before continuing with the installation, it is presumed that you have read and understood the Prerequisites section.

The following flowchart illustrates a high-level view of the various installation steps with an overview of the various activities involved while installing AVEVA Work Tasks. Using this flowchart you can visually keep track of your activities while following the sequence of tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation

This section provides installation steps for the scenario the Web Server and the AVEVA Work Tasks Server on Machine 1 and the Database on Machine 2.

In this scenario the AVEVA Work Tasks Server is installed on Machine 1 and the Database is installed on Machine 2.

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
4. Click **Next**.

   The License Server details are configured successfully.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click **Next**.

   The **Edit Farm Settings** window appears.

5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.

6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.

7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.

8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.

9. Click **Next**.

   The License Server details are configured successfully.

Select the type of installation

To select the type of installation:

1. In the Select the type of installation page, select **Complete Installation (Server)**.

   A Complete Standalone Deployment does not require a client installation.

2. Under **Create AVEVA Work Tasks Farm**:
   - Click **Create AVEVA Work Tasks Farm database** to create a new Farm database.
   - Click **Use existing AVEVA Work Tasks Farm database (Creates AVEVA Work Tasks Farm tables only if they do not exist)** to use an existing database as the Farm database.

3. Click **Next**.

Create Farm Database

Use the Create AVEVA Work Tasks Farm Database page to provide **New Database** details. Select the **Connection String** option to establish a connection with the database. This connection information is encrypted and stored.

You can enter either Windows or Microsoft SQL Server user settings for authentication.

Note:
- The Farm database password does not support **&** and **;** special characters.
- While setting up **Farm Database** with **Windows Authentication**, do not enter **FQDN** in the **Domain Name** field.

To Create Farm Database

1. Enter **Database Title**.
2. Select **Connection String**.

   See *Connection Provider* later in this topic, if you have selected the **Connection Provider** option.

   - Enter the following details:

     Server Name:

     Provide information about the database server on which the Farm database can be installed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If the **Database Type** is **SQL Server**, then you can specify an IPv6 address for the **Server Name**.

If SQLExpress Instance need to be used as a Database instance of the Farm database, then **Server Name** should be <server name>\SQLExpress.

To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

- **Authentication Type**

  This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports Windows and SQL Server authentication process.

  Example: Windows
  If Windows Authentication is selected, the user configured for the Farm configuration will be used to access the database. This is the authentication recommended by AVEVA Work Tasks.

- **Login**

  This field is used to enter the SQL login credential for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

- **Password**

  This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

- **Advanced Settings**

  This is an optional field. It is used to enter any additional connection information, such as connection Timeout. Example: Connection Timeout = 20.

  While creating a data source for repository, use the following custom settings to set properties for connection pool. These settings are case sensitive and should be mentioned only in lower case.

  ```
  encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
  ```

  encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

  enableltm: AVEVA Work Tasks internally implements light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive. This property can be used to turn off this feature. The default value is set as True.

  connectionretrywaittime: While opening a new connection, if the data base server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

  maxconnectionretries: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

  sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

3. Click **Test Connection** to ensure database connectivity.
4. Click **Next**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The message *AVEVA Work Tasks Farm is Created* appears.

5. Click **OK**.

AlwaysOn Availability Groups feature of Microsoft SQL Server

The **AlwaysOn Availability Groups** feature of Microsoft SQL Server is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in Microsoft SQL Server 2012, AlwaysOn Availability Groups maximizes the availability of a set of user databases for an enterprise.

After you configure the AVEVA Work Tasks, you must add the created Farm Database to the SQL Server AlwaysOn Availability Groups. Contact your Database Administrator to add the database to SQL Server AlwaysOn Availability Groups.

Notes:

- The authentication user needs different rights assigned for different scenarios. If a user wants to create database (Farm or Repository Data Source Database), Authentication User needs to have minimum 'dbcreator' access rights on Database Server Instance.

- If database is already created, user needs to have following permissions on Database Server:

- SELECT, INSERT, UPDATE, and DELETE permissions.

- EXECUTE permission for Stored Procedures.

To choose Connection Provider

Complete the following steps to choose a connection provider:

1. Enter **Database Title**.
2. In the **Database Type** box, select a database type.
3. Select **Connection Provider**.

   This option is used to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.
4. Enter the following details:
   - **Assembly Path**:

     Click the icon to browse and choose the required dll file.

     If custom properties (enableltm, connectionretrywaittime, maxconnectionretries and sqlcommandtimeout) are required to be used in Assembly connection string, then user should select **Use existing Farm database** while creating a the Farm. See Creating or Using an Existing Database.

     See the topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub chapter in the *Build Business Process* section for creating the assembly. Make sure that, there should be only one class implementing *IDBConnectionString* interface within this assembly.
5. Click **Test Connection** to ensure database connectivity.
6. Click **Next**.
7. The message *AVEVA Work Tasks Farm is created* appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

8. Click **OK**.

Define Security Settings

You can define the Security Settings using the Farm Configuration Wizard.

For more information on Security settings, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client.

To define the Security Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Security Settings** and **Service Settings** and then, click **Next**.
4. Select **Secure Communication** to establish secured communication between the AVEVA Work Tasks services through HTTPS.
5. Select **Secure Web Sites** check box to configure secured web sites. Uncheck the **Secure Web Sites** check box to use HTTP sites. By default the sites are created in HTTPS mode.
6. Enter the Port number. The default port for the HTTPS site is 8001, and for the HTTP site is 8000. You can enter any port number that IIS accepts. But if any other services or applications are already using the entered port number, AVEVA Work Tasks will not start.
7. Select the available certificate from the **Certificate** list. If your certificate is not listed, click the **Refresh** icon to repopulate the list.

   Note: If the required certificate is not listed, generate one using the System Management Server (Configurator tool). Do not use a self-signed certificate without a proper root chain.
8. Click the **Validate** button to validate the certificate before configuring the security settings.

   Note: An Internet connection is needed to validate the certificate.
9. Click **Configure**, then **Next**, and then **Finish** to apply the new security settings.

Note:
- To update the port number, certificate, and websites from unsecure to secure or vice-versa, select **Security Settings** and **Service Settings** check box in **Edit AVEVA Work Task Farm Settings** window before updating the values. After the required details are provided, these settings get updated only when you click **Configure**, then **Next**, and then **Finish** as you navigate through the Farm Configuration Wizard..
- If the Quickflow is already deployed on a machine, you need to deploy it again after you update the security settings. If secured web site is configured, then you cannot change the server configuration for Quickflow in Central Configuration. However, if the **Secure Web Sites** checkbox is not selected, you can make the changes to the server configuration.
- Advance Server communication cannot be configured on HTTP/HTTPS.
- For resources of Load Balanced Server, ensure to configure all services with the same protocol (HTTP or HTTPS).
- It is recommended to use Standard SSL certificate instead of self-signed SSL certificate.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using ASB Certificate for Secured Communication

ASB Certificate is a PCS-managed SSL certificate and renews automatically every 30 days when the certificate is still valid. The certificate is valid for two years. The PCS watchdog service on the runtime node checks the validity of the SSL certificate within 5 minutes from when it starts and once a day. PCS also renews the SSL certificate if it is expired.

If the runtime node is offline and later back online, the PCS watchdog service regularly checks the certificate validity and renews it if it is required. Consider restarting the PCS watchdog service to enforce the SSL certificate validity check within 5 minutes from the restarting time.

Furthermore, when the SSL certificate is renewed, PCS binds it to port 443 (by default) and changes the SslThumbprint PCS registry value. If AVEVA Work Tasks sites are hosted on port 443, certificate binding will be auto-renewed when the SSL certificate is renewed. It is recommended to check the bindings in IIS with every certificate renewal to ensure the application is running seamlessly.

To disable the certificate renewal, consider disabling the **ArchestrA Certificate Renewal Service** windows service.

Note: Client Service will try to automatically bind the renewed certificate. If there are any issues in browsing the website or starting the services after certificate renewal, users need to reconfigure the secure communication in Farm Configuration. To reconfigure, see the steps mentioned in above section *To define the Security Settings*.

Securing Communication between Services

To have a secure communication between the services, apply the below changes to the Windows registry and reboot the server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
```

For more information on securing communication, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client and

https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2015/2960358.

Define Services and Global Settings

You can define the Services and Global Settings using the Farm Configuration Wizard or through Configuration XMLs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To define Services

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.

   Note: You can also customize the settings for these services and websites using the **Custom Settings** option. For more information about Service-related changes and Website-related changes, see Advanced Settings and Administering Web Servers respectively.

2. Click **Next**.

   The Global Settings dialog box appears.

3. In the **Communication** tab, enter key fields in **Email Settings** and **Exception Email settings** to configure exception e-mails.

   The following are the details:

   - **Host**: Enter information related to mail server.
     Example: Enter details related to Microsoft Exchange Server (IP address of the exchange server).
   - **Port**: Enter the port number for the mail server. The default port number is 25.
   - **From email**: Enter the from e-mail address.
     Example: admin@telematics.com
   - **To email**: Enter the to e-mail address.
     Example: alex@telematics.com

4. Click **OK**.

Note: To confirm the exception e-mail settings, click **Test SMTP**. An e-mail will be generated and sent to the **To email** address confirming successful configuration of the e-mail. The advanced settings like **Secure Socket Layer (SSL), Email Timeout, User Name,** and **Password** for the e-mail account holder, **Password Encrypted** flag will differ based on the mail server settings.

Provision Services

While the installation is in progress, you can view the services that will be installed. This provides information on the progress of installation. At the end of the installation, a status indicating the completion of installation appears.

- Click **OK** to complete the installation.

Create a Repository DataSource

A **Repository DataSource** is used to store all repository related data. The **Repository DataSource** must be created on a machine where the **Central Configuration** site is installed.

Example

A TelematicsDataSource is created for TelematicsHR Repository.

A Repository contains Workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating the repository.

To create a Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Central Configuration web page appears.

2. Click **DataSource Settings**.

   The Datasource List page appears.

3. Click **Add**.

   The Manage Datasource page appears.

4. Enter the required details.

   a. **Name**: Enter a valid data source name. Example: *TelematicsDataSource*.

   b. **Description**: Enter a description for the data source. Example: *This data source is used to store connection details*.

   c. **Connection String**: Select this option to establish a connection with the database. The connection string contains fields that that defines the connection. The connection information is encrypted and stored.

   Server Name

   To use a database on the local machine, enter "localhost", ".", or the local computer's name for the **Server Name**.

   Note: Server Name can be machine name/IP address/SQL Server AlwaysON/SQL Server instance name. Instead of entering "local host", it is preferable to give the local machine name.

   If the **Database Type** is **SQL Server**, then specify an IPV6/IPV4 address for the **Server Name**.
   To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: *localhost,1455*
   To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be *<server name>\SQLExpress*.

   Authentication Type

   Use this field to determine the process to validate a user who is trying to connect to the database. AVEVA Work Tasks uses Windows and SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. The **User ID** and **Password** field is disabled if Windows Authentication is selected. Example: *Windows*

   Advanced Settings

   Use this field to define advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection.
   Example: *Connection Timeout = 20*
   or
   For defining maximum connections that can be opened in a connection pool.
   Example: *max pool size = 250*
   Default connection pool is set as 250. We recommend to enter a higher value on heavily loaded environment.
   While creating a data source for a repository, use the following custom settings to set properties for the connection pool.
   encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
   encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.
   enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

distributed is reduced to a great extent. With this feature, AVEVA Work Tasks controls the transaction from escalating to an expensive distributed transaction.
The above property can be used to turn off this feature. The default value is set as *True*.
connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is 30 seconds. The value should be set in seconds.
maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is infinite.
sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.
These settings are case sensitive and should be mentioned only in lower case.

Database:

Specify or select a database that can be used as a repository database. This database stores all repository related data. The repository database is created if it does not exist on the database server instance.
Example: *TelematicsDB*

Enable Table Partitioning:

Select to enable database table partitioning. For more information, see Partitioning Database.

d. **Connection Provider**: Use this to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.

The selected dll must be copied to all the machines, all servers and all clients, that are connected to the current Farm.

Assembly Path

Click to browse and choose the required dll file.
To use custom properties (enableltm, connectionretrywaittime, maxconnectionretries, and sqlcommandtimeout) in the Assembly connection string, the database must be created before.
See *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Developer Guide* for creating the assembly. Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.

   A message about successful test connection appears.

6. Click **OK**.

   A message about successful creation of the data source appears.The message also states that the required tables will be created in the data source after a repository is associated with the data source.

When a data source is saved, an entry is created for the data source in the **Farm Database**. When a first repository is created for the data source, AVEVA Work Tasks tables are added to the data source. While creating a data source, the archival database is created with the same database name suffixed by *_BI*. This database is used for archival in future use.

Note: After you configure the AVEVA Work Tasks, you must add the created **Farm Database** to the **SQL Server AlwaysOn Availability Groups**. Contact your Database Administrator to add the database to **SQL Server AlwaysOn Availability Groups**.

To view the data source created, click **DataSource Settings** link.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create a Repository

A **Repository** contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

To create a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The Central Configuration page appears.
2. Click **Repository Settings**.
   The Repository List page appears.
3. Click **Add**.

   Note: If the number of repositories exceeds the permitted limit for the license, then the message "You have exceeded the maximum number of repositories" appears, and the repository creation is stopped. Click **Close**.

   The **Manage Repository** page appears.
4. Enter the required details.
5. Click **OK**.
   The **Config Administrator** page appears.
6. Click to configure an administrator.
   The User Lookup dialog box appears.
7. Select the provider from the provider drop-down list.
8. Enter the details to search the administrator for the repository that is created.

   Note: You can also search a user based on *ObjectSID* and *objectGUID* (applicable only for Active Directory provider).
9. Click **Search**.
   The user is listed under **Available Resources**.
10. Select and click the arrow to list the user under **Selected Resource**.
11. Click **Update**.
    The selected user is displayed in **User Lookup**.
12. Click **OK**.
    A message confirming the successful creation of an administrator appears.
13. Click **OK**.
    The **Repository List** page appears.
    For more information about configuring the system user, see Configure System User.

Manage Repository Window Fields

The following table explains the fields of the Manage Repository window:

Field Name	Description	Example
Repository DataSource	Lists all the data sources created in	TelematicsDatasource

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	a Farm. The selected data source will be used to store data for the current repository.	
Attach DataSource	Select Attach DataSource checkbox and then select a repository from the selected DataSource .Data associated with the selected repository will be copied to the new repository. The version of the installed product and destination DataSource should be the same as that of the DataSource from which the repository data is to be copied. For example, in an upgraded environment, from build Work Tasks version 1 to Work Tasks version 2, you cannot attach a DataSource created in build Work Tasks version 1 to build Work Tasks version 2. To match the build version of the installed product to the version of the DataSource: 1. If the DataSource was created in build Work Tasks version 1, then the entire build Work Tasks version 1 must be upgraded to Work Tasks version 2 by running the product installer in the upgrade mode. 2. All the Farm, DataSources, Repositories, and Lists of build Work Tasks version 1 must be upgraded to build Work Tasks version 2 by running the AVEVAWorkTasksUpgradeManager.exe located at the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder. 3. After upgrading all the AVEVA Work Tasks components to	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	build Work Tasks version 2 build, that DataSource can be used in any machine that has same version by using this feature.	
Repository Name	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the repository name.	Telematics Repository
Description	Provide a description for the repository.	This stores Telematics Employee Leave Details.
From mail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might occur in the Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be sent to the mail ID defined in **Exception to mail**. The From address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To avoid this, restart the Workflow Engine Service and Communication Service.	Provide the valid email id that is specified in the **User Name** field while configuring **SMTP Settings** in the **Mail Server Settings** page. For more details, see Mail Server Setting.
Exception to mail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails is sent to the email ID specified in **Exception to mail**.	Provide the valid email id.
Alert initiator of the Workflow	Determines if the status of a workflow should be sent as a	A process with a rule to check the pending leave available for the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	notification mail to the initiator of the workflow.	employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
Servers	Lists the servers available in the Farm. Repository will be available only to the mapped servers.	Work Tasks Server Name
Enterprise Consoles	Lists all the Enterprise Consoles for a selected AVEVA Work Tasks Server. Repository will be available only to the mapped Enterprise Console.	http://worktaskservername:8001/EnterpriseConsole
Enable Theme	Select this checkbox to enable the defined theme.	-
Theme Name	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. It provides **Default** and **Lite** themes.	-
Resource Provider	Lists the various default resource providers supported by AVEVA Work Tasks. Resource providers are also termed as User Provider. A User Provider is a data store containing information about the user. AVEVA Work Tasks also supports custom resource provider. For more information about Resource Provider, see User Providers. AVEVA Connect and Azure Active Directory resource providers are also available after you configure them. To configure these providers, see AVEVA Connect Settings and Azure Active Directory Settings. AVEVA Work Tasks allows secured LDAP configuration. The default port number for LDAPS is 636. You	AVEVA Work Tasks Lists or Active Directory

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	can map the repository in IDE and use the Secured LDAP. The logged-in Active Directory is displayed by default, and **Active Directory Datasource** expects **LDAP://** or **Fully Qualified Domain Name** as value.	
Custom Application Folder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with a semicolon.	-
Hide Repository	Select this option to make the repository unavailable to the Farm.	-
Source path	Enter the path to the Source Elements Folder.	-
Attachment Storage Path	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an absolute path. - The default path is the **Attachments** folder in *C:\ProgramData\AVEVA Work Tasks*. - The default path is used when no path is specified. - Ensure appropriate rights are set to the folder, where attachments are stored, for the following identities: -- The application pool identity for **Central Configuration** and **Enterprise Console(s)**. -- The account configured for the **Workflow Engine** service. - In case of mapped drives, ensure they are available to the identities. - In multiple-server environments, the path specified must be in a shared location.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Auto Log Off Settings for Enterprise Console	Automatically logs out the logged in user after the designated period of inactivity. Enter the time in minutes. This field accepts only valid integer values between 0 to 360. 0 minute means the user will not get automatically logged off. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. The default value for automatic log off is 30 minutes. You use this field to define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML.	0 to 360 (minutes)
Auto Log Off Settings for AVEVA Work Tasks Pro	Request for user credentials for AVEVA Work Tasks Pro app after the set time is elapsed. Enter the time in minutes. This field accepts only valid integer values between 0 to 2880. 0 minute means the user will not get automatically logged off. The default value is 30 minutes. If AVEVA Work Tasks Pro app is offline, the Auto Log Off will not get active, and the user will not be automatically logged off as per the defined settings.	0 to 2880 (in minutes)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Offline Support	Select this option to enable offline submission of items in AVEVA Work Tasks Pro app.	-
Refresh on Work Item submission	Select this option to enable Inbox refresh on Work Item submission in AVEVA Work Tasks Pro . Refresh option is enabled by default.	-
Timeout	Enter the time in seconds to set the timeout. If the value is set as 0, then there is no timeout. Default value is 60 seconds.	-
Label for Fill Form	Enter the label for Fill Form. The customized label will be displayed for the Fill Form in AVEVA Work Tasks Pro.	Work Tasks
Date Configuration Flag for Alternate Actors	Select the **Enable Date Settings** option to set the **From** date and **To** date as mandatory while creating alternate actors. For more information about creating alternate actors, see *Alternate Actors* topic in the *User Guide*.	-
Display Domain name with User Name	Select this checkbox to display the domain name along with the user name in the Enterprise Console and the List. The domain name will be displayed only: • If this checkbox is selected. • If the **Resource Provider** is selected as **Active Directory**.	-

User Providers

AVEVA Work Tasks supports the following User Providers by default:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Active Directory

When you select **Active Directory** as the **Resource Provider**, details such as **Active Directory datasource** and the **Active Directory search base** are populated by default.

Authentication User, **Authentication Password**, **User Log-on ID**, and **User Log-on Display Name** are optional fields. Access to the active directory might not be available for all users. In such cases, provide the active directory authentication user name and the active directory authentication password with which you can connect to the Active Directory.

AVEVA Work Tasks considers *SAM-Account-Name* as **User Log-on ID** by default. If you want to use active directory attributes other than the *SAM-Account-Name* user, then enter the desired active directory attribute details in the **User Log-on ID** field and **User Log-on Display Name** field. **User Log-on Display Name** is the user friendly display name of the active directory attribute. This **User Log-on Display Name** is referred in the User Lookup dialog box against the **User Log-on ID**.

Steps

1. When you select **Active Directory** as the **Resource Provider**, then the **Use Token Service** field appears. In this field, select one of the following:
   - If you do not want to use AIM authentication, select **Without Token Service**.
   - If you want to use AIM authentication, then select a server from the list of alias server names.
2. Enter **Active Directory datasource**.
3. Enter **Active Directory search base**.
4. Enter **Domain Name**.

   The domain name is case-sensitive.
5. Enter **Authentication User**.
6. Enter **Authentication Password**.
7. Check **Is Cross Domain Role Provider** checkbox, if cross domain users and roles mapping is required.

   Example: Users *John* and *Jim* of domain *TelematicsU* and users *Mike* and *Bob* from domain *TelematicsAU* are added to role Team HR of domain *TelematicsMain*.
8. Enter **User Log-on ID** if you want to use another Active Directory attributes instead of *SAM-Account-Name*.

   We recommend to use attributes of type *string* as input for **User Log-on ID.** However, *ObjectSID* and *objectGUID* are also supported.

   After you set the **User Log-on ID**, you cannot change it later.
9. Enter the **User Log-on Display Name**.

   After you set the **User Log-on Display Name**, you cannot change it later.

   This is the default display name of the user account of the active directory user used. **User Log-on Display Name** is used in user lookups.
10. Click **Test LDAP**.

    A message about successful LDAP test appears.

Note:
- Some of the active directory attributes or properties are not indexed. If you use an active directory attribute that is not indexed, it might impact the log-on validation performance based on the active directory entries. We recommend to use indexed attributes as **User Log-on ID** or index the attributes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- For an upgraded repository, you cannot change both **User Log-on ID** and **User Log-on Display Name**.
- You cannot localize **User Log-on ID** and **User log-on Display Name** fields.
- Disabled users will not be shown in user lookup search.

Using Forms

For Forms Resource Provider:

- If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.

1. In the **Resource Provider** drop-down list**,** select **Forms.**
2. In the **User Entity properties** field, enter XML data.

   Click **Sample Code** to get the XML Schema Template.

   Example

   ```
   <properties>
   <property name="userid" type="string" purpose="identifier" />
   <property name="username" type="string" purpose="name" />
   <property name="pwd" type="string" purpose="password" />
   </properties>
   ```
3. In the **User Provider Properties** field, enter the same XML Schema.

   There must be at least one identifier in the XML Schema. For example, UserID
4. In the **Datasource** field, enter a valid data source or specify a dll.

   The value for the data source can be specified as **Physical path of the dll** or **Physical path of the dll::param. Param** is not mandatory.

   See topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Build Business Process* section for creating the assembly.

   Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly, whereas one class can return multiple connection strings.

   Examples of data source parameter:

   - c:\DLLStore\GetConnectionString.dll
   - c:\GetConnectionString.dll::Leave

   If there are two data sources, *Leave* and *Licensing*, and only one DLL is used to return a connection string based on the data source selected. The parameter can be specified as **c:\GetConnectionString.dll::Leave** or **c:\ GetConnectionString.dll::Licensing**. The connection string can be returned from the connection dll based on this parameter.
5. Enter a proper table name or view name which holds the User Name and Password for the Authentication purpose.

   Example: Passmast (the table which is available in IMS_NEW Database)
6. Click **Test SQL Connection**.

   A message about successful test connection appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using ASP.NET Membership

This resource provider lists users defined in the asp.net membership database.

Lists

This resource provider lists users created using AVEVA Work Tasks Users Creation. If you are unable to determine a User Provider to create a repository, use AVEVA Work Tasks Lists.

To verify successful creation of the Repository:

1. Click **Enterprise Console** URL from the site path column.
   The repository created is listed.
   Add the created Enterprise Console link into "Add to Favorites" to easily navigate in future.
2. Click **Repository User Provider** link to logon to the repository.
   The Enterprise Console log-on page appears.
3. Enter the logon details, and then click **Login**.
   The Enterprise Console home page appears. This indicates successful creation of a Repository.
   All features can only be available for a user with administrative privileges.

Note: Consider you created the Repository and then deleted it. If you create the same Repository again immediately and try to log into it through Enterprise Console, the Provider does not list the Repository. You need to access IIS and recycle AVEVA Work Tasks from the Application Pool to get the Repository listed in the Provider list..

Verify the Installation

After you install AVEVA Work Tasks, verify that the product is installed successfully. To verify successful installation, create a simple Workflow using Process Designer. The tasks includes the following:

- Creating a Workflow
- Publishing the Workflow
- Testing the Workflow

Creating a Workflow

1. Log on to the **Enterprise Console**.
2. If the Repository is configured for Active Directory provider, go to step 4.
3. Enter the **User Name** and **Password**.
   The **User Name** and **Password** will be validated with the user provider selected while creating the repository.
   The default authentication provider is the Repository User Provider List.
4. In the **Enterprise Console**, click menu icon.
   All features can only be available for a user with administrative privileges.
5. From the menu, select **Workflows**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Workflow window appears.

6. On the Action bar, click **New**, and then click **Workflow**.
   The New Workflow window appears.
7. Enter **Title** and **Description**.
8. Click **Save & Continue**.
   The successful message appears.
9. Select **Design the Workflow now**.
   This is the default option.
10. Click **Finish**.
    The Process Designer window appears.
11. From the activities pane on the left, drag the Information activity to the Process Designer window.
12. The activity *Information1* is created.
    Click to create a link between two activities.
    The Link Action window appears.
    a. Select from the **From Action** drop-down box.
    b. Select from the **To Action** drop-down box.
    c. Click **OK**.
       The link is established.
13. Right-click *Information1*.
14. Select **Activity Properties** from the context menu.
    The Activity Properties pane appears.
    a. Enter **Name** and **Description**.
    b. Click to **Assign Actor(s)**.
       The Assign Actor(s) window appears.
    c. Select the required provider from the **Providers** drop-down list.
    d. Select values from the drop-down list.
    e. Click **Add**.
    f. Click **Update**.
15. Click to enter the **Subject** for Notification Contents.
    The window to enter subject for notification appears.
16. Click **Update**.
17. Click to enter the **Body** for Notification Contents.
    The window to enter the Body content appears.
18. Click **Update**.
19. From the Activity Properties pane, click **Save**.

Publishing the Workflow

1. On the **Publish** menu, click **Publish**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Published successful message appears.

2. Close the dialog box.

   The Enterprise Console screen lists the workflow created.

Testing the Workflow

1. Select the listed workflow.
2. Right-click the selected workflow.
3. Click **Publish and Test Run**.

   The **Status Finished successfully** appears.

   If Status is **Execution Pending** in the Process Execution window, then click on the **Refresh** button to get the status.
4. In the **Enterprise Console** menu, select **Inbox**.

   The Inbox lists the work item.
5. Click to view the work item listed.

   This indicates that the installation is successful.

Web Server, AVEVA Work Tasks Server, and Database Server on Different Machines

This section describes the activities involved while installing the Single Server for the scenario Web Server, AVEVA Work Tasks Server, and Database Server on Different Machines.

In this scenario, Server is installed on Machine 1, Database is installed on Machine 2 and Client is installed on Machine 3.

Ensure that you install the product in the following sequence:

- High-Level View of Server Installation
- High-Level View of Client Installation

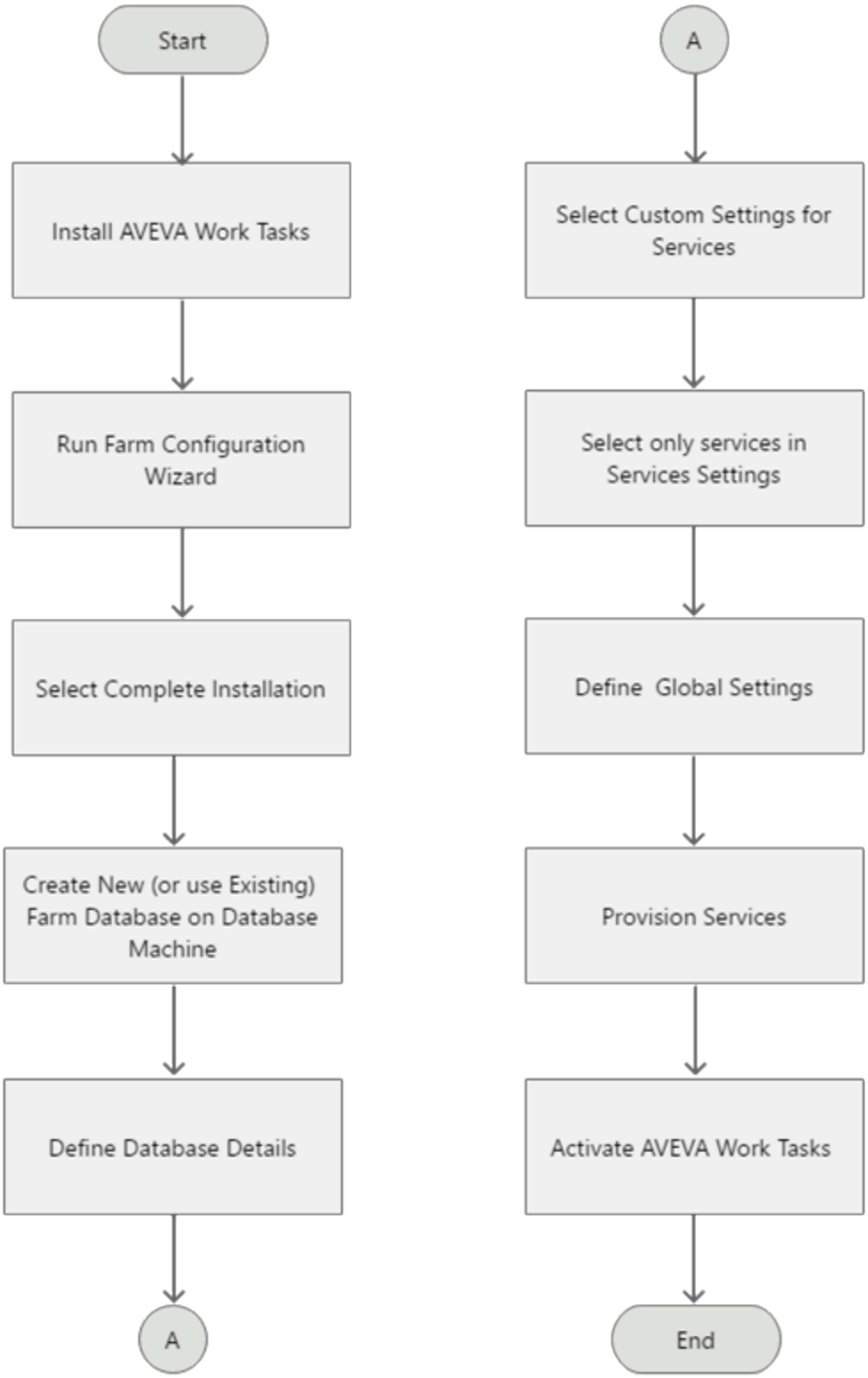
See Also: AVEVA Work Tasks Web Server, AVEVA Work Tasks Server, and AVEVA Work Tasks Database Server on Different Machines

High-Level View of Server Installation

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

The following flowchart illustrates a high-level view of the various installation steps with an overview of the various activities involved while installing AVEVA Work Tasks. Using this flowchart you can visually keep track of your activities while following the sequence of tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



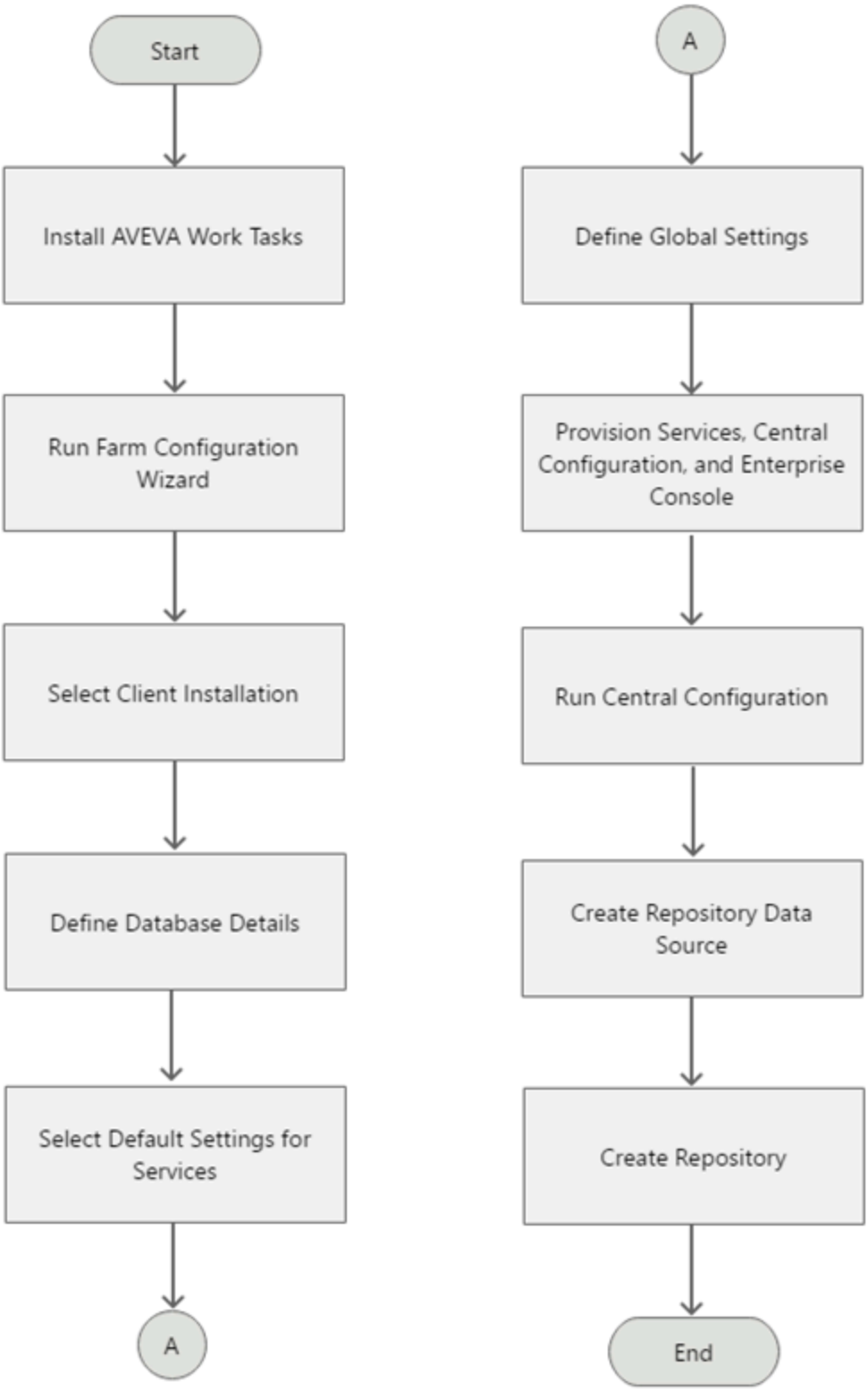
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

High-Level View of Client Installation

This section provides a high-level view of AVEVA Work Tasks Client installation. The Client components are installed on Web Servers. For more information on this scenario, see Single Server.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation

This section provides installation steps for the scenario Web Server, AVEVA Work Tasks Server, and Database on Different Machines.

The installation includes the following:

- Installing AVEVA Work Tasks Server
- Installing AVEVA Work Tasks Client

AVEVA Work Tasks Server has to be installed before installing the Client. In this scenario, the Client has to be installed on a single machine.

AVEVA Work Tasks Server

This section describes the activities involved while installing AVEVA Work Tasks on a server machine.

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
4. Click **Next**.

   The License Server details are configured successfully.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.
4. Click **Next**.

   The **Edit Farm Settings** window appears.
5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.
6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
9. Click **Next**.

   The License Server details are configured successfully.

Select the type of installation

To select the type of installation:

1. In the Select the type of installation page, select **Complete Installation (Server)**.

   A Complete Standalone Deployment does not require a client installation.
2. Under **Create AVEVA Work Tasks Farm**:
   - Click **Create AVEVA Work Tasks Farm database** to create a new Farm database.
   - Click **Use existing AVEVA Work Tasks Farm database (Creates AVEVA Work Tasks Farm tables only if they do not exist)** to use an existing database as the Farm database.
3. Click **Next**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create Farm Database

Use the Create AVEVA Work Tasks Farm Database page to provide **New Database** details. Select the **Connection String** option to establish a connection with the database. This connection information is encrypted and stored.

You can enter either Windows or Microsoft SQL Server user settings for authentication.

Note:
- The Farm database password does not support **&** and **;** special characters.
- While setting up **Farm Database** with **Windows Authentication**, do not enter **FQDN** in the **Domain Name** field.

To Create Farm Database

1. Enter **Database Title**.
2. Select **Connection String**.

   See *Connection Provider* later in this topic, if you have selected the **Connection Provider** option.

   - Enter the following details:

     Server Name:

     Provide information about the database server on which the Farm database can be installed.

     If the **Database Type** is **SQL Server**, then you can specify an IPv6 address for the **Server Name**.

     If SQLExpress Instance need to be used as a Database instance of the Farm database, then **Server Name** should be <server name>\SQLExpress.

     To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

   - **Authentication Type**

     This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports Windows and SQL Server authentication process.

     Example: Windows
     If Windows Authentication is selected, the user configured for the Farm configuration will be used to access the database. This is the authentication recommended by AVEVA Work Tasks.

   - **Login**

     This field is used to enter the SQL login credential for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Password**

     This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Advanced Settings**

     This is an optional field. It is used to enter any additional connection information, such as connection Timeout. Example: Connection Timeout = 20.

     While creating a data source for repository, use the following custom settings to set properties for connection pool. These settings are case sensitive and should be mentioned only in lower case.

     ```
     encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
     ```

     encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

certificate is installed on the SQL Server for SSL encryption.

enableltm: AVEVA Work Tasks internally implements light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive. This property can be used to turn off this feature. The default value is set as True.

connectionretrywaittime: While opening a new connection, if the data base server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

maxconnectionretries: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

3. Click **Test Connection** to ensure database connectivity.
4. Click **Next**.

   The message *AVEVA Work Tasks Farm is Created* appears.
5. Click **OK**.

AlwaysOn Availability Groups feature of Microsoft SQL Server

The **AlwaysOn Availability Groups** feature of Microsoft SQL Server is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in Microsoft SQL Server 2012, AlwaysOn Availability Groups maximizes the availability of a set of user databases for an enterprise.

After you configure the AVEVA Work Tasks, you must add the created Farm Database to the SQL Server AlwaysOn Availability Groups. Contact your Database Administrator to add the database to SQL Server AlwaysOn Availability Groups.

Notes:

- The authentication user needs different rights assigned for different scenarios. If a user wants to create database (Farm or Repository Data Source Database), Authentication User needs to have minimum 'dbcreator' access rights on Database Server Instance.

- If database is already created, user needs to have following permissions on Database Server:

- SELECT, INSERT, UPDATE, and DELETE permissions.

- EXECUTE permission for Stored Procedures.

To choose Connection Provider

Complete the following steps to choose a connection provider:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Enter **Database Title**.
2. In the **Database Type** box, select a database type.
3. Select **Connection Provider**.

   This option is used to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.
4. Enter the following details:
   - **Assembly Path**:

     Click the icon to browse and choose the required dll file.

     If custom properties (enableltm, connectionretrywaittime, maxconnectionretries and sqlcommandtimeout) are required to be used in Assembly connection string, then user should select **Use existing Farm database** while creating a the Farm. See Creating or Using an Existing Database.

     See the topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub chapter in the *Build Business Process* section for creating the assembly. Make sure that, there should be only one class implementing *IDBConnectionString* interface within this assembly.
5. Click **Test Connection** to ensure database connectivity.
6. Click **Next**.
7. The message *AVEVA Work Tasks Farm is created* appears.
8. Click **OK**.

Define Security Settings

You can define the Security Settings using the Farm Configuration Wizard.

For more information on Security settings, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client.

To define the Security Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Security Settings** and **Service Settings** and then, click **Next**.
4. Select **Secure Communication** to establish secured communication between the AVEVA Work Tasks services through HTTPS.
5. Select **Secure Web Sites** check box to configure secured web sites. Uncheck the **Secure Web Sites** check box to use HTTP sites. By default the sites are created in HTTPS mode.
6. Enter the Port number. The default port for the HTTPS site is 8001, and for the HTTP site is 8000. You can enter any port number that IIS accepts. But if any other services or applications are already using the entered port number, AVEVA Work Tasks will not start.
7. Select the available certificate from the **Certificate** list. If your certificate is not listed, click the **Refresh** icon to repopulate the list.

   Note: If the required certificate is not listed, generate one using the System Management Server

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

(Configurator tool). Do not use a self-signed certificate without a proper root chain.

8. Click the **Validate** button to validate the certificate before configuring the security settings.

   Note: An Internet connection is needed to validate the certificate.

9. Click **Configure**, then **Next**, and then **Finish** to apply the new security settings.

Note:
- To update the port number, certificate, and websites from unsecure to secure or vice-versa, select **Security Settings** and **Service Settings** check box in **Edit AVEVA Work Task Farm Settings** window before updating the values. After the required details are provided, these settings get updated only when you click **Configure**, then **Next**, and then **Finish** as you navigate through the Farm Configuration Wizard..
- If the Quickflow is already deployed on a machine, you need to deploy it again after you update the security settings. If secured web site is configured, then you cannot change the server configuration for Quickflow in Central Configuration. However, if the **Secure Web Sites** checkbox is not selected, you can make the changes to the server configuration.
- Advance Server communication cannot be configured on HTTP/HTTPS.
- For resources of Load Balanced Server, ensure to configure all services with the same protocol (HTTP or HTTPS).
- It is recommended to use Standard SSL certificate instead of self-signed SSL certificate.

Using ASB Certificate for Secured Communication

ASB Certificate is a PCS-managed SSL certificate and renews automatically every 30 days when the certificate is still valid. The certificate is valid for two years. The PCS watchdog service on the runtime node checks the validity of the SSL certificate within 5 minutes from when it starts and once a day. PCS also renews the SSL certificate if it is expired.

If the runtime node is offline and later back online, the PCS watchdog service regularly checks the certificate validity and renews it if it is required. Consider restarting the PCS watchdog service to enforce the SSL certificate validity check within 5 minutes from the restarting time.

Furthermore, when the SSL certificate is renewed, PCS binds it to port 443 (by default) and changes the SslThumbprint PCS registry value. If AVEVA Work Tasks sites are hosted on port 443, certificate binding will be auto-renewed when the SSL certificate is renewed. It is recommended to check the bindings in IIS with every certificate renewal to ensure the application is running seamlessly.

To disable the certificate renewal, consider disabling the **ArchestrA Certificate Renewal Service** windows service.

Note: Client Service will try to automatically bind the renewed certificate. If there are any issues in browsing the website or starting the services after certificate renewal, users need to reconfigure the secure communication in Farm Configuration. To reconfigure, see the steps mentioned in above section *To define the Security Settings*.

Securing Communication between Services

To have a secure communication between the services, apply the below changes to the Windows registry and reboot the server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
```

For more information on securing communication, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client and https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2015/2960358.

Define Services and Global Settings

You can define the Services and Global Settings using the Farm Configuration Wizard or through Configuration XMLs.

To define Services

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.

   Note: You can also customize the settings for these services and websites using the **Custom Settings** option. For more information about Service-related changes and Website-related changes, see Advanced Settings and Administering Web Servers respectively.

2. Click **Next**.

   The Global Settings dialog box appears.

3. In the **Communication** tab, enter key fields in **Email Settings** and **Exception Email settings** to configure exception e-mails.

   The following are the details:

   - **Host**: Enter information related to mail server.
     Example: Enter details related to Microsoft Exchange Server (IP address of the exchange server).
   - **Port**: Enter the port number for the mail server. The default port number is 25.
   - **From email**: Enter the from e-mail address.
     Example: admin@telematics.com
   - **To email**: Enter the to e-mail address.
     Example: alex@telematics.com

4. Click **OK**.

Note: To confirm the exception e-mail settings, click **Test SMTP**. An e-mail will be generated and sent to the **To email** address confirming successful configuration of the e-mail. The advanced settings like **Secure Socket Layer (SSL), Email Timeout, User Name,** and **Password** for the e-mail account holder, **Password Encrypted** flag will differ based on the mail server settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Define Central Configuration

1. Run **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Farm Configuration Wizard window appears.
2. Click **Next**.
3. The Edit Farm settings window appears.
4. Select **Service Settings**.
5. Click **Next**.

   The Services/Global Settings window appears.
6. Click against **Central Configuration** to edit AVEVA Work Tasks **Central Configuration settings**.

   The Central Configuration Settings page appears.
   - **http/https**: This option enables you to determine if you want a secured (https) or a non-secured site (http).
   - **Authentication Type**: This option allows you to provide the authentication type to access the Central Configuration. The available authentication types are Windows, AVEVA Connect, and Azure Active Directory.

     Note: The AVEVA Connect and Azure Active Directory authentication types are available only if these user providers are configured in the system.
     To configure AVEVA Connect user provider, see AVEVA Connect Settings.
     To configure Azure Active Directory user provider see Azure Active Directory Settings.
   - **Create at**: This field allows you to provide the Internet Information Service (IIS) path where the new site is created.
   - **Port Number**: This field is used to provide a port number for the Central Configuration.
7. Click against **Create at** to change the path of the Central Configuration.

   A message indicating that if the path has been changed the Central Configuration will no longer be accessible appears.
8. Click **Yes**.

   The IIS structure screen is displayed. Select the desired IIS path for the Central Configuration.
9. Click **OK**.

   The selected URL appears in the **Central Configuration Settings** screen.
10. Click **OK**.

Note:

- The default IIS path for Central Configuration is displayed. Click the edit icon to change the site path.

- If you want to update the authentication type to AVEVA Connect or Azure Active Directory of the already configured Central Configuration, you need to reconfigure the AVEVA Connect and Azure Active Directory user providers for the Central Configuration. For information on configuring user providers, see AVEVA Connect Settings and Azure Active Directory Settings.

- Central Configuration cannot be accessed in Chrome browser when configured with unsecured access (http://)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

for Azure Active Directory and AVEVA Connect authentication types.

Provision Services

While the installation is in progress, you can view the services that will be installed. This provides information on the progress of installation. At the end of the installation, a status indicating the completion of installation appears.

- Click **OK** to complete the installation.

AVEVA Work Tasks Client

This section describes the activities involved while installing AVEVA Work Tasks on a client machine.

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.
   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.
   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.
   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.
   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.
   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.

2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.

3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.

5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.

6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.

4. Click **Next**.

   The License Server details are configured successfully.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.
4. Click **Next**.

   The **Edit Farm Settings** window appears.
5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.
6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
9. Click **Next**.

   The License Server details are configured successfully.

Select Client Installation

1. In the Select the type of installation page, select **Client**.
2. Click **Next**.

   The Database User Settings page appears.

Connect to an Existing Farm Database

You can use the Connect to existing database dialog box to connect to the existing Farm. You can select the Connection String or Connection provider for the new database.

1. Select the computer name on which the Farm Database is installed from the **Server Name** field.
2. Select **Connection String**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Enter the following details:
   a. **Authentication Type**: This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports both Windows and SQL server authentication process. Example: Windows.
   b. **User ID**: This field is used to enter the SQL user ID for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   c. **Password**: This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   d. **Advanced Settings**: This is an optional field. It is used to enter any additional connection information such as connection Timeout etc. Example: Connection Timeout = 20.

      While creating a data source for repository, use the following custom settings to set properties for the connection pool.

      encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;

      encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

      enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks controls the transaction from escalating to a distributed transaction which is very expensive.

      The above property can be used to turn off this feature. The default value is set as True.

      connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

      maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

      sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

      Note: These settings are case sensitive and should be mentioned only in lower case.
   e. **Database**: This field is used to select the machine with database installed. Example: FarmDB.

      AVEVA Work Tasks will validate if at least one server is installed.
4. Click **Test Connection** to ensure database connectivity.

   You must have the following permissions on the Database Server:

   SELECT, INSERT, UPDATE, and DELETE permissions.

   EXECUTE permission for Stored Procedures.
5. Click **Next** to continue with the wizard.
6. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Define and Provision the Enterprise Console

You can define the required AVEVA Work Tasks services.

Steps to Define Enterprise Console

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.
2. Click **Next**.

   The **Installing Services** screen appears, indicating the progress of installation.

   A message *Installation is Complete* appears after all the services are installed.
3. Click **OK**.

Create a Repository DataSource

A **Repository DataSource** is used to store all repository related data. The **Repository DataSource** must be created on a machine where the **Central Configuration** site is installed.

Example

A TelematicsDataSource is created for TelematicsHR Repository.

A Repository contains Workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating the repository.

To create a Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**.

   The Central Configuration web page appears.
2. Click **DataSource Settings**.

   The Datasource List page appears.
3. Click **Add**.

   The Manage Datasource page appears.
4. Enter the required details.
   a. **Name**: Enter a valid data source name. Example: *TelematicsDataSource*.
   b. **Description**: Enter a description for the data source. Example: *This data source is used to store connection details*.
   c. **Connection String**: Select this option to establish a connection with the database. The connection string contains fields that that defines the connection. The connection information is encrypted and stored.

      Server Name

      To use a database on the local machine, enter "localhost", ".", or the local computer's name for the **Server Name**.

      Note: Server Name can be machine name/IP address/SQL Server AlwaysON/SQL Server instance name. Instead of entering "local host", it is preferable to give the local machine name.

      If the **Database Type** is **SQL Server**, then specify an IPV6/IPV4 address for the **Server Name**.
      To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: *localhost,1455*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be *<server name>\SQLExpress*.

Authentication Type

Use this field to determine the process to validate a user who is trying to connect to the database. AVEVA Work Tasks uses Windows and SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. The **User ID** and **Password** field is disabled if Windows Authentication is selected. Example: *Windows*

Advanced Settings

Use this field to define advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection.
Example: *Connection Timeout = 20*
or
For defining maximum connections that can be opened in a connection pool.
Example: *max pool size = 250*
Default connection pool is set as 250. We recommend to enter a higher value on heavily loaded environment.
While creating a data source for a repository, use the following custom settings to set properties for the connection pool.
encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.
enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature, AVEVA Work Tasks controls the transaction from escalating to an expensive distributed transaction.
The above property can be used to turn off this feature. The default value is set as *True*.
connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is 30 seconds. The value should be set in seconds.
maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is infinite.
sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.
These settings are case sensitive and should be mentioned only in lower case.

Database:

Specify or select a database that can be used as a repository database. This database stores all repository related data. The repository database is created if it does not exist on the database server instance.
Example: *TelematicsDB*

Enable Table Partitioning:

Select to enable database table partitioning. For more information, see Partitioning Database.

d. **Connection Provider**: Use this to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The selected dll must be copied to all the machines, all servers and all clients, that are connected to the current Farm.

Assembly Path

Click to browse and choose the required dll file.
To use custom properties (enableltm, connectionretrywaittime, maxconnectionretries, and sqlcommandtimeout) in the Assembly connection string, the database must be created before.
See *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Developer Guide* for creating the assembly. Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.

   A message about successful test connection appears.

6. Click **OK**.

   A message about successful creation of the data source appears.The message also states that the required tables will be created in the data source after a repository is associated with the data source.

When a data source is saved, an entry is created for the data source in the **Farm Database**. When a first repository is created for the data source, AVEVA Work Tasks tables are added to the data source. While creating a data source, the archival database is created with the same database name suffixed by *_BI*. This database is used for archival in future use.

Note: After you configure the AVEVA Work Tasks, you must add the created **Farm Database** to the **SQL Server AlwaysOn Availability Groups**. Contact your Database Administrator to add the database to **SQL Server AlwaysOn Availability Groups**.

To view the data source created, click **DataSource Settings** link.

Create a Repository

A **Repository** contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

To create a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The Central Configuration page appears.

2. Click **Repository Settings**.

   The Repository List page appears.

3. Click **Add**.

   Note: If the number of repositories exceeds the permitted limit for the license, then the message "You have exceeded the maximum number of repositories" appears, and the repository creation is stopped. Click **Close**.

   The **Manage Repository** page appears.

4. Enter the required details.
5. Click **OK**.

   The **Config Administrator** page appears.

6. Click to configure an administrator.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The User Lookup dialog box appears.

7. Select the provider from the provider drop-down list.
8. Enter the details to search the administrator for the repository that is created.

   Note: You can also search a user based on *ObjectSID* and *objectGUID* (applicable only for Active Directory provider).

9. Click **Search**.

   The user is listed under **Available Resources**.

10. Select and click the arrow to list the user under **Selected Resource**.
11. Click **Update**.

    The selected user is displayed in **User Lookup**.

12. Click **OK**.

    A message confirming the successful creation of an administrator appears.

13. Click **OK**.

    The **Repository List** page appears.

    For more information about configuring the system user, see Configure System User.

Manage Repository Window Fields

The following table explains the fields of the Manage Repository window:

Field Name	Description	Example
Repository DataSource	Lists all the data sources created in a Farm. The selected data source will be used to store data for the current repository.	TelematicsDatasource
Attach DataSource	Select Attach DataSource checkbox and then select a repository from the selected DataSource .Data associated with the selected repository will be copied to the new repository. The version of the installed product and destination DataSource should be the same as that of the DataSource from which the repository data is to be copied. For example, in an upgraded environment, from build Work Tasks version 1 to Work Tasks version 2, you cannot attach a DataSource created in build Work Tasks version 1 to build Work Tasks version 2.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	To match the build version of the installed product to the version of the DataSource: 1. If the DataSource was created in build Work Tasks version 1, then the entire build Work Tasks version 1 must be upgraded to Work Tasks version 2 by running the product installer in the upgrade mode. 2. All the Farm, DataSources, Repositories, and Lists of build Work Tasks version 1 must be upgraded to build Work Tasks version 2 by running the AVEVAWorkTasksUpgradeManager.exe located at the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder. 3. After upgrading all the AVEVA Work Tasks components to build Work Tasks version 2 build, that DataSource can be used in any machine that has same version by using this feature.	
Repository Name	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the repository name.	Telematics Repository
Description	Provide a description for the repository.	This stores Telematics Employee Leave Details.
From mail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might	Provide the valid email id that is specified in the **User Name** field while configuring **SMTP Settings** in the **Mail Server Settings** page. For more details, see Mail Server Setting.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	occur in the Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be sent to the mail ID defined in **Exception to mail**. The From address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To avoid this, restart the Workflow Engine Service and Communication Service.	
Exception to mail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails is sent to the email ID specified in **Exception to mail**.	Provide the valid email id.
Alert initiator of the Workflow	Determines if the status of a workflow should be sent as a notification mail to the initiator of the workflow.	A process with a rule to check the pending leave available for the employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
Servers	Lists the servers available in the Farm. Repository will be available only to the mapped servers.	Work Tasks Server Name
Enterprise Consoles	Lists all the Enterprise Consoles for a selected AVEVA Work Tasks Server. Repository will be available only to the mapped Enterprise Console.	http://worktaskservername:8001/EnterpriseConsole
Enable Theme	Select this checkbox to enable the defined theme.	-
Theme Name	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. It provides **Default** and **Lite** themes.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Resource Provider	Lists the various default resource providers supported by AVEVA Work Tasks. Resource providers are also termed as User Provider. A User Provider is a data store containing information about the user. AVEVA Work Tasks also supports custom resource provider. For more information about Resource Provider, see User Providers. AVEVA Connect and Azure Active Directory resource providers are also available after you configure them. To configure these providers, see AVEVA Connect Settings and Azure Active Directory Settings. AVEVA Work Tasks allows secured LDAP configuration. The default port number for LDAPS is 636. You can map the repository in IDE and use the Secured LDAP. The logged-in Active Directory is displayed by default, and **Active Directory Datasource** expects **LDAP://** or **Fully Qualified Domain Name** as value.	AVEVA Work Tasks Lists or Active Directory
Custom Application Folder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with a semicolon.	-
Hide Repository	Select this option to make the repository unavailable to the Farm.	-
Source path	Enter the path to the Source Elements Folder.	-
Attachment Storage Path	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	absolute path. - The default path is the **Attachments** folder in *C:\ProgramData\AVEVA Work Tasks*\. - The default path is used when no path is specified. - Ensure appropriate rights are set to the folder, where attachments are stored, for the following identities: -- The application pool identity for **Central Configuration** and **Enterprise Console(s)**. -- The account configured for the **Workflow Engine** service. - In case of mapped drives, ensure they are available to the identities. - In multiple-server environments, the path specified must be in a shared location.	

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Auto Log Off Settings for Enterprise Console	Automatically logs out the logged in user after the designated period of inactivity. Enter the time in minutes. This field accepts only valid integer values between 0 to 360. 0 minute means the user will not get automatically logged off. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. The default value for automatic log off is 30 minutes. You use this field to define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML.	0 to 360 (minutes)
Auto Log Off Settings for AVEVA Work Tasks Pro	Request for user credentials for AVEVA Work Tasks Pro app after the set time is elapsed. Enter the time in minutes. This field accepts only valid integer values between 0 to 2880. 0 minute means the user will not get automatically logged off. The default value is 30 minutes. If AVEVA Work Tasks Pro app is offline, the Auto Log Off will not get active, and the user will not be automatically logged off as per the defined settings.	0 to 2880 (in minutes)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Offline Support	Select this option to enable offline submission of items in AVEVA Work Tasks Pro app.	-
Refresh on Work Item submission	Select this option to enable Inbox refresh on Work Item submission in AVEVA Work Tasks Pro . Refresh option is enabled by default.	-
Timeout	Enter the time in seconds to set the timeout. If the value is set as 0, then there is no timeout. Default value is 60 seconds.	-
Label for Fill Form	Enter the label for Fill Form. The customized label will be displayed for the Fill Form in AVEVA Work Tasks Pro.	Work Tasks
Date Configuration Flag for Alternate Actors	Select the **Enable Date Settings** option to set the **From** date and **To** date as mandatory while creating alternate actors. For more information about creating alternate actors, see *Alternate Actors* topic in the *User Guide*.	-
Display Domain name with User Name	Select this checkbox to display the domain name along with the user name in the Enterprise Console and the List. The domain name will be displayed only: • If this checkbox is selected. • If the **Resource Provider** is selected as **Active Directory**.	-

User Providers

AVEVA Work Tasks supports the following User Providers by default:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Active Directory

When you select **Active Directory** as the **Resource Provider**, details such as **Active Directory datasource** and the **Active Directory search base** are populated by default.

Authentication User, **Authentication Password**, **User Log-on ID**, and **User Log-on Display Name** are optional fields. Access to the active directory might not be available for all users. In such cases, provide the active directory authentication user name and the active directory authentication password with which you can connect to the Active Directory.

AVEVA Work Tasks considers *SAM-Account-Name* as **User Log-on ID** by default. If you want to use active directory attributes other than the *SAM-Account-Name* user, then enter the desired active directory attribute details in the **User Log-on ID** field and **User Log-on Display Name** field. **User Log-on Display Name** is the user friendly display name of the active directory attribute. This **User Log-on Display Name** is referred in the User Lookup dialog box against the **User Log-on ID**.

Steps

1. When you select **Active Directory** as the **Resource Provider**, then the **Use Token Service** field appears. In this field, select one of the following:
   - If you do not want to use AIM authentication, select **Without Token Service**.
   - If you want to use AIM authentication, then select a server from the list of alias server names.
2. Enter **Active Directory datasource**.
3. Enter **Active Directory search base**.
4. Enter **Domain Name**.

   The domain name is case-sensitive.
5. Enter **Authentication User**.
6. Enter **Authentication Password**.
7. Check **Is Cross Domain Role Provider** checkbox, if cross domain users and roles mapping is required.

   Example: Users *John* and *Jim* of domain *TelematicsU* and users *Mike* and *Bob* from domain *TelematicsAU* are added to role Team HR of domain *TelematicsMain*.
8. Enter **User Log-on ID** if you want to use another Active Directory attributes instead of *SAM-Account-Name*.

   We recommend to use attributes of type *string* as input for **User Log-on ID.** However, *ObjectSID* and *objectGUID* are also supported.

   After you set the **User Log-on ID**, you cannot change it later.
9. Enter the **User Log-on Display Name**.

   After you set the **User Log-on Display Name**, you cannot change it later.

   This is the default display name of the user account of the active directory user used. **User Log-on Display Name** is used in user lookups.
10. Click **Test LDAP**.

    A message about successful LDAP test appears.

Note:
- Some of the active directory attributes or properties are not indexed. If you use an active directory attribute that is not indexed, it might impact the log-on validation performance based on the active directory entries. We recommend to use indexed attributes as **User Log-on ID** or index the attributes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- For an upgraded repository, you cannot change both **User Log-on ID** and **User Log-on Display Name**.
- You cannot localize **User Log-on ID** and **User log-on Display Name** fields.
- Disabled users will not be shown in user lookup search.

Using Forms

For Forms Resource Provider:

- If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.

1. In the **Resource Provider** drop-down list**,** select **Forms.**
2. In the **User Entity properties** field, enter XML data.

   Click **Sample Code** to get the XML Schema Template.

   Example

   ```
   <properties>
   <property name="userid" type="string" purpose="identifier" />
   <property name="username" type="string" purpose="name" />
   <property name="pwd" type="string" purpose="password" />
   </properties>
   ```
3. In the **User Provider Properties** field, enter the same XML Schema.

   There must be at least one identifier in the XML Schema. For example, UserID
4. In the **Datasource** field, enter a valid data source or specify a dll.

   The value for the data source can be specified as **Physical path of the dll** or **Physical path of the dll::param. Param** is not mandatory.

   See topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Build Business Process* section for creating the assembly.

   Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly, whereas one class can return multiple connection strings.

   Examples of data source parameter:

   - c:\DLLStore\GetConnectionString.dll
   - c:\GetConnectionString.dll::Leave

   If there are two data sources, *Leave* and *Licensing*, and only one DLL is used to return a connection string based on the data source selected. The parameter can be specified as **c:\GetConnectionString.dll::Leave** or **c:\ GetConnectionString.dll::Licensing**. The connection string can be returned from the connection dll based on this parameter.
5. Enter a proper table name or view name which holds the User Name and Password for the Authentication purpose.

   Example: Passmast (the table which is available in IMS_NEW Database)
6. Click **Test SQL Connection**.

   A message about successful test connection appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using ASP.NET Membership

This resource provider lists users defined in the asp.net membership database.

Lists

This resource provider lists users created using AVEVA Work Tasks Users Creation. If you are unable to determine a User Provider to create a repository, use AVEVA Work Tasks Lists.

To verify successful creation of the Repository:

1. Click **Enterprise Console** URL from the site path column.
   The repository created is listed.
   Add the created Enterprise Console link into "Add to Favorites" to easily navigate in future.
2. Click **Repository User Provider** link to logon to the repository.
   The Enterprise Console log-on page appears.
3. Enter the logon details, and then click **Login**.
   The Enterprise Console home page appears. This indicates successful creation of a Repository.
   All features can only be available for a user with administrative privileges.

Note: Consider you created the Repository and then deleted it. If you create the same Repository again immediately and try to log into it through Enterprise Console, the Provider does not list the Repository. You need to access IIS and recycle AVEVA Work Tasks from the Application Pool to get the Repository listed in the Provider list..

Verify the Installation

After you install AVEVA Work Tasks, verify that the product is installed successfully. To verify successful installation, create a simple Workflow using Process Designer. The tasks includes the following:

- Creating a Workflow
- Publishing the Workflow
- Testing the Workflow

Creating a Workflow

1. Log on to the **Enterprise Console**.
2. If the Repository is configured for Active Directory provider, go to step 4.
3. Enter the **User Name** and **Password**.
   The **User Name** and **Password** will be validated with the user provider selected while creating the repository.
   The default authentication provider is the Repository User Provider List.
4. In the **Enterprise Console**, click menu icon.
   All features can only be available for a user with administrative privileges.
5. From the menu, select **Workflows**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Workflow window appears.

6. On the Action bar, click **New**, and then click **Workflow**.
   The New Workflow window appears.
7. Enter **Title** and **Description**.
8. Click **Save & Continue**.
   The successful message appears.
9. Select **Design the Workflow now**.
   This is the default option.
10. Click **Finish**.
    The Process Designer window appears.
11. From the activities pane on the left, drag the Information activity to the Process Designer window.
12. The activity *Information1* is created.
    Click to create a link between two activities.
    The Link Action window appears.
    a. Select from the **From Action** drop-down box.
    b. Select from the **To Action** drop-down box.
    c. Click **OK**.
       The link is established.
13. Right-click *Information1*.
14. Select **Activity Properties** from the context menu.
    The Activity Properties pane appears.
    a. Enter **Name** and **Description**.
    b. Click to **Assign Actor(s)**.
       The Assign Actor(s) window appears.
    c. Select the required provider from the **Providers** drop-down list.
    d. Select values from the drop-down list.
    e. Click **Add**.
    f. Click **Update**.
15. Click to enter the **Subject** for Notification Contents.
    The window to enter subject for notification appears.
16. Click **Update**.
17. Click to enter the **Body** for Notification Contents.
    The window to enter the Body content appears.
18. Click **Update**.
19. From the Activity Properties pane, click **Save**.

Publishing the Workflow

1. On the **Publish** menu, click **Publish**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Published successful message appears.

2. Close the dialog box.

   The Enterprise Console screen lists the workflow created.

Testing the Workflow

1. Select the listed workflow.
2. Right-click the selected workflow.
3. Click **Publish and Test Run**.

   The **Status Finished successfully** appears.

   If Status is **Execution Pending** in the Process Execution window, then click on the **Refresh** button to get the status.
4. In the **Enterprise Console** menu, select **Inbox**.

   The Inbox lists the work item.
5. Click to view the work item listed.

   This indicates that the installation is successful.

Web Farm (Multiple Web servers), AVEVA Work Tasks Server, and Database on Different Machines

This section describes the activities involved while installing the Single Server for the scenario Web Farm (Multiple Web servers), AVEVA Work Tasks Server, and Database Server on different machines.

In this scenario, the BPM Server is installed on Machine 1 and the Client is installed on multiple servers namely Machine 3 (or Server 1), Server 2 and Server N in a Web Farm. "Server N" represents the number of machines in the Web Farm.

Ensure that you install the product in the following sequence:

- High-Level View of Server Installation
- High-Level View of Client Installation

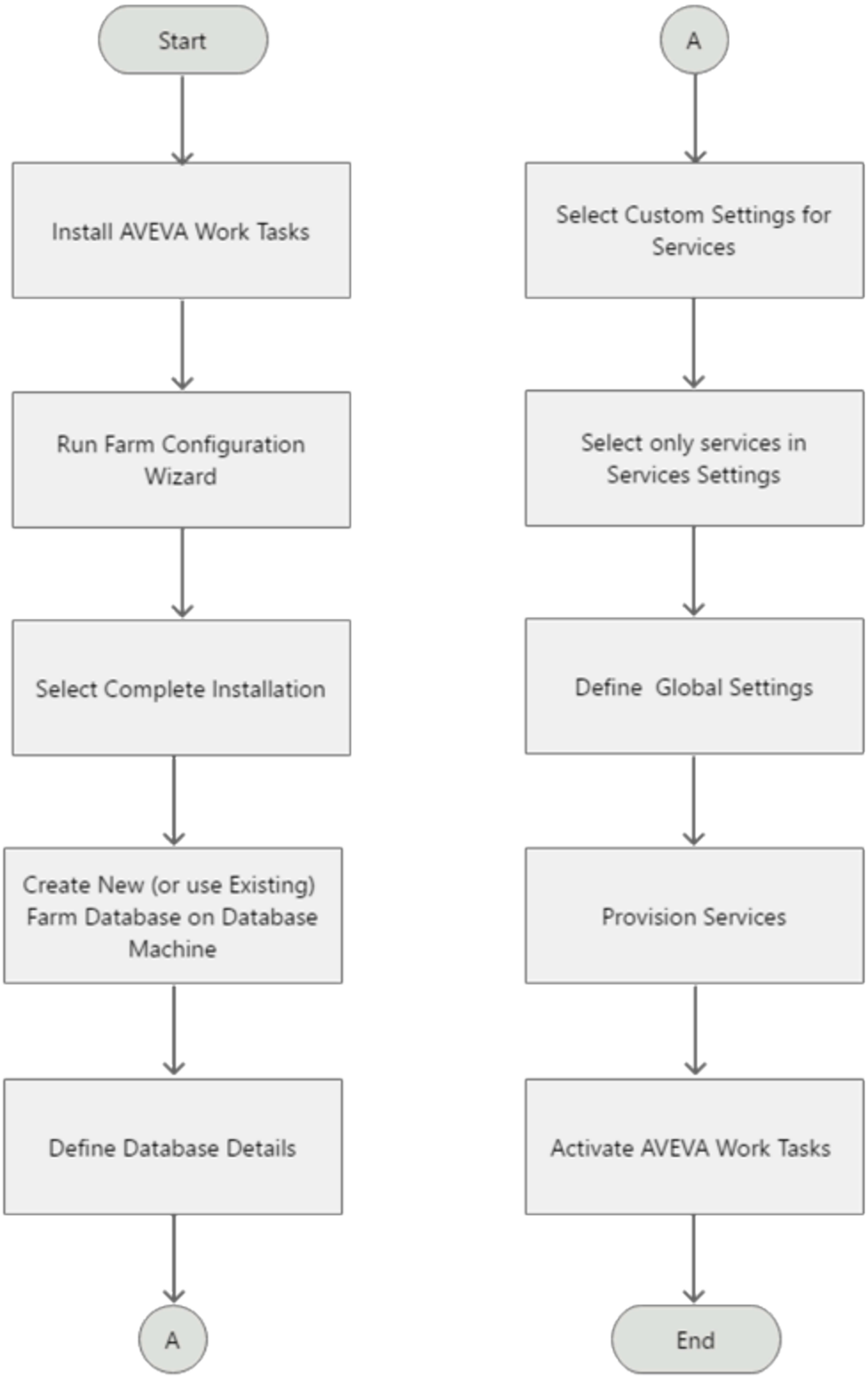
See Also: AVEVA Work Tasks Web Server Farm, Work Tasks Server, and AVEVA Work Tasks Database Server on Different Machines

High-Level View of Server Installation

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

The following flowchart illustrates a high-level view of the various installation steps with an overview of the various activities involved while installing AVEVA Work Tasks. Using this flowchart you can visually keep track of your activities while following the sequence of tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



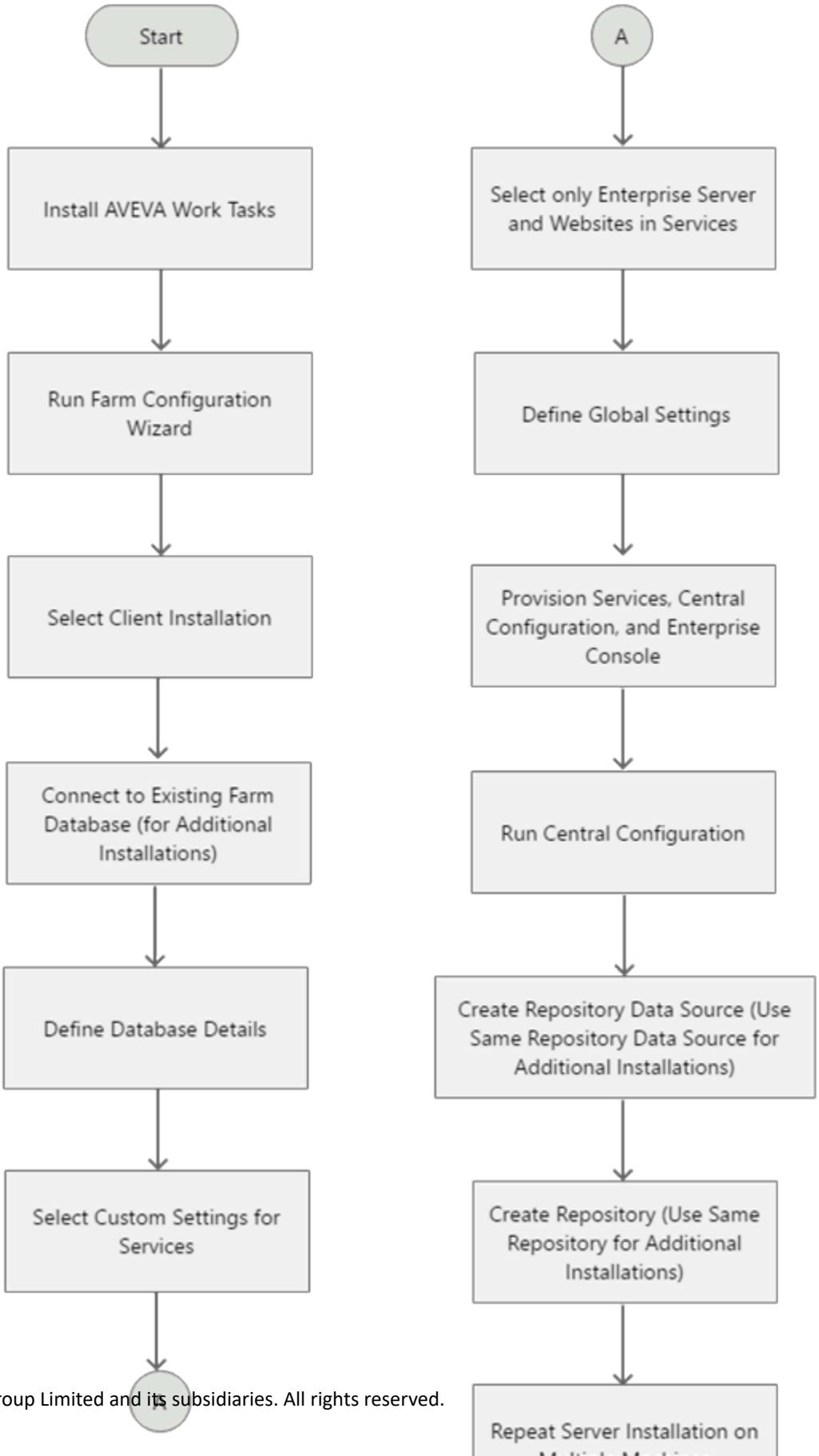
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

High-Level View of Client Installation

This section provides a high-level view of the Client installation. The Client components are installed on Web Servers. For more information on this scenario, see Single Server.

Before continuing with the installation, it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation

This section provides installation steps for the scenario Web Farm (Multiple Web servers), AVEVA Work Tasks Server, and Databases on different machines.

The installation includes the following:

- Installing Workflow Management Server
- Installing Workflow Management Client

AVEVA Work TasksServer has to be installed before installing AVEVA Work Tasks Client. In this scenario, the Client has to be installed on multiple machines.

AVEVA Work Tasks Server

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the current MSDTC settings.

2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.

3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.

5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.

6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.

4. Click **Next**.

   The License Server details are configured successfully.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.
4. Click **Next**.

   The **Edit Farm Settings** window appears.
5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.
6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
9. Click **Next**.

   The License Server details are configured successfully.

Select the type of installation

To select the type of installation:

1. In the Select the type of installation page, select **Complete Installation (Server)**.

   A Complete Standalone Deployment does not require a client installation.
2. Under **Create AVEVA Work Tasks Farm**:
   - Click **Create AVEVA Work Tasks Farm database** to create a new Farm database.
   - Click **Use existing AVEVA Work Tasks Farm database (Creates AVEVA Work Tasks Farm tables only if they do not exist)** to use an existing database as the Farm database.
3. Click **Next**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create Farm Database

Use the Create AVEVA Work Tasks Farm Database page to provide **New Database** details. Select the **Connection String** option to establish a connection with the database. This connection information is encrypted and stored.

You can enter either Windows or Microsoft SQL Server user settings for authentication.

Note:
- The Farm database password does not support **&** and **;** special characters.
- While setting up **Farm Database** with **Windows Authentication**, do not enter **FQDN** in the **Domain Name** field.

To Create Farm Database

1. Enter **Database Title**.
2. Select **Connection String**.

   See *Connection Provider* later in this topic, if you have selected the **Connection Provider** option.

   - Enter the following details:

     Server Name:

     Provide information about the database server on which the Farm database can be installed.

     If the **Database Type** is **SQL Server**, then you can specify an IPv6 address for the **Server Name**.

     If SQLExpress Instance need to be used as a Database instance of the Farm database, then **Server Name** should be <server name>\SQLExpress.

     To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

   - **Authentication Type**

     This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports Windows and SQL Server authentication process.

     Example: Windows
     If Windows Authentication is selected, the user configured for the Farm configuration will be used to access the database. This is the authentication recommended by AVEVA Work Tasks.

   - **Login**

     This field is used to enter the SQL login credential for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Password**

     This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Advanced Settings**

     This is an optional field. It is used to enter any additional connection information, such as connection Timeout. Example: Connection Timeout = 20.

     While creating a data source for repository, use the following custom settings to set properties for connection pool. These settings are case sensitive and should be mentioned only in lower case.

     ```
     encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
     ```

     encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

certificate is installed on the SQL Server for SSL encryption.

enableltm: AVEVA Work Tasks internally implements light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive. This property can be used to turn off this feature. The default value is set as True.

connectionretrywaittime: While opening a new connection, if the data base server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

maxconnectionretries: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

3. Click **Test Connection** to ensure database connectivity.
4. Click **Next**.

   The message *AVEVA Work Tasks Farm is Created* appears.
5. Click **OK**.

AlwaysOn Availability Groups feature of Microsoft SQL Server

The **AlwaysOn Availability Groups** feature of Microsoft SQL Server is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in Microsoft SQL Server 2012, AlwaysOn Availability Groups maximizes the availability of a set of user databases for an enterprise.

After you configure the AVEVA Work Tasks, you must add the created Farm Database to the SQL Server AlwaysOn Availability Groups. Contact your Database Administrator to add the database to SQL Server AlwaysOn Availability Groups.

Notes:

- The authentication user needs different rights assigned for different scenarios. If a user wants to create database (Farm or Repository Data Source Database), Authentication User needs to have minimum 'dbcreator' access rights on Database Server Instance.

- If database is already created, user needs to have following permissions on Database Server:

- SELECT, INSERT, UPDATE, and DELETE permissions.

- EXECUTE permission for Stored Procedures.

To choose Connection Provider

Complete the following steps to choose a connection provider:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Enter **Database Title**.
2. In the **Database Type** box, select a database type.
3. Select **Connection Provider**.

   This option is used to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.
4. Enter the following details:
   - **Assembly Path**:

     Click the icon to browse and choose the required dll file.

     If custom properties (enableltm, connectionretrywaittime, maxconnectionretries and sqlcommandtimeout) are required to be used in Assembly connection string, then user should select **Use existing Farm database** while creating a the Farm. See Creating or Using an Existing Database.

     See the topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub chapter in the *Build Business Process* section for creating the assembly. Make sure that, there should be only one class implementing *IDBConnectionString* interface within this assembly.
5. Click **Test Connection** to ensure database connectivity.
6. Click **Next**.
7. The message *AVEVA Work Tasks Farm is created* appears.
8. Click **OK**.

Define Security Settings

You can define the Security Settings using the Farm Configuration Wizard.

For more information on Security settings, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client.

To define the Security Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Security Settings** and **Service Settings** and then, click **Next**.
4. Select **Secure Communication** to establish secured communication between the AVEVA Work Tasks services through HTTPS.
5. Select **Secure Web Sites** check box to configure secured web sites. Uncheck the **Secure Web Sites** check box to use HTTP sites. By default the sites are created in HTTPS mode.
6. Enter the Port number. The default port for the HTTPS site is 8001, and for the HTTP site is 8000. You can enter any port number that IIS accepts. But if any other services or applications are already using the entered port number, AVEVA Work Tasks will not start.
7. Select the available certificate from the **Certificate** list. If your certificate is not listed, click the **Refresh** icon to repopulate the list.

   Note: If the required certificate is not listed, generate one using the System Management Server

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

(Configurator tool). Do not use a self-signed certificate without a proper root chain.

8. Click the **Validate** button to validate the certificate before configuring the security settings.

   Note: An Internet connection is needed to validate the certificate.

9. Click **Configure**, then **Next**, and then **Finish** to apply the new security settings.

Note:
- To update the port number, certificate, and websites from unsecure to secure or vice-versa, select **Security Settings** and **Service Settings** check box in **Edit AVEVA Work Task Farm Settings** window before updating the values. After the required details are provided, these settings get updated only when you click **Configure**, then **Next**, and then **Finish** as you navigate through the Farm Configuration Wizard..
- If the Quickflow is already deployed on a machine, you need to deploy it again after you update the security settings. If secured web site is configured, then you cannot change the server configuration for Quickflow in Central Configuration. However, if the **Secure Web Sites** checkbox is not selected, you can make the changes to the server configuration.
- Advance Server communication cannot be configured on HTTP/HTTPS.
- For resources of Load Balanced Server, ensure to configure all services with the same protocol (HTTP or HTTPS).
- It is recommended to use Standard SSL certificate instead of self-signed SSL certificate.

Using ASB Certificate for Secured Communication

ASB Certificate is a PCS-managed SSL certificate and renews automatically every 30 days when the certificate is still valid. The certificate is valid for two years. The PCS watchdog service on the runtime node checks the validity of the SSL certificate within 5 minutes from when it starts and once a day. PCS also renews the SSL certificate if it is expired.

If the runtime node is offline and later back online, the PCS watchdog service regularly checks the certificate validity and renews it if it is required. Consider restarting the PCS watchdog service to enforce the SSL certificate validity check within 5 minutes from the restarting time.

Furthermore, when the SSL certificate is renewed, PCS binds it to port 443 (by default) and changes the SslThumbprint PCS registry value. If AVEVA Work Tasks sites are hosted on port 443, certificate binding will be auto-renewed when the SSL certificate is renewed. It is recommended to check the bindings in IIS with every certificate renewal to ensure the application is running seamlessly.

To disable the certificate renewal, consider disabling the **ArchestrA Certificate Renewal Service** windows service.

Note: Client Service will try to automatically bind the renewed certificate. If there are any issues in browsing the website or starting the services after certificate renewal, users need to reconfigure the secure communication in Farm Configuration. To reconfigure, see the steps mentioned in above section *To define the Security Settings*.

Securing Communication between Services

To have a secure communication between the services, apply the below changes to the Windows registry and reboot the server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
```

For more information on securing communication, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client and https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2015/2960358.

Define Services and Global Settings

You can define the Services and Global Settings using the Farm Configuration Wizard or through Configuration XMLs.

To define Services

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.

   Note: You can also customize the settings for these services and websites using the **Custom Settings** option. For more information about Service-related changes and Website-related changes, see Advanced Settings and Administering Web Servers respectively.

2. Click **Next**.

   The Global Settings dialog box appears.

3. In the **Communication** tab, enter key fields in **Email Settings** and **Exception Email settings** to configure exception e-mails.

   The following are the details:

   - **Host**: Enter information related to mail server.
     Example: Enter details related to Microsoft Exchange Server (IP address of the exchange server).
   - **Port**: Enter the port number for the mail server. The default port number is 25.
   - **From email**: Enter the from e-mail address.
     Example: admin@telematics.com
   - **To email**: Enter the to e-mail address.
     Example: alex@telematics.com

4. Click **OK**.

Note: To confirm the exception e-mail settings, click **Test SMTP**. An e-mail will be generated and sent to the **To email** address confirming successful configuration of the e-mail. The advanced settings like **Secure Socket Layer (SSL), Email Timeout, User Name,** and **Password** for the e-mail account holder, **Password Encrypted** flag will differ based on the mail server settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Define Central Configuration

1. Run **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Farm Configuration Wizard window appears.
2. Click **Next**.
3. The Edit Farm settings window appears.
4. Select **Service Settings**.
5. Click **Next**.

   The Services/Global Settings window appears.
6. Click against **Central Configuration** to edit AVEVA Work Tasks **Central Configuration settings**.

   The Central Configuration Settings page appears.
   - **http/https**: This option enables you to determine if you want a secured (https) or a non-secured site (http).
   - **Authentication Type**: This option allows you to provide the authentication type to access the Central Configuration. The available authentication types are Windows, AVEVA Connect, and Azure Active Directory.

     Note: The AVEVA Connect and Azure Active Directory authentication types are available only if these user providers are configured in the system.
     To configure AVEVA Connect user provider, see AVEVA Connect Settings.
     To configure Azure Active Directory user provider see Azure Active Directory Settings.
   - **Create at**: This field allows you to provide the Internet Information Service (IIS) path where the new site is created.
   - **Port Number**: This field is used to provide a port number for the Central Configuration.
7. Click against **Create at** to change the path of the Central Configuration.

   A message indicating that if the path has been changed the Central Configuration will no longer be accessible appears.
8. Click **Yes**.

   The IIS structure screen is displayed. Select the desired IIS path for the Central Configuration.
9. Click **OK**.

   The selected URL appears in the **Central Configuration Settings** screen.
10. Click **OK**.

Note:

- The default IIS path for Central Configuration is displayed. Click the edit icon to change the site path.

- If you want to update the authentication type to AVEVA Connect or Azure Active Directory of the already configured Central Configuration, you need to reconfigure the AVEVA Connect and Azure Active Directory user providers for the Central Configuration. For information on configuring user providers, see AVEVA Connect Settings and Azure Active Directory Settings.

- Central Configuration cannot be accessed in Chrome browser when configured with unsecured access (http://)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

for Azure Active Directory and AVEVA Connect authentication types.

Provision Services

While the installation is in progress, you can view the services that will be installed. This provides information on the progress of installation. At the end of the installation, a status indicating the completion of installation appears.

- Click **OK** to complete the installation.

AVEVA Work Tasks Client

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
4. Click **Next**.

   The License Server details are configured successfully.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.
4. Click **Next**.

   The **Edit Farm Settings** window appears.
5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.
6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
9. Click **Next**.

   The License Server details are configured successfully.

Select Client Installation

1. In the Select the type of installation page, select **Client**.
2. Click **Next**.

   The Database User Settings page appears.

Connect to an Existing Farm Database

You can use the Connect to existing database dialog box to connect to the existing Farm. You can select the Connection String or Connection provider for the new database.

1. Select the computer name on which the Farm Database is installed from the **Server Name** field.
2. Select **Connection String**.
3. Enter the following details:
   a. **Authentication Type**: This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports both Windows and SQL server authentication process. Example: Windows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

b. **User ID**: This field is used to enter the SQL user ID for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

c. **Password**: This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

d. **Advanced Settings**: This is an optional field. It is used to enter any additional connection information such as connection Timeout etc. Example: Connection Timeout = 20.

   While creating a data source for repository, use the following custom settings to set properties for the connection pool.

   encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;

   encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

   enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks controls the transaction from escalating to a distributed transaction which is very expensive.

   The above property can be used to turn off this feature. The default value is set as True.

   connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

   maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

   sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

   Note: These settings are case sensitive and should be mentioned only in lower case.

e. **Database**: This field is used to select the machine with database installed. Example: FarmDB.

   AVEVA Work Tasks will validate if at least one server is installed.

4. Click **Test Connection** to ensure database connectivity.

   You must have the following permissions on the Database Server:

   SELECT, INSERT, UPDATE, and DELETE permissions.

   EXECUTE permission for Stored Procedures.

5. Click **Next** to continue with the wizard.
6. Click **OK**.

Define and Provision the Enterprise Console

You can define the required AVEVA Work Tasks services.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to Define Enterprise Console

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.
2. Click **Next**.

   The **Installing Services** screen appears, indicating the progress of installation.

   A message *Installation is Complete* appears after all the services are installed.
3. Click **OK**.

Create a Repository DataSource

A **Repository DataSource** is used to store all repository related data. The **Repository DataSource** must be created on a machine where the **Central Configuration** site is installed.

Example

A TelematicsDataSource is created for TelematicsHR Repository.

A Repository contains Workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating the repository.

To create a Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**.

   The Central Configuration web page appears.
2. Click **DataSource Settings**.

   The Datasource List page appears.
3. Click **Add**.

   The Manage Datasource page appears.
4. Enter the required details.
   a. **Name**: Enter a valid data source name. Example: *TelematicsDataSource*.
   b. **Description**: Enter a description for the data source. Example: *This data source is used to store connection details*.
   c. **Connection String**: Select this option to establish a connection with the database. The connection string contains fields that that defines the connection. The connection information is encrypted and stored.

      Server Name

      To use a database on the local machine, enter "localhost", ".", or the local computer's name for the **Server Name**.

      Note: Server Name can be machine name/IP address/SQL Server AlwaysON/SQL Server instance name. Instead of entering "local host", it is preferable to give the local machine name.

      If the **Database Type** is **SQL Server**, then specify an IPV6/IPV4 address for the **Server Name**.
      To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: *localhost,1455*
      To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be *<server name>\SQLExpress*.

      Authentication Type

      Use this field to determine the process to validate a user who is trying to connect to the database.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

AVEVA Work Tasks uses Windows and SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. The **User ID** and **Password** field is disabled if Windows Authentication is selected. Example: *Windows*

Advanced Settings

Use this field to define advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection.
Example: *Connection Timeout = 20*
or
For defining maximum connections that can be opened in a connection pool.
Example: *max pool size = 250*
Default connection pool is set as 250. We recommend to enter a higher value on heavily loaded environment.
While creating a data source for a repository, use the following custom settings to set properties for the connection pool.
encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.
enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature, AVEVA Work Tasks controls the transaction from escalating to an expensive distributed transaction.
The above property can be used to turn off this feature. The default value is set as *True*.
connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is 30 seconds. The value should be set in seconds.
maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is infinite.
sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.
These settings are case sensitive and should be mentioned only in lower case.

Database:

Specify or select a database that can be used as a repository database. This database stores all repository related data. The repository database is created if it does not exist on the database server instance.
Example: *TelematicsDB*

Enable Table Partitioning:

Select to enable database table partitioning. For more information, see Partitioning Database.

d. **Connection Provider**: Use this to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.

The selected dll must be copied to all the machines, all servers and all clients, that are connected to the current Farm.

Assembly Path

Click to browse and choose the required dll file.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To use custom properties (enableltm, connectionretrywaittime, maxconnectionretries, and sqlcommandtimeout) in the Assembly connection string, the database must be created before. See *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Developer Guide* for creating the assembly. Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.

   A message about successful test connection appears.

6. Click **OK**.

   A message about successful creation of the data source appears.The message also states that the required tables will be created in the data source after a repository is associated with the data source.

When a data source is saved, an entry is created for the data source in the **Farm Database**. When a first repository is created for the data source, AVEVA Work Tasks tables are added to the data source. While creating a data source, the archival database is created with the same database name suffixed by *_BI*. This database is used for archival in future use.

Note: After you configure the AVEVA Work Tasks, you must add the created **Farm Database** to the **SQL Server AlwaysOn Availability Groups**. Contact your Database Administrator to add the database to **SQL Server AlwaysOn Availability Groups**.

To view the data source created, click **DataSource Settings** link.

Create a Repository

A **Repository** contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

To create a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The Central Configuration page appears.

2. Click **Repository Settings**.

   The Repository List page appears.

3. Click **Add**.

   Note: If the number of repositories exceeds the permitted limit for the license, then the message "You have exceeded the maximum number of repositories" appears, and the repository creation is stopped. Click **Close**.

   The **Manage Repository** page appears.

4. Enter the required details.
5. Click **OK**.

   The **Config Administrator** page appears.

6. Click to configure an administrator.

   The User Lookup dialog box appears.

7. Select the provider from the provider drop-down list.
8. Enter the details to search the administrator for the repository that is created.

   Note: You can also search a user based on *ObjectSID* and *objectGUID* (applicable only for Active Directory

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

provider).

9. Click **Search**.

   The user is listed under **Available Resources**.
10. Select and click the arrow to list the user under **Selected Resource**.
11. Click **Update**.

    The selected user is displayed in **User Lookup**.
12. Click **OK**.

    A message confirming the successful creation of an administrator appears.
13. Click **OK**.

    The **Repository List** page appears.

    For more information about configuring the system user, see Configure System User.

Manage Repository Window Fields

The following table explains the fields of the Manage Repository window:

Field Name	Description	Example
Repository DataSource	Lists all the data sources created in a Farm. The selected data source will be used to store data for the current repository.	TelematicsDatasource
Attach DataSource	Select Attach DataSource checkbox and then select a repository from the selected DataSource .Data associated with the selected repository will be copied to the new repository. The version of the installed product and destination DataSource should be the same as that of the DataSource from which the repository data is to be copied. For example, in an upgraded environment, from build Work Tasks version 1 to Work Tasks version 2, you cannot attach a DataSource created in build Work Tasks version 1 to build Work Tasks version 2. To match the build version of the installed product to the version of the DataSource:	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	1. If the DataSource was created in build Work Tasks version 1, then the entire build Work Tasks version 1 must be upgraded to Work Tasks version 2 by running the product installer in the upgrade mode. 2. All the Farm, DataSources, Repositories, and Lists of build Work Tasks version 1 must be upgraded to build Work Tasks version 2 by running the AVEVAWorkTasksUpgradeManager.exe located at the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder. 3. After upgrading all the AVEVA Work Tasks components to build Work Tasks version 2 build, that DataSource can be used in any machine that has same version by using this feature.	
Repository Name	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the repository name.	Telematics Repository
Description	Provide a description for the repository.	This stores Telematics Employee Leave Details.
From mail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might occur in the Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be	Provide the valid email id that is specified in the **User Name** field while configuring **SMTP Settings** in the **Mail Server Settings** page. For more details, see Mail Server Setting.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	sent to the mail ID defined in **Exception to mail**. The From address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To avoid this, restart the Workflow Engine Service and Communication Service.	
Exception to mail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails is sent to the email ID specified in **Exception to mail**.	Provide the valid email id.
Alert initiator of the Workflow	Determines if the status of a workflow should be sent as a notification mail to the initiator of the workflow.	A process with a rule to check the pending leave available for the employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
Servers	Lists the servers available in the Farm. Repository will be available only to the mapped servers.	Work Tasks Server Name
Enterprise Consoles	Lists all the Enterprise Consoles for a selected AVEVA Work Tasks Server. Repository will be available only to the mapped Enterprise Console.	http://worktaskservername:8001/EnterpriseConsole
Enable Theme	Select this checkbox to enable the defined theme.	-
Theme Name	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. It provides **Default** and **Lite** themes.	-
Resource Provider	Lists the various default resource providers supported by AVEVA Work Tasks.	AVEVA Work Tasks Lists or Active Directory

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	Resource providers are also termed as User Provider. A User Provider is a data store containing information about the user. AVEVA Work Tasks also supports custom resource provider. For more information about Resource Provider, see User Providers. AVEVA Connect and Azure Active Directory resource providers are also available after you configure them. To configure these providers, see AVEVA Connect Settings and Azure Active Directory Settings. AVEVA Work Tasks allows secured LDAP configuration. The default port number for LDAPS is 636. You can map the repository in IDE and use the Secured LDAP. The logged-in Active Directory is displayed by default, and **Active Directory Datasource** expects **LDAP://** or **Fully Qualified Domain Name** as value.	
Custom Application Folder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with a semicolon.	-
Hide Repository	Select this option to make the repository unavailable to the Farm.	-
Source path	Enter the path to the Source Elements Folder.	-
Attachment Storage Path	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an absolute path. - The default path is the **Attachments** folder in *C:\ProgramData\AVEVA Work*	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	Tasks\. - The default path is used when no path is specified. - Ensure appropriate rights are set to the folder, where attachments are stored, for the following identities: -- The application pool identity for **Central Configuration** and **Enterprise Console(s)**. -- The account configured for the **Workflow Engine** service. - In case of mapped drives, ensure they are available to the identities. - In multiple-server environments, the path specified must be in a shared location.	

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Auto Log Off Settings for Enterprise Console	Automatically logs out the logged in user after the designated period of inactivity. Enter the time in minutes. This field accepts only valid integer values between 0 to 360. 0 minute means the user will not get automatically logged off. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. The default value for automatic log off is 30 minutes. You use this field to define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML.	0 to 360 (minutes)
Auto Log Off Settings for AVEVA Work Tasks Pro	Request for user credentials for AVEVA Work Tasks Pro app after the set time is elapsed. Enter the time in minutes. This field accepts only valid integer values between 0 to 2880. 0 minute means the user will not get automatically logged off. The default value is 30 minutes. If AVEVA Work Tasks Pro app is offline, the Auto Log Off will not get active, and the user will not be automatically logged off as per the defined settings.	0 to 2880 (in minutes)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Offline Support	Select this option to enable offline submission of items in AVEVA Work Tasks Pro app.	-
Refresh on Work Item submission	Select this option to enable Inbox refresh on Work Item submission in AVEVA Work Tasks Pro . Refresh option is enabled by default.	-
Timeout	Enter the time in seconds to set the timeout. If the value is set as 0, then there is no timeout. Default value is 60 seconds.	-
Label for Fill Form	Enter the label for Fill Form. The customized label will be displayed for the Fill Form in AVEVA Work Tasks Pro.	Work Tasks
Date Configuration Flag for Alternate Actors	Select the **Enable Date Settings** option to set the **From** date and **To** date as mandatory while creating alternate actors. For more information about creating alternate actors, see *Alternate Actors* topic in the *User Guide*.	-
Display Domain name with User Name	Select this checkbox to display the domain name along with the user name in the Enterprise Console and the List. The domain name will be displayed only: • If this checkbox is selected. • If the **Resource Provider** is selected as **Active Directory**.	-

User Providers

AVEVA Work Tasks supports the following User Providers by default:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Active Directory

When you select **Active Directory** as the **Resource Provider**, details such as **Active Directory datasource** and the **Active Directory search base** are populated by default.

Authentication User, **Authentication Password**, **User Log-on ID**, and **User Log-on Display Name** are optional fields. Access to the active directory might not be available for all users. In such cases, provide the active directory authentication user name and the active directory authentication password with which you can connect to the Active Directory.

AVEVA Work Tasks considers *SAM-Account-Name* as **User Log-on ID** by default. If you want to use active directory attributes other than the *SAM-Account-Name* user, then enter the desired active directory attribute details in the **User Log-on ID** field and **User Log-on Display Name** field. **User Log-on Display Name** is the user friendly display name of the active directory attribute. This **User Log-on Display Name** is referred in the User Lookup dialog box against the **User Log-on ID**.

Steps

1. When you select **Active Directory** as the **Resource Provider**, then the **Use Token Service** field appears. In this field, select one of the following:
   - If you do not want to use AIM authentication, select **Without Token Service**.
   - If you want to use AIM authentication, then select a server from the list of alias server names.
2. Enter **Active Directory datasource**.
3. Enter **Active Directory search base**.
4. Enter **Domain Name**.

   The domain name is case-sensitive.
5. Enter **Authentication User**.
6. Enter **Authentication Password**.
7. Check **Is Cross Domain Role Provider** checkbox, if cross domain users and roles mapping is required.

   Example: Users *John* and *Jim* of domain *TelematicsU* and users *Mike* and *Bob* from domain *TelematicsAU* are added to role Team HR of domain *TelematicsMain*.
8. Enter **User Log-on ID** if you want to use another Active Directory attributes instead of *SAM-Account-Name*.

   We recommend to use attributes of type *string* as input for **User Log-on ID.** However, *ObjectSID* and *objectGUID* are also supported.

   After you set the **User Log-on ID**, you cannot change it later.
9. Enter the **User Log-on Display Name**.

   After you set the **User Log-on Display Name**, you cannot change it later.

   This is the default display name of the user account of the active directory user used. **User Log-on Display Name** is used in user lookups.
10. Click **Test LDAP**.

    A message about successful LDAP test appears.

Note:
- Some of the active directory attributes or properties are not indexed. If you use an active directory attribute that is not indexed, it might impact the log-on validation performance based on the active directory entries. We recommend to use indexed attributes as **User Log-on ID** or index the attributes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- For an upgraded repository, you cannot change both **User Log-on ID** and **User Log-on Display Name**.
- You cannot localize **User Log-on ID** and **User log-on Display Name** fields.
- Disabled users will not be shown in user lookup search.

Using Forms

For Forms Resource Provider:

- If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.

1. In the **Resource Provider** drop-down list**,** select **Forms.**
2. In the **User Entity properties** field, enter XML data.

   Click **Sample Code** to get the XML Schema Template.

   Example

   ```
   <properties>
   <property name="userid" type="string" purpose="identifier" />
   <property name="username" type="string" purpose="name" />
   <property name="pwd" type="string" purpose="password" />
   </properties>
   ```
3. In the **User Provider Properties** field, enter the same XML Schema.

   There must be at least one identifier in the XML Schema. For example, UserID
4. In the **Datasource** field, enter a valid data source or specify a dll.

   The value for the data source can be specified as **Physical path of the dll** or **Physical path of the dll::param. Param** is not mandatory.

   See topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Build Business Process* section for creating the assembly.

   Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly, whereas one class can return multiple connection strings.

   Examples of data source parameter:

   - c:\DLLStore\GetConnectionString.dll
   - c:\GetConnectionString.dll::Leave

   If there are two data sources, *Leave* and *Licensing*, and only one DLL is used to return a connection string based on the data source selected. The parameter can be specified as **c:\GetConnectionString.dll::Leave** or **c:\ GetConnectionString.dll::Licensing**. The connection string can be returned from the connection dll based on this parameter.
5. Enter a proper table name or view name which holds the User Name and Password for the Authentication purpose.

   Example: Passmast (the table which is available in IMS_NEW Database)
6. Click **Test SQL Connection**.

   A message about successful test connection appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using ASP.NET Membership

This resource provider lists users defined in the asp.net membership database.

Lists

This resource provider lists users created using AVEVA Work Tasks Users Creation. If you are unable to determine a User Provider to create a repository, use AVEVA Work Tasks Lists.

To verify successful creation of the Repository:

1. Click **Enterprise Console** URL from the site path column.
   The repository created is listed.
   Add the created Enterprise Console link into "Add to Favorites" to easily navigate in future.
2. Click **Repository User Provider** link to logon to the repository.
   The Enterprise Console log-on page appears.
3. Enter the logon details, and then click **Login**.
   The Enterprise Console home page appears. This indicates successful creation of a Repository.
   All features can only be available for a user with administrative privileges.

Note: Consider you created the Repository and then deleted it. If you create the same Repository again immediately and try to log into it through Enterprise Console, the Provider does not list the Repository. You need to access IIS and recycle AVEVA Work Tasks from the Application Pool to get the Repository listed in the Provider list..

Verify the Installation

After you install AVEVA Work Tasks, verify that the product is installed successfully. To verify successful installation, create a simple Workflow using Process Designer. The tasks includes the following:

- Creating a Workflow
- Publishing the Workflow
- Testing the Workflow

Creating a Workflow

1. Log on to the **Enterprise Console**.
2. If the Repository is configured for Active Directory provider, go to step 4.
3. Enter the **User Name** and **Password**.
   The **User Name** and **Password** will be validated with the user provider selected while creating the repository.
   The default authentication provider is the Repository User Provider List.
4. In the **Enterprise Console**, click menu icon.
   All features can only be available for a user with administrative privileges.
5. From the menu, select **Workflows**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Workflow window appears.

6. On the Action bar, click **New**, and then click **Workflow**.
   The New Workflow window appears.
7. Enter **Title** and **Description**.
8. Click **Save & Continue**.
   The successful message appears.
9. Select **Design the Workflow now**.
   This is the default option.
10. Click **Finish**.
    The Process Designer window appears.
11. From the activities pane on the left, drag the Information activity to the Process Designer window.
12. The activity *Information1* is created.
    Click to create a link between two activities.
    The Link Action window appears.
    a. Select from the **From Action** drop-down box.
    b. Select from the **To Action** drop-down box.
    c. Click **OK**.
       The link is established.
13. Right-click *Information1*.
14. Select **Activity Properties** from the context menu.
    The Activity Properties pane appears.
    a. Enter **Name** and **Description**.
    b. Click to **Assign Actor(s)**.
       The Assign Actor(s) window appears.
    c. Select the required provider from the **Providers** drop-down list.
    d. Select values from the drop-down list.
    e. Click **Add**.
    f. Click **Update**.
15. Click to enter the **Subject** for Notification Contents.
    The window to enter subject for notification appears.
16. Click **Update**.
17. Click to enter the **Body** for Notification Contents.
    The window to enter the Body content appears.
18. Click **Update**.
19. From the Activity Properties pane, click **Save**.

Publishing the Workflow

1. On the **Publish** menu, click **Publish**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Published successful message appears.

2. Close the dialog box.

   The Enterprise Console screen lists the workflow created.

Testing the Workflow

1. Select the listed workflow.
2. Right-click the selected workflow.
3. Click **Publish and Test Run**.

   The **Status Finished successfully** appears.

   If Status is **Execution Pending** in the Process Execution window, then click on the **Refresh** button to get the status.
4. In the **Enterprise Console** menu, select **Inbox**.

   The Inbox lists the work item.
5. Click to view the work item listed.

   This indicates that the installation is successful.

Intranet and Extranet Web Servers on Different Machines

This section describes the activities involved while installing the Single Server for the scenario Intranet and Extranet Web Servers on Different Machines.

In this scenario, the Server is installed on Machine 1, the Database is installed on Machine 2 and the Client is installed on multiple Web Farms namely Machine 3 (or Server 1), Server 2 and Server N. "Server N" represents the number of machines in a Web Farm.

Ensure that you install the product in the following sequence:

- High-Level View of Server Installation
- High-Level View of Client Installation

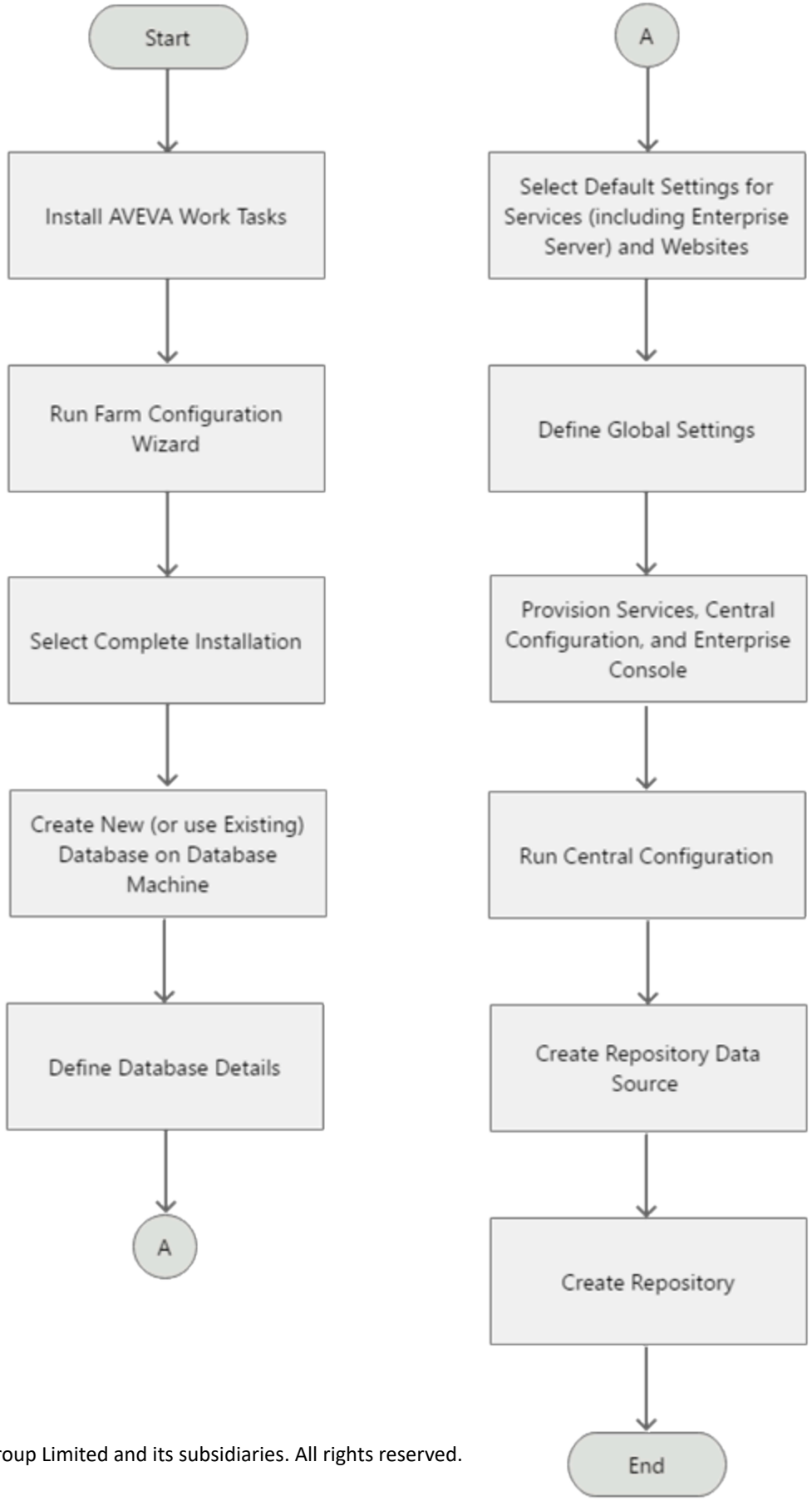
See Also: Setting up Intranet and Extranet Web Fronts

High-Level View of Server Installation

This section provides a high-level view of the various installation steps. The objective of this section is to provide an overview of the various activities involved while installing AVEVA Work Tasks Server. This section also enables you to visually keep track of your activities while following the sequence of tasks.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



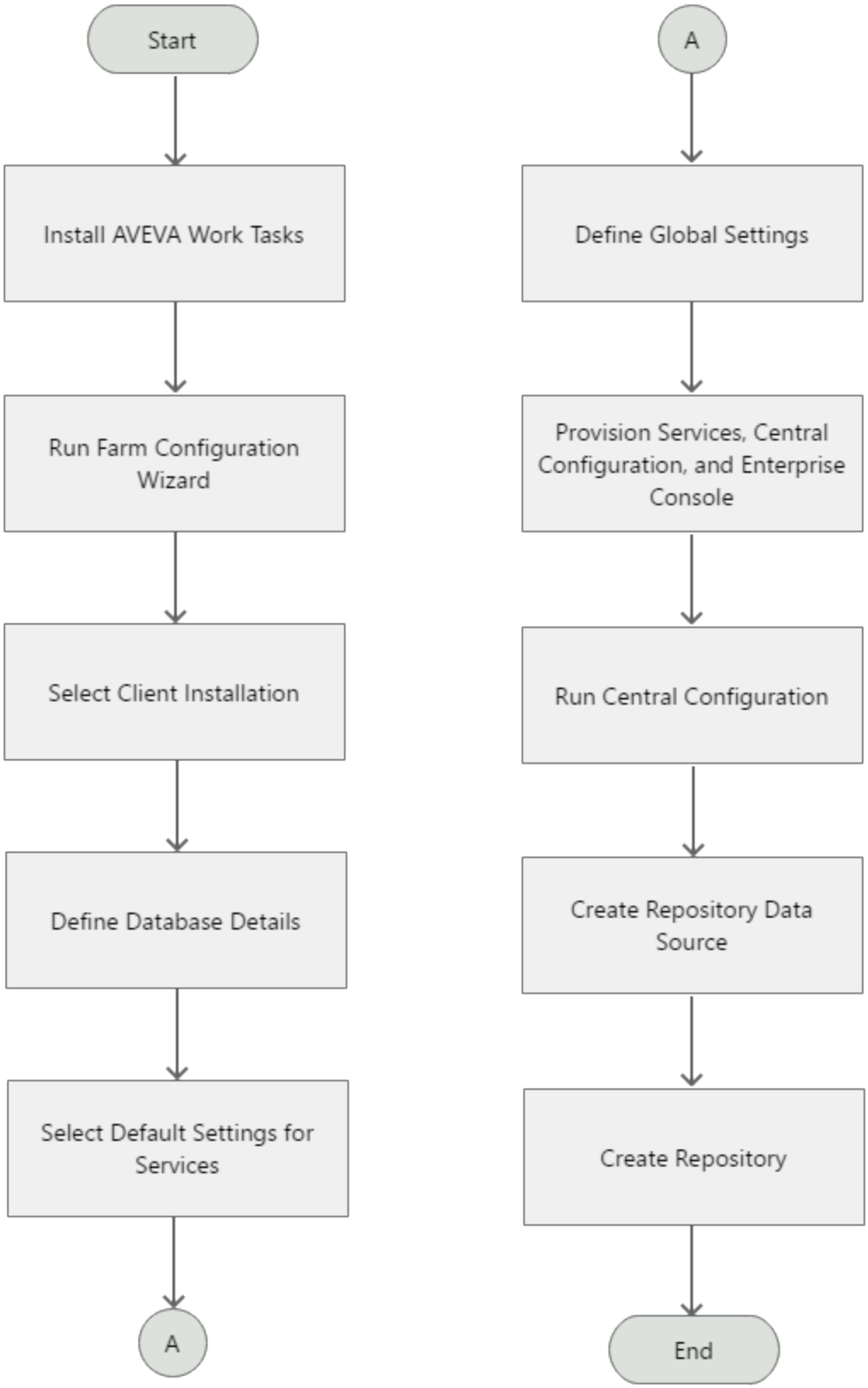
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

High-Level View of Client Installation

This section provides a high-level view of AVEVA Work Tasks Client installation. The Client components are installed on Web Servers. For more information on this scenario, see Wonderware Skelta BPM Professional Edition.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation

This section provides installation steps for the scenario Intranet and Extranet Web Servers on Different Machines.

The installation includes the following:

Installing Workflow Management Server

Installing Workflow Management Client

AVEVA Work Tasks Server has to be installed before installing the Client. In this scenario, the Client has to be installed on multiple machines on separate Web Farms.

AVEVA Work Tasks Server

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
4. Click **Next**.

   The License Server details are configured successfully.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.
4. Click **Next**.

   The **Edit Farm Settings** window appears.
5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.
6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
9. Click **Next**.

   The License Server details are configured successfully.

Select the type of installation

To select the type of installation:

1. In the Select the type of installation page, select **Complete Installation (Server)**.

   A Complete Standalone Deployment does not require a client installation.
2. Under **Create AVEVA Work Tasks Farm**:
   - Click **Create AVEVA Work Tasks Farm database** to create a new Farm database.
   - Click **Use existing AVEVA Work Tasks Farm database (Creates AVEVA Work Tasks Farm tables only if they do not exist)** to use an existing database as the Farm database.
3. Click **Next**.

Create Farm Database

Use the Create AVEVA Work Tasks Farm Database page to provide **New Database** details. Select the **Connection String** option to establish a connection with the database. This connection information is encrypted and stored.

You can enter either Windows or Microsoft SQL Server user settings for authentication.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:
- The Farm database password does not support **&** and **;** special characters.
- While setting up **Farm Database** with **Windows Authentication**, do not enter **FQDN** in the **Domain Name** field.

To Create Farm Database

1. Enter **Database Title**.
2. Select **Connection String**.

   See *Connection Provider* later in this topic, if you have selected the **Connection Provider** option.

   - Enter the following details:

     Server Name:

     Provide information about the database server on which the Farm database can be installed.

     If the **Database Type** is **SQL Server**, then you can specify an IPv6 address for the **Server Name**.

     If SQLExpress Instance need to be used as a Database instance of the Farm database, then **Server Name** should be <server name>\SQLExpress.

     To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

   - **Authentication Type**

     This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports Windows and SQL Server authentication process.

     Example: Windows
     If Windows Authentication is selected, the user configured for the Farm configuration will be used to access the database. This is the authentication recommended by AVEVA Work Tasks.

   - **Login**

     This field is used to enter the SQL login credential for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Password**

     This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Advanced Settings**

     This is an optional field. It is used to enter any additional connection information, such as connection Timeout. Example: Connection Timeout = 20.

     While creating a data source for repository, use the following custom settings to set properties for connection pool. These settings are case sensitive and should be mentioned only in lower case.

     ```
     encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
     ```

     encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

     enableltm: AVEVA Work Tasks internally implements light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive. This property can be used to turn off this

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

feature. The default value is set as True.

connectionretrywaittime: While opening a new connection, if the data base server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

maxconnectionretries: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

3. Click **Test Connection** to ensure database connectivity.
4. Click **Next**.

   The message *AVEVA Work Tasks Farm is Created* appears.
5. Click **OK**.

AlwaysOn Availability Groups feature of Microsoft SQL Server

The **AlwaysOn Availability Groups** feature of Microsoft SQL Server is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in Microsoft SQL Server 2012, AlwaysOn Availability Groups maximizes the availability of a set of user databases for an enterprise.

After you configure the AVEVA Work Tasks, you must add the created Farm Database to the SQL Server AlwaysOn Availability Groups. Contact your Database Administrator to add the database to SQL Server AlwaysOn Availability Groups.

Notes:

- The authentication user needs different rights assigned for different scenarios. If a user wants to create database (Farm or Repository Data Source Database), Authentication User needs to have minimum 'dbcreator' access rights on Database Server Instance.

- If database is already created, user needs to have following permissions on Database Server:

- SELECT, INSERT, UPDATE, and DELETE permissions.

- EXECUTE permission for Stored Procedures.

To choose Connection Provider

Complete the following steps to choose a connection provider:

1. Enter **Database Title**.
2. In the **Database Type** box, select a database type.
3. Select **Connection Provider**.

   This option is used to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.
4. Enter the following details:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Assembly Path**:

  Click the icon to browse and choose the required dll file.

  If custom properties (enableltm, connectionretrywaittime, maxconnectionretries and sqlcommandtimeout) are required to be used in Assembly connection string, then user should select **Use existing Farm database** while creating a the Farm. See Creating or Using an Existing Database.

  See the topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub chapter in the *Build Business Process* section for creating the assembly. Make sure that, there should be only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.
6. Click **Next**.
7. The message *AVEVA Work Tasks Farm is created* appears.
8. Click **OK**.

Define Security Settings

You can define the Security Settings using the Farm Configuration Wizard.

For more information on Security settings, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client.

To define the Security Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Security Settings** and **Service Settings** and then, click **Next**.
4. Select **Secure Communication** to establish secured communication between the AVEVA Work Tasks services through HTTPS.
5. Select **Secure Web Sites** check box to configure secured web sites. Uncheck the **Secure Web Sites** check box to use HTTP sites. By default the sites are created in HTTPS mode.
6. Enter the Port number. The default port for the HTTPS site is 8001, and for the HTTP site is 8000. You can enter any port number that IIS accepts. But if any other services or applications are already using the entered port number, AVEVA Work Tasks will not start.
7. Select the available certificate from the **Certificate** list. If your certificate is not listed, click the **Refresh** icon to repopulate the list.

   Note: If the required certificate is not listed, generate one using the System Management Server (Configurator tool). Do not use a self-signed certificate without a proper root chain.
8. Click the **Validate** button to validate the certificate before configuring the security settings.

   Note: An Internet connection is needed to validate the certificate.
9. Click **Configure**, then **Next**, and then **Finish** to apply the new security settings.

Note:
- To update the port number, certificate, and websites from unsecure to secure or vice-versa, select **Security**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings and **Service Settings** check box in **Edit AVEVA Work Task Farm Settings** window before updating the values. After the required details are provided, these settings get updated only when you click **Configure**, then **Next**, and then **Finish** as you navigate through the Farm Configuration Wizard..
- If the Quickflow is already deployed on a machine, you need to deploy it again after you update the security settings. If secured web site is configured, then you cannot change the server configuration for Quickflow in Central Configuration. However, if the **Secure Web Sites** checkbox is not selected, you can make the changes to the server configuration.
- Advance Server communication cannot be configured on HTTP/HTTPS.
- For resources of Load Balanced Server, ensure to configure all services with the same protocol (HTTP or HTTPS).
- It is recommended to use Standard SSL certificate instead of self-signed SSL certificate.

Using ASB Certificate for Secured Communication

ASB Certificate is a PCS-managed SSL certificate and renews automatically every 30 days when the certificate is still valid. The certificate is valid for two years. The PCS watchdog service on the runtime node checks the validity of the SSL certificate within 5 minutes from when it starts and once a day. PCS also renews the SSL certificate if it is expired.

If the runtime node is offline and later back online, the PCS watchdog service regularly checks the certificate validity and renews it if it is required. Consider restarting the PCS watchdog service to enforce the SSL certificate validity check within 5 minutes from the restarting time.

Furthermore, when the SSL certificate is renewed, PCS binds it to port 443 (by default) and changes the SslThumbprint PCS registry value. If AVEVA Work Tasks sites are hosted on port 443, certificate binding will be auto-renewed when the SSL certificate is renewed. It is recommended to check the bindings in IIS with every certificate renewal to ensure the application is running seamlessly.

To disable the certificate renewal, consider disabling the **ArchestrA Certificate Renewal Service** windows service.

Note: Client Service will try to automatically bind the renewed certificate. If there are any issues in browsing the website or starting the services after certificate renewal, users need to reconfigure the secure communication in Farm Configuration. To reconfigure, see the steps mentioned in above section *To define the Security Settings*.

Securing Communication between Services

To have a secure communication between the services, apply the below changes to the Windows registry and reboot the server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"SchUseStrongCrypto" = dword:00000001
```

For more information on securing communication, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client and

https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2015/2960358.

Define Services and Global Settings

You can define the Services and Global Settings using the Farm Configuration Wizard or through Configuration XMLs.

To define Services

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.

   Note: You can also customize the settings for these services and websites using the **Custom Settings** option. For more information about Service-related changes and Website-related changes, see Advanced Settings and Administering Web Servers respectively.

2. Click **Next**.

   The Global Settings dialog box appears.

3. In the **Communication** tab, enter key fields in **Email Settings** and **Exception Email settings** to configure exception e-mails.

   The following are the details:

   - **Host**: Enter information related to mail server.
     Example: Enter details related to Microsoft Exchange Server (IP address of the exchange server).
   - **Port**: Enter the port number for the mail server. The default port number is 25.
   - **From email**: Enter the from e-mail address.
     Example: admin@telematics.com
   - **To email**: Enter the to e-mail address.
     Example: alex@telematics.com

4. Click **OK**.

Note: To confirm the exception e-mail settings, click **Test SMTP**. An e-mail will be generated and sent to the **To email** address confirming successful configuration of the e-mail. The advanced settings like **Secure Socket Layer (SSL), Email Timeout, User Name,** and **Password** for the e-mail account holder, **Password Encrypted** flag will differ based on the mail server settings.

Define Central Configuration

1. Run **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Farm Configuration Wizard window appears.

2. Click **Next**.
3. The Edit Farm settings window appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Select **Service Settings**.
5. Click **Next**.

   The Services/Global Settings window appears.
6. Click against **Central Configuration** to edit AVEVA Work Tasks **Central Configuration settings**.

   The Central Configuration Settings page appears.
   - **http/https**: This option enables you to determine if you want a secured (https) or a non-secured site (http).
   - **Authentication Type**: This option allows you to provide the authentication type to access the Central Configuration. The available authentication types are Windows, AVEVA Connect, and Azure Active Directory.

     Note: The AVEVA Connect and Azure Active Directory authentication types are available only if these user providers are configured in the system.
     To configure AVEVA Connect user provider, see AVEVA Connect Settings.
     To configure Azure Active Directory user provider see Azure Active Directory Settings.
   - **Create at**: This field allows you to provide the Internet Information Service (IIS) path where the new site is created.
   - **Port Number**: This field is used to provide a port number for the Central Configuration.
7. Click against **Create at** to change the path of the Central Configuration.

   A message indicating that if the path has been changed the Central Configuration will no longer be accessible appears.
8. Click **Yes**.

   The IIS structure screen is displayed. Select the desired IIS path for the Central Configuration.
9. Click **OK**.

   The selected URL appears in the **Central Configuration Settings** screen.
10. Click **OK**.

Note:

- The default IIS path for Central Configuration is displayed. Click the edit icon to change the site path.

- If you want to update the authentication type to AVEVA Connect or Azure Active Directory of the already configured Central Configuration, you need to reconfigure the AVEVA Connect and Azure Active Directory user providers for the Central Configuration. For information on configuring user providers, see AVEVA Connect Settings and Azure Active Directory Settings.

- Central Configuration cannot be accessed in Chrome browser when configured with unsecured access (http://) for Azure Active Directory and AVEVA Connect authentication types.

Provision Services

While the installation is in progress, you can view the services that will be installed. This provides information on the progress of installation. At the end of the installation, a status indicating the completion of installation appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Click **OK** to complete the installation.

AVEVA Work Tasks Client

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
4. Click **Next**.

   The License Server details are configured successfully.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: The Farm Configuration Wizard should be run with administrative privileges.

4. Click **Next**.

   The **Edit Farm Settings** window appears.

5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.

6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

Note: AVEVA Work Tasks does not validate the License Server details entered.

9. Click **Next**.

   The License Server details are configured successfully.

Select Client Installation

1. In the Select the type of installation page, select **Client**.
2. Click **Next**.

   The Database User Settings page appears.

Connect to an Existing Farm Database

You can use the Connect to existing database dialog box to connect to the existing Farm. You can select the Connection String or Connection provider for the new database.

1. Select the computer name on which the Farm Database is installed from the **Server Name** field.
2. Select **Connection String**.
3. Enter the following details:
   a. **Authentication Type**: This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports both Windows and SQL server authentication process. Example: Windows.
   b. **User ID**: This field is used to enter the SQL user ID for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   c. **Password**: This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   d. **Advanced Settings**: This is an optional field. It is used to enter any additional connection information such as connection Timeout etc. Example: Connection Timeout = 20.

      While creating a data source for repository, use the following custom settings to set properties for the connection pool.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;

encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks controls the transaction from escalating to a distributed transaction which is very expensive.

The above property can be used to turn off this feature. The default value is set as True.

connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

Note: These settings are case sensitive and should be mentioned only in lower case.

e. **Database**: This field is used to select the machine with database installed. Example: FarmDB.

AVEVA Work Tasks will validate if at least one server is installed.

4. Click **Test Connection** to ensure database connectivity.

   You must have the following permissions on the Database Server:

   SELECT, INSERT, UPDATE, and DELETE permissions.

   EXECUTE permission for Stored Procedures.
5. Click **Next** to continue with the wizard.
6. Click **OK**.

Define and Provision the Enterprise Console

You can define the required AVEVA Work Tasks services.

Steps to Define Enterprise Console

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.
2. Click **Next**.

   The **Installing Services** screen appears, indicating the progress of installation.

   A message *Installation is Complete* appears after all the services are installed.
3. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create a Repository DataSource

A **Repository DataSource** is used to store all repository related data. The **Repository DataSource** must be created on a machine where the **Central Configuration** site is installed.

Example

A TelematicsDataSource is created for TelematicsHR Repository.

A Repository contains Workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating the repository.

To create a Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**.

   The Central Configuration web page appears.
2. Click **DataSource Settings**.

   The Datasource List page appears.
3. Click **Add**.

   The Manage Datasource page appears.
4. Enter the required details.
   a. **Name**: Enter a valid data source name. Example: *TelematicsDataSource*.
   b. **Description**: Enter a description for the data source. Example: *This data source is used to store connection details*.
   c. **Connection String**: Select this option to establish a connection with the database. The connection string contains fields that that defines the connection. The connection information is encrypted and stored.

      Server Name

      To use a database on the local machine, enter "localhost", ".", or the local computer's name for the **Server Name**.

      Note: Server Name can be machine name/IP address/SQL Server AlwaysON/SQL Server instance name. Instead of entering "local host", it is preferable to give the local machine name.

      If the **Database Type** is **SQL Server**, then specify an IPV6/IPV4 address for the **Server Name**.
      To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: *localhost,1455*
      To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be *<server name>\SQLExpress*.

      Authentication Type

      Use this field to determine the process to validate a user who is trying to connect to the database. AVEVA Work Tasks uses Windows and SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. The **User ID** and **Password** field is disabled if Windows Authentication is selected. Example: *Windows*

      Advanced Settings

      Use this field to define advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection. Example: *Connection Timeout = 20*
      or
      For defining maximum connections that can be opened in a connection pool.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example: *max pool size = 250*
Default connection pool is set as 250. We recommend to enter a higher value on heavily loaded environment.
While creating a data source for a repository, use the following custom settings to set properties for the connection pool.
encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.
enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature, AVEVA Work Tasks controls the transaction from escalating to an expensive distributed transaction.
The above property can be used to turn off this feature. The default value is set as *True*.
connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is 30 seconds. The value should be set in seconds.
maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is infinite.
sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.
These settings are case sensitive and should be mentioned only in lower case.

Database:

Specify or select a database that can be used as a repository database. This database stores all repository related data. The repository database is created if it does not exist on the database server instance.
Example: *TelematicsDB*

Enable Table Partitioning:

Select to enable database table partitioning. For more information, see Partitioning Database.

d. **Connection Provider**: Use this to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.

The selected dll must be copied to all the machines, all servers and all clients, that are connected to the current Farm.

Assembly Path

Click to browse and choose the required dll file.
To use custom properties (enableltm, connectionretrywaittime, maxconnectionretries, and sqlcommandtimeout) in the Assembly connection string, the database must be created before.
See *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Developer Guide* for creating the assembly. Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.

   A message about successful test connection appears.

6. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

A message about successful creation of the data source appears.The message also states that the required tables will be created in the data source after a repository is associated with the data source.

When a data source is saved, an entry is created for the data source in the **Farm Database**. When a first repository is created for the data source, AVEVA Work Tasks tables are added to the data source. While creating a data source, the archival database is created with the same database name suffixed by *_BI*. This database is used for archival in future use.

Note: After you configure the AVEVA Work Tasks, you must add the created **Farm Database** to the **SQL Server AlwaysOn Availability Groups**. Contact your Database Administrator to add the database to **SQL Server AlwaysOn Availability Groups**.

To view the data source created, click **DataSource Settings** link.

Create a Repository

A **Repository** contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

To create a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The Central Configuration page appears.
2. Click **Repository Settings**.
   The Repository List page appears.
3. Click **Add**.
   Note: If the number of repositories exceeds the permitted limit for the license, then the message "You have exceeded the maximum number of repositories" appears, and the repository creation is stopped. Click **Close**.
   The **Manage Repository** page appears.
4. Enter the required details.
5. Click **OK**.
   The **Config Administrator** page appears.
6. Click to configure an administrator.
   The User Lookup dialog box appears.
7. Select the provider from the provider drop-down list.
8. Enter the details to search the administrator for the repository that is created.
   Note: You can also search a user based on *ObjectSID* and *objectGUID* (applicable only for Active Directory provider).
9. Click **Search**.
   The user is listed under **Available Resources**.
10. Select and click the arrow to list the user under **Selected Resource**.
11. Click **Update**.
    The selected user is displayed in **User Lookup**.
12. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

A message confirming the successful creation of an administrator appears.

13. Click **OK**.

    The **Repository List** page appears.

    For more information about configuring the system user, see Configure System User.

Manage Repository Window Fields

The following table explains the fields of the Manage Repository window:

Field Name	Description	Example
Repository DataSource	Lists all the data sources created in a Farm. The selected data source will be used to store data for the current repository.	TelematicsDatasource
Attach DataSource	Select Attach DataSource checkbox and then select a repository from the selected DataSource .Data associated with the selected repository will be copied to the new repository. The version of the installed product and destination DataSource should be the same as that of the DataSource from which the repository data is to be copied. For example, in an upgraded environment, from build Work Tasks version 1 to Work Tasks version 2, you cannot attach a DataSource created in build Work Tasks version 1 to build Work Tasks version 2. To match the build version of the installed product to the version of the DataSource: 1. If the DataSource was created in build Work Tasks version 1, then the entire build Work Tasks version 1 must be upgraded to Work Tasks version 2 by running the product installer in the upgrade mode. 2. All the Farm, DataSources, Repositories, and Lists of build	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	Work Tasks version 1 must be upgraded to build Work Tasks version 2 by running the AVEVAWorkTasksUpgradeManager.exe located at the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder. 3. After upgrading all the AVEVA Work Tasks components to build Work Tasks version 2 build, that DataSource can be used in any machine that has same version by using this feature.	
Repository Name	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the repository name.	Telematics Repository
Description	Provide a description for the repository.	This stores Telematics Employee Leave Details.
From mail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might occur in the Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be sent to the mail ID defined in **Exception to mail**. The From address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To avoid this, restart the Workflow Engine Service and Communication	Provide the valid email id that is specified in the **User Name** field while configuring **SMTP Settings** in the **Mail Server Settings** page. For more details, see Mail Server Setting.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	Service.	
Exception to mail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails is sent to the email ID specified in **Exception to mail**.	Provide the valid email id.
Alert initiator of the Workflow	Determines if the status of a workflow should be sent as a notification mail to the initiator of the workflow.	A process with a rule to check the pending leave available for the employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
Servers	Lists the servers available in the Farm. Repository will be available only to the mapped servers.	Work Tasks Server Name
Enterprise Consoles	Lists all the Enterprise Consoles for a selected AVEVA Work Tasks Server. Repository will be available only to the mapped Enterprise Console.	http://worktaskservername:8001/EnterpriseConsole
Enable Theme	Select this checkbox to enable the defined theme.	-
Theme Name	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. It provides **Default** and **Lite** themes.	-
Resource Provider	Lists the various default resource providers supported by AVEVA Work Tasks. Resource providers are also termed as User Provider. A User Provider is a data store containing information about the user. AVEVA Work Tasks also supports custom resource provider. For more information about Resource Provider, see User Providers.	AVEVA Work Tasks Lists or Active Directory

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	AVEVA Connect and Azure Active Directory resource providers are also available after you configure them. To configure these providers, see AVEVA Connect Settings and Azure Active Directory Settings. AVEVA Work Tasks allows secured LDAP configuration. The default port number for LDAPS is 636. You can map the repository in IDE and use the Secured LDAP. The logged-in Active Directory is displayed by default, and **Active Directory Datasource** expects **LDAP://** or **Fully Qualified Domain Name** as value.	
Custom Application Folder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with a semicolon.	-
Hide Repository	Select this option to make the repository unavailable to the Farm.	-
Source path	Enter the path to the Source Elements Folder.	-
Attachment Storage Path	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an absolute path. - The default path is the **Attachments** folder in *C:\ProgramData\AVEVA Work Tasks*. - The default path is used when no path is specified. - Ensure appropriate rights are set to the folder, where attachments are stored, for the following identities: -- The application pool identity for **Central Configuration** and	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	Enterprise Console(s). -- The account configured for the **Workflow Engine** service. - In case of mapped drives, ensure they are available to the identities. - In multiple-server environments, the path specified must be in a shared location.	
Auto Log Off Settings for Enterprise Console	Automatically logs out the logged in user after the designated period of inactivity. Enter the time in minutes. This field accepts only valid integer values between 0 to 360. 0 minute means the user will not get automatically logged off. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. The default value for automatic log off is 30 minutes. You use this field to define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML.	0 to 360 (minutes)
Auto Log Off Settings for AVEVA Work Tasks Pro	Request for user credentials for AVEVA Work Tasks Pro app after the set time is elapsed. Enter the time in minutes. This field accepts only valid integer values between 0 to 2880. 0 minute means the user will not get	0 to 2880 (in minutes)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	automatically logged off. The default value is 30 minutes. If AVEVA Work Tasks Pro app is offline, the Auto Log Off will not get active, and the user will not be automatically logged off as per the defined settings.	
Offline Support	Select this option to enable offline submission of items in AVEVA Work Tasks Pro app.	-
Refresh on Work Item submission	Select this option to enable Inbox refresh on Work Item submission in AVEVA Work Tasks Pro . Refresh option is enabled by default.	-
Timeout	Enter the time in seconds to set the timeout. If the value is set as 0, then there is no timeout. Default value is 60 seconds.	-
Label for Fill Form	Enter the label for Fill Form. The customized label will be displayed for the Fill Form in AVEVA Work Tasks Pro.	Work Tasks
Date Configuration Flag for Alternate Actors	Select the **Enable Date Settings** option to set the **From** date and **To** date as mandatory while creating alternate actors. For more information about creating alternate actors, see *Alternate Actors* topic in the *User Guide*.	-
Display Domain name with User Name	Select this checkbox to display the domain name along with the user name in the Enterprise Console and the List. The domain name will be displayed only: • If this checkbox is selected. • If the **Resource Provider** is	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	selected as **Active Directory**.	

User Providers

AVEVA Work Tasks supports the following User Providers by default:

Active Directory

When you select **Active Directory** as the **Resource Provider**, details such as **Active Directory datasource** and the **Active Directory search base** are populated by default.

Authentication User, **Authentication Password**, **User Log-on ID**, and **User Log-on Display Name** are optional fields. Access to the active directory might not be available for all users. In such cases, provide the active directory authentication user name and the active directory authentication password with which you can connect to the Active Directory.

AVEVA Work Tasks considers *SAM-Account-Name* as **User Log-on ID** by default. If you want to use active directory attributes other than the *SAM-Account-Name* user, then enter the desired active directory attribute details in the **User Log-on ID** field and **User Log-on Display Name** field. **User Log-on Display Name** is the user friendly display name of the active directory attribute. This **User Log-on Display Name** is referred in the User Lookup dialog box against the **User Log-on ID**.

Steps

1. When you select **Active Directory** as the **Resource Provider**, then the **Use Token Service** field appears. In this field, select one of the following:
   - If you do not want to use AIM authentication, select **Without Token Service**.
   - If you want to use AIM authentication, then select a server from the list of alias server names.
2. Enter **Active Directory datasource**.
3. Enter **Active Directory search base**.
4. Enter **Domain Name**.

   The domain name is case-sensitive.
5. Enter **Authentication User**.
6. Enter **Authentication Password**.
7. Check **Is Cross Domain Role Provider** checkbox, if cross domain users and roles mapping is required.

   Example: Users *John* and *Jim* of domain *TelematicsU* and users *Mike* and *Bob* from domain *TelematicsAU* are added to role Team HR of domain *TelematicsMain*.
8. Enter **User Log-on ID** if you want to use another Active Directory attributes instead of *SAM-Account-Name*.

   We recommend to use attributes of type *string* as input for **User Log-on ID.** However, *ObjectSID* and *objectGUID* are also supported.

   After you set the **User Log-on ID**, you cannot change it later.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Enter the **User Log-on Display Name**.

   After you set the **User Log-on Display Name**, you cannot change it later.

   This is the default display name of the user account of the active directory user used. **User Log-on Display Name** is used in user lookups.

10. Click **Test LDAP**.

    A message about successful LDAP test appears.

Note:
- Some of the active directory attributes or properties are not indexed. If you use an active directory attribute that is not indexed, it might impact the log-on validation performance based on the active directory entries. We recommend to use indexed attributes as **User Log-on ID** or index the attributes.
- For an upgraded repository, you cannot change both **User Log-on ID** and **User Log-on Display Name**.
- You cannot localize **User Log-on ID** and **User log-on Display Name** fields.
- Disabled users will not be shown in user lookup search.

Using Forms

For Forms Resource Provider:

- If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.

1. In the **Resource Provider** drop-down list**,** select **Forms.**
2. In the **User Entity properties** field, enter XML data.

   Click **Sample Code** to get the XML Schema Template.

   Example

   ```
   <properties>
   <property name="userid" type="string" purpose="identifier" />
   <property name="username" type="string" purpose="name" />
   <property name="pwd" type="string" purpose="password" />
   </properties>
   ```

3. In the **User Provider Properties** field, enter the same XML Schema.

   There must be at least one identifier in the XML Schema. For example, UserID

4. In the **Datasource** field, enter a valid data source or specify a dll.

   The value for the data source can be specified as **Physical path of the dll** or **Physical path of the dll::param. Param** is not mandatory.

   See topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Build Business Process* section for creating the assembly.

   Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly, whereas one class can return multiple connection strings.

   Examples of data source parameter:

   - c:\DLLStore\GetConnectionString.dll
   - c:\GetConnectionString.dll::Leave

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If there are two data sources, *Leave* and *Licensing*, and only one DLL is used to return a connection string based on the data source selected. The parameter can be specified as **c:\GetConnectionString.dll::Leave** or **c:\ GetConnectionString.dll::Licensing**. The connection string can be returned from the connection dll based on this parameter.

5. Enter a proper table name or view name which holds the User Name and Password for the Authentication purpose.

   Example: Passmast (the table which is available in IMS_NEW Database)

6. Click **Test SQL Connection**.

   A message about successful test connection appears.

Using ASP.NET Membership

This resource provider lists users defined in the asp.net membership database.

Lists

This resource provider lists users created using AVEVA Work Tasks Users Creation. If you are unable to determine a User Provider to create a repository, use AVEVA Work Tasks Lists.

To verify successful creation of the Repository:

1. Click **Enterprise Console** URL from the site path column.

   The repository created is listed.

   Add the created Enterprise Console link into "Add to Favorites" to easily navigate in future.

2. Click **Repository User Provider** link to logon to the repository.

   The Enterprise Console log-on page appears.

3. Enter the logon details, and then click **Login**.

   The Enterprise Console home page appears. This indicates successful creation of a Repository.

   All features can only be available for a user with administrative privileges.

Note: Consider you created the Repository and then deleted it. If you create the same Repository again immediately and try to log into it through Enterprise Console, the Provider does not list the Repository. You need to access IIS and recycle AVEVA Work Tasks from the Application Pool to get the Repository listed in the Provider list..

Verify the Installation

After you install AVEVA Work Tasks, verify that the product is installed successfully. To verify successful installation, create a simple Workflow using Process Designer. The tasks includes the following:

- Creating a Workflow
- Publishing the Workflow
- Testing the Workflow

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Creating a Workflow

1. Log on to the **Enterprise Console**.
2. If the Repository is configured for Active Directory provider, go to step 4.
3. Enter the **User Name** and **Password**.
   The **User Name** and **Password** will be validated with the user provider selected while creating the repository.
   The default authentication provider is the Repository User Provider List.
4. In the **Enterprise Console**, click menu icon.
   All features can only be available for a user with administrative privileges.
5. From the menu, select **Workflows**.
   The Workflow window appears.
6. On the Action bar, click **New**, and then click **Workflow**.
   The New Workflow window appears.
7. Enter **Title** and **Description**.
8. Click **Save & Continue**.
   The successful message appears.
9. Select **Design the Workflow now**.
   This is the default option.
10. Click **Finish**.
    The Process Designer window appears.
11. From the activities pane on the left, drag the Information activity to the Process Designer window.
12. The activity *Information1* is created.
    Click to create a link between two activities.
    The Link Action window appears.
    a. Select from the **From Action** drop-down box.
    b. Select from the **To Action** drop-down box.
    c. Click **OK**.
       The link is established.
13. Right-click *Information1*.
14. Select **Activity Properties** from the context menu.
    The Activity Properties pane appears.
    a. Enter **Name** and **Description**.
    b. Click to **Assign Actor(s)**.
       The Assign Actor(s) window appears.
    c. Select the required provider from the **Providers** drop-down list.
    d. Select values from the drop-down list.
    e. Click **Add**.
    f. Click **Update**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

15. Click to enter the **Subject** for Notification Contents.
    The window to enter subject for notification appears.
16. Click **Update**.
17. Click to enter the **Body** for Notification Contents.
    The window to enter the Body content appears.
18. Click **Update**.
19. From the Activity Properties pane, click **Save**.

Publishing the Workflow

1. On the **Publish** menu, click **Publish**.
   The Published successful message appears.
2. Close the dialog box.
   The Enterprise Console screen lists the workflow created.

Testing the Workflow

1. Select the listed workflow.
2. Right-click the selected workflow.
3. Click **Publish and Test Run**.
   The **Status Finished successfully** appears.
   If Status is **Execution Pending** in the Process Execution window, then click on the **Refresh** button to get the status.
4. In the **Enterprise Console** menu, select **Inbox**.
   The Inbox lists the work item.
5. Click to view the work item listed.
   This indicates that the installation is successful.

Install Load Balanced Server

The Load Balanced Server supports the following scenarios:

- Web Farm with Clustered Services on Web Farm Machines with Database on a Different Machine
- Web Farm Connecting to Separate Services Cluster with Database on a Different Machine
- Single Server Capable of Adding Machines at Any Time
- Intranet and Extranet Web Servers on Different Machines
- Client and Server Installation

Complete the installation tasks for the scenario that matches your configuration.

Notes:
- For Enterprise Server deployment, it is mandatory to synchronize the date and time in all enterprise servers.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- For optimal performance, we recommend to run all the enterprise servers in the same version of the operating system. For example, ensure to run all the enterprise servers in 64-bit version of operating system.

- The Multicast tool ensures that all machines used for the Enterprise Server installation are able to communicate with each other. For more information about the multicast tool, see Tools Used in Enterprise Server.

Web Farm with Clustered Services on Web Farm Machines with Database on a Different Machine

This section describes the activities involved while installing the Load Balanced Server for the scenario Web Farm with Clustered AVEVA Work Tasks Services on Web Farm Machines with Database on a Different Machine.

In this scenario, AVEVA Work Tasks Server is installed on multiple servers namely Machine 2 (or Server 1), Server 2 and Server N in a Web Farm. "Server N" represents the number of machines in the Web Farm. The Database is installed on Machine 1.

Before installing ensure all Prerequisites are taken care.

See Also:

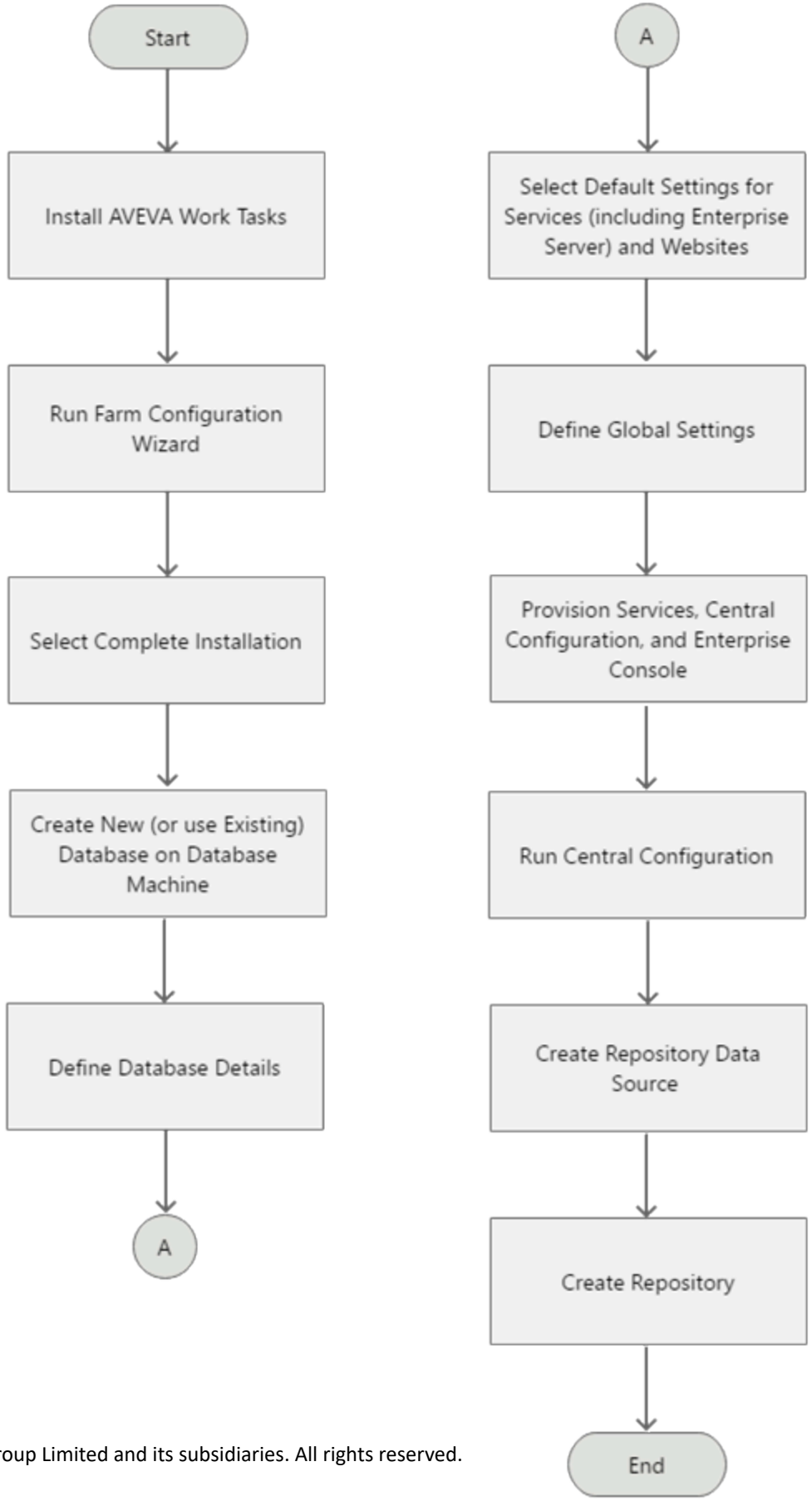
Web Farm with Clustered Services on Web Farm Machines with Database on a Different Machine

High-Level View of Server Installation

This section provides a high-level view of the various installation steps. The objective of this section is to provide an overview of the various activities involved while installing the AVEVA Work Tasks Server. This section also enables you to visually keep track of your activities while following the sequence of tasks.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation

This section provides installation steps for the scenario Web Farm (Multiple Web servers), AVEVA Work Tasks Server, and Database Server on different machines.

In this scenario, the Server is installed on multiple machines in a Web Farm.

AVEVA Work Tasks Server

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
4. Click **Next**.

   The License Server details are configured successfully.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Wizard.

3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.

4. Click **Next**.

   The **Edit Farm Settings** window appears.

5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.

6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.

9. Click **Next**.

   The License Server details are configured successfully.

Select the Type of Installation - Enterprise Edition

Following points must be considered while installing AVEVA Work Tasks on the Enterprise Server:

- Ensure that you select the **Complete Installation** and **Enable load balancing** options.
- You can either create a new Farm database or use an existing one.

1. In the Select the type of installation page, select **Complete Installation (Server)**.

   A Complete Standalone Deployment does not require a client installation.

2. Select the **Enable load balancing (This feature requires and Enterprise license.)** check box for installing AVEVA Work Tasks on the Enterprise Server.

   The **Farm Configuration Wizard** dialog box appears.

3. Click **OK**.
4. To create a new Farm database, click **Create AVEVA Work Tasks Farm**, and then select one of the following:
   - **Create AVEVA Work Tasks Farm database**
   - **Use existing AVEVA Work Tasks Farm database (Creates AVEVA Work Tasks Farm tables only if they do not exist)**
5. To connect to an existing Farm database, click **Connect to existing AVEVA Work Tasks Farm (if the first server of the Enterprise Server is already installed)**.
6. Click **Next**.

   The Database User Settings page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create Farm Database

Use the Create AVEVA Work Tasks Farm Database page to provide **New Database** details. Select the **Connection String** option to establish a connection with the database. This connection information is encrypted and stored.

You can enter either Windows or Microsoft SQL Server user settings for authentication.

Note:
- The Farm database password does not support **&** and **;** special characters.
- While setting up **Farm Database** with **Windows Authentication**, do not enter **FQDN** in the **Domain Name** field.

To Create Farm Database

1. Enter **Database Title**.
2. Select **Connection String**.

   See *Connection Provider* later in this topic, if you have selected the **Connection Provider** option.

   - Enter the following details:

     Server Name:

     Provide information about the database server on which the Farm database can be installed.

     If the **Database Type** is **SQL Server**, then you can specify an IPv6 address for the **Server Name**.

     If SQLExpress Instance need to be used as a Database instance of the Farm database, then **Server Name** should be <server name>\SQLExpress.

     To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

   - **Authentication Type**

     This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports Windows and SQL Server authentication process.

     Example: Windows
     If Windows Authentication is selected, the user configured for the Farm configuration will be used to access the database. This is the authentication recommended by AVEVA Work Tasks.

   - **Login**

     This field is used to enter the SQL login credential for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Password**

     This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Advanced Settings**

     This is an optional field. It is used to enter any additional connection information, such as connection Timeout. Example: Connection Timeout = 20.

     While creating a data source for repository, use the following custom settings to set properties for connection pool. These settings are case sensitive and should be mentioned only in lower case.

     ```
     encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
     ```

     encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

certificate is installed on the SQL Server for SSL encryption.

enableltm: AVEVA Work Tasks internally implements light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive. This property can be used to turn off this feature. The default value is set as True.

connectionretrywaittime: While opening a new connection, if the data base server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

maxconnectionretries: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

3. Click **Test Connection** to ensure database connectivity.
4. Click **Next**.

   The message *AVEVA Work Tasks Farm is Created* appears.
5. Click **OK**.

AlwaysOn Availability Groups feature of Microsoft SQL Server

The **AlwaysOn Availability Groups** feature of Microsoft SQL Server is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in Microsoft SQL Server 2012, AlwaysOn Availability Groups maximizes the availability of a set of user databases for an enterprise.

After you configure the AVEVA Work Tasks, you must add the created Farm Database to the SQL Server AlwaysOn Availability Groups. Contact your Database Administrator to add the database to SQL Server AlwaysOn Availability Groups.

Notes:

- The authentication user needs different rights assigned for different scenarios. If a user wants to create database (Farm or Repository Data Source Database), Authentication User needs to have minimum 'dbcreator' access rights on Database Server Instance.

- If database is already created, user needs to have following permissions on Database Server:

- SELECT, INSERT, UPDATE, and DELETE permissions.

- EXECUTE permission for Stored Procedures.

To choose Connection Provider

Complete the following steps to choose a connection provider:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Enter **Database Title**.
2. In the **Database Type** box, select a database type.
3. Select **Connection Provider**.

   This option is used to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.
4. Enter the following details:
   - **Assembly Path**:

     Click the icon to browse and choose the required dll file.

     If custom properties (enableltm, connectionretrywaittime, maxconnectionretries and sqlcommandtimeout) are required to be used in Assembly connection string, then user should select **Use existing Farm database** while creating a the Farm. See Creating or Using an Existing Database.

     See the topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub chapter in the *Build Business Process* section for creating the assembly. Make sure that, there should be only one class implementing *IDBConnectionString* interface within this assembly.
5. Click **Test Connection** to ensure database connectivity.
6. Click **Next**.
7. The message *AVEVA Work Tasks Farm is created* appears.
8. Click **OK**.

Define Security Settings

You can define the Security Settings using the Farm Configuration Wizard.

For more information on Security settings, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client.

To define the Security Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Security Settings** and **Service Settings** and then, click **Next**.
4. Select **Secure Communication** to establish secured communication between the AVEVA Work Tasks services through HTTPS.
5. Select **Secure Web Sites** check box to configure secured web sites. Uncheck the **Secure Web Sites** check box to use HTTP sites. By default the sites are created in HTTPS mode.
6. Enter the Port number. The default port for the HTTPS site is 8001, and for the HTTP site is 8000. You can enter any port number that IIS accepts. But if any other services or applications are already using the entered port number, AVEVA Work Tasks will not start.
7. Select the available certificate from the **Certificate** list. If your certificate is not listed, click the **Refresh** icon to repopulate the list.

   Note: If the required certificate is not listed, generate one using the System Management Server

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

(Configurator tool). Do not use a self-signed certificate without a proper root chain.

8. Click the **Validate** button to validate the certificate before configuring the security settings.

   Note: An Internet connection is needed to validate the certificate.

9. Click **Configure**, then **Next**, and then **Finish** to apply the new security settings.

Note:
- To update the port number, certificate, and websites from unsecure to secure or vice-versa, select **Security Settings** and **Service Settings** check box in **Edit AVEVA Work Task Farm Settings** window before updating the values. After the required details are provided, these settings get updated only when you click **Configure**, then **Next**, and then **Finish** as you navigate through the Farm Configuration Wizard..
- If the Quickflow is already deployed on a machine, you need to deploy it again after you update the security settings. If secured web site is configured, then you cannot change the server configuration for Quickflow in Central Configuration. However, if the **Secure Web Sites** checkbox is not selected, you can make the changes to the server configuration.
- Advance Server communication cannot be configured on HTTP/HTTPS.
- For resources of Load Balanced Server, ensure to configure all services with the same protocol (HTTP or HTTPS).
- It is recommended to use Standard SSL certificate instead of self-signed SSL certificate.

Using ASB Certificate for Secured Communication

ASB Certificate is a PCS-managed SSL certificate and renews automatically every 30 days when the certificate is still valid. The certificate is valid for two years. The PCS watchdog service on the runtime node checks the validity of the SSL certificate within 5 minutes from when it starts and once a day. PCS also renews the SSL certificate if it is expired.

If the runtime node is offline and later back online, the PCS watchdog service regularly checks the certificate validity and renews it if it is required. Consider restarting the PCS watchdog service to enforce the SSL certificate validity check within 5 minutes from the restarting time.

Furthermore, when the SSL certificate is renewed, PCS binds it to port 443 (by default) and changes the SslThumbprint PCS registry value. If AVEVA Work Tasks sites are hosted on port 443, certificate binding will be auto-renewed when the SSL certificate is renewed. It is recommended to check the bindings in IIS with every certificate renewal to ensure the application is running seamlessly.

To disable the certificate renewal, consider disabling the **ArchestrA Certificate Renewal Service** windows service.

Note: Client Service will try to automatically bind the renewed certificate. If there are any issues in browsing the website or starting the services after certificate renewal, users need to reconfigure the secure communication in Farm Configuration. To reconfigure, see the steps mentioned in above section *To define the Security Settings*.

Securing Communication between Services

To have a secure communication between the services, apply the below changes to the Windows registry and reboot the server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
```

For more information on securing communication, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client and https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2015/2960358.

Define Services and Global Settings

You can define the Services and Global Settings using the Farm Configuration Wizard or through Configuration XMLs.

To define Services

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.

   Note: You can also customize the settings for these services and websites using the **Custom Settings** option. For more information about Service-related changes and Website-related changes, see Advanced Settings and Administering Web Servers respectively.

2. Click **Next**.

   The Global Settings dialog box appears.

3. In the **Communication** tab, enter key fields in **Email Settings** and **Exception Email settings** to configure exception e-mails.

   The following are the details:

   - **Host**: Enter information related to mail server.
     Example: Enter details related to Microsoft Exchange Server (IP address of the exchange server).
   - **Port**: Enter the port number for the mail server. The default port number is 25.
   - **From email**: Enter the from e-mail address.
     Example: admin@telematics.com
   - **To email**: Enter the to e-mail address.
     Example: alex@telematics.com

4. Click **OK**.

Note: To confirm the exception e-mail settings, click **Test SMTP**. An e-mail will be generated and sent to the **To email** address confirming successful configuration of the e-mail. The advanced settings like **Secure Socket Layer (SSL), Email Timeout, User Name,** and **Password** for the e-mail account holder, **Password Encrypted** flag will differ based on the mail server settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Define Central Configuration

1. Run **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Farm Configuration Wizard window appears.
2. Click **Next**.
3. The Edit Farm settings window appears.
4. Select **Service Settings**.
5. Click **Next**.

   The Services/Global Settings window appears.
6. Click against **Central Configuration** to edit AVEVA Work Tasks **Central Configuration settings**.

   The Central Configuration Settings page appears.
   - **http/https**: This option enables you to determine if you want a secured (https) or a non-secured site (http).
   - **Authentication Type**: This option allows you to provide the authentication type to access the Central Configuration. The available authentication types are Windows, AVEVA Connect, and Azure Active Directory.

     Note: The AVEVA Connect and Azure Active Directory authentication types are available only if these user providers are configured in the system.
     To configure AVEVA Connect user provider, see AVEVA Connect Settings.
     To configure Azure Active Directory user provider see Azure Active Directory Settings.
   - **Create at**: This field allows you to provide the Internet Information Service (IIS) path where the new site is created.
   - **Port Number**: This field is used to provide a port number for the Central Configuration.
7. Click against **Create at** to change the path of the Central Configuration.

   A message indicating that if the path has been changed the Central Configuration will no longer be accessible appears.
8. Click **Yes**.

   The IIS structure screen is displayed. Select the desired IIS path for the Central Configuration.
9. Click **OK**.

   The selected URL appears in the **Central Configuration Settings** screen.
10. Click **OK**.

Note:

- The default IIS path for Central Configuration is displayed. Click the edit icon to change the site path.

- If you want to update the authentication type to AVEVA Connect or Azure Active Directory of the already configured Central Configuration, you need to reconfigure the AVEVA Connect and Azure Active Directory user providers for the Central Configuration. For information on configuring user providers, see AVEVA Connect Settings and Azure Active Directory Settings.

- Central Configuration cannot be accessed in Chrome browser when configured with unsecured access (http://)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

for Azure Active Directory and AVEVA Connect authentication types.

Define Enterprise Console

1. Run **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Farm Configuration Wizard page appears.
2. Click **Next**.

   The Edit AVEVA Work Tasks Farm settings page appears.
3. Select **Service Settings**.
4. Click **Next**.

   The Services/Global Settings page appears.
5. Click the **Advanced Settings for Enterprise Console** button.

   The Enterprise Console Settings page appears.

   Note: If Enterprise Console is running in secured mode (https), update the **Intranet Path** field with the associated fully qualified domain name in place of the server name.
   Example: If the fully qualified domain name is **mycert.mydomain.com**, then provide *https://mycert.mydomain.com/Enterpriseconsole* in the **Intranet Path** field.
6. Click **Save**.

   The created Enterprise Console is displayed in the **Existing Enterprise Consoles** grid.
7. Click **OK**.

   The Installing Service screen appears, indicating the progress of installation. The message *Installation is Complete* appears after all the services are installed.
8. Click **OK**.

Accessing Enterprise Console Settings in AVEVA Work Tasks Pro App App

You can configure the Enterprise Console settings such as Enterprise Console path, Repository name, corresponding Provider name, and User name in the Farm Configuration Wizard. You can generate a QR code to store these settings. This QR code can then be scanned by the AVEVA Work Tasks Pro App to automatically fill the settings in the Settings screen.

1. Run **Farm Configuration Wizard**.

   The Welcome to the **AVEVA Work Tasks** Farm Configuration Wizard page appears.
2. Click **Next**.

   The Edit **AVEVA Work Tasks** Farm settings page appears.
3. Select **Service Settings**.
4. Click **Next**.

   The Services/Global Settings page appears.
5. Click the **Advanced Settings for Enterprise Console** button.

   The Enterprise Console Settings page appears.
6. Click **Generate QR Code**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Scan QR Code** dialog box appears.

7. Click **Download**.

   The **Save As** dialog box appears.
8. Enter the **File name** for the QR code and select the **Save as type** as **PNG**.
9. Click **Save**.

   The QR Code is generated successfully.
10. Scan this QR Code in the AVEVA Work Tasks Pro App.

    The values configured for the following settings in the Farm Configuration wizard are considered by default.
    - Enterprise Console path. The local intranet Enterprise Console path is considered.
    - Repository name
    - Provider name
    - User name
11. Enter the password.

Provision Services

While the installation is in progress, you can view the services that will be installed. This provides information on the progress of installation. At the end of the installation, a status indicating the completion of installation appears.

- Click **OK** to complete the installation.

Create a Repository DataSource

A **Repository DataSource** is used to store all repository related data. The **Repository DataSource** must be created on a machine where the **Central Configuration** site is installed.

Example

A TelematicsDataSource is created for TelematicsHR Repository.

A Repository contains Workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating the repository.

To create a Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**.

   The Central Configuration web page appears.
2. Click **DataSource Settings**.

   The Datasource List page appears.
3. Click **Add**.

   The Manage Datasource page appears.
4. Enter the required details.
   a. **Name**: Enter a valid data source name. Example: *TelematicsDataSource*.
   b. **Description**: Enter a description for the data source. Example: *This data source is used to store*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

connection details.

c. **Connection String**: Select this option to establish a connection with the database. The connection string contains fields that that defines the connection. The connection information is encrypted and stored.

Server Name

To use a database on the local machine, enter "localhost", ".", or the local computer's name for the **Server Name**.

Note: Server Name can be machine name/IP address/SQL Server AlwaysON/SQL Server instance name. Instead of entering "local host", it is preferable to give the local machine name.

If the **Database Type** is **SQL Server**, then specify an IPV6/IPV4 address for the **Server Name**.
To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: *localhost,1455*
To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be *<server name>\SQLExpress*.

Authentication Type

Use this field to determine the process to validate a user who is trying to connect to the database. AVEVA Work Tasks uses Windows and SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. The **User ID** and **Password** field is disabled if Windows Authentication is selected. Example: *Windows*

Advanced Settings

Use this field to define advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection.
Example: *Connection Timeout = 20*
or
For defining maximum connections that can be opened in a connection pool.
Example: *max pool size = 250*
Default connection pool is set as 250. We recommend to enter a higher value on heavily loaded environment.
While creating a data source for a repository, use the following custom settings to set properties for the connection pool.
encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.
enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature, AVEVA Work Tasks controls the transaction from escalating to an expensive distributed transaction.
The above property can be used to turn off this feature. The default value is set as *True*.
connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is 30 seconds. The value should be set in seconds.
maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is infinite.
sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

generating an error. The default value is set as 300.
These settings are case sensitive and should be mentioned only in lower case.

Database:

Specify or select a database that can be used as a repository database. This database stores all repository related data. The repository database is created if it does not exist on the database server instance.
Example: *TelematicsDB*

Enable Table Partitioning:

Select to enable database table partitioning. For more information, see Partitioning Database.

d. **Connection Provider**: Use this to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.

The selected dll must be copied to all the machines, all servers and all clients, that are connected to the current Farm.

Assembly Path

Click to browse and choose the required dll file.
To use custom properties (enableltm, connectionretrywaittime, maxconnectionretries, and sqlcommandtimeout) in the Assembly connection string, the database must be created before.
See *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Developer Guide* for creating the assembly. Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.

   A message about successful test connection appears.

6. Click **OK**.

   A message about successful creation of the data source appears.The message also states that the required tables will be created in the data source after a repository is associated with the data source.

When a data source is saved, an entry is created for the data source in the **Farm Database**. When a first repository is created for the data source, AVEVA Work Tasks tables are added to the data source. While creating a data source, the archival database is created with the same database name suffixed by *_BI*. This database is used for archival in future use.

Note: After you configure the AVEVA Work Tasks, you must add the created **Farm Database** to the **SQL Server AlwaysOn Availability Groups**. Contact your Database Administrator to add the database to **SQL Server AlwaysOn Availability Groups**.

To view the data source created, click **DataSource Settings** link.

Create a Repository

A **Repository** contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

To create a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The Central Configuration page appears.

2. Click **Repository Settings**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Repository List page appears.

3. Click **Add**.

   Note: If the number of repositories exceeds the permitted limit for the license, then the message "You have exceeded the maximum number of repositories" appears, and the repository creation is stopped. Click **Close**.

   The **Manage Repository** page appears.
4. Enter the required details.
5. Click **OK**.

   The **Config Administrator** page appears.
6. Click to configure an administrator.

   The User Lookup dialog box appears.
7. Select the provider from the provider drop-down list.
8. Enter the details to search the administrator for the repository that is created.

   Note: You can also search a user based on *ObjectSID* and *objectGUID* (applicable only for Active Directory provider).
9. Click **Search**.

   The user is listed under **Available Resources**.
10. Select and click the arrow to list the user under **Selected Resource**.
11. Click **Update**.

    The selected user is displayed in **User Lookup**.
12. Click **OK**.

    A message confirming the successful creation of an administrator appears.
13. Click **OK**.

    The **Repository List** page appears.

    For more information about configuring the system user, see Configure System User.

Manage Repository Window Fields

The following table explains the fields of the Manage Repository window:

Field Name	Description	Example
Repository DataSource	Lists all the data sources created in a Farm. The selected data source will be used to store data for the current repository.	TelematicsDatasource
Attach DataSource	Select Attach DataSource checkbox and then select a repository from the selected DataSource .Data associated with the selected repository will be copied to the new repository.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	The version of the installed product and destination DataSource should be the same as that of the DataSource from which the repository data is to be copied. For example, in an upgraded environment, from build Work Tasks version 1 to Work Tasks version 2, you cannot attach a DataSource created in build Work Tasks version 1 to build Work Tasks version 2. To match the build version of the installed product to the version of the DataSource: 1. If the DataSource was created in build Work Tasks version 1, then the entire build Work Tasks version 1 must be upgraded to Work Tasks version 2 by running the product installer in the upgrade mode. 2. All the Farm, DataSources, Repositories, and Lists of build Work Tasks version 1 must be upgraded to build Work Tasks version 2 by running the AVEVAWorkTasksUpgradeManager.exe located at the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder. 3. After upgrading all the AVEVA Work Tasks components to build Work Tasks version 2 build, that DataSource can be used in any machine that has same version by using this feature.	
Repository Name	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the	Telematics Repository

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	repository name.	
Description	Provide a description for the repository.	This stores Telematics Employee Leave Details.
From mail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might occur in the Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be sent to the mail ID defined in **Exception to mail**. The From address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To avoid this, restart the Workflow Engine Service and Communication Service.	Provide the valid email id that is specified in the **User Name** field while configuring **SMTP Settings** in the **Mail Server Settings** page. For more details, see Mail Server Setting.
Exception to mail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails is sent to the email ID specified in **Exception to mail**.	Provide the valid email id.
Alert initiator of the Workflow	Determines if the status of a workflow should be sent as a notification mail to the initiator of the workflow.	A process with a rule to check the pending leave available for the employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
Servers	Lists the servers available in the Farm. Repository will be available only to the mapped servers.	Work Tasks Server Name
Enterprise Consoles	Lists all the Enterprise Consoles for	http://worktaskservername:8001/

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	a selected AVEVA Work Tasks Server. Repository will be available only to the mapped Enterprise Console.	EnterpriseConsole
Enable Theme	Select this checkbox to enable the defined theme.	-
Theme Name	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. It provides **Default** and **Lite** themes.	-
Resource Provider	Lists the various default resource providers supported by AVEVA Work Tasks. Resource providers are also termed as User Provider. A User Provider is a data store containing information about the user. AVEVA Work Tasks also supports custom resource provider. For more information about Resource Provider, see User Providers. AVEVA Connect and Azure Active Directory resource providers are also available after you configure them. To configure these providers, see AVEVA Connect Settings and Azure Active Directory Settings. AVEVA Work Tasks allows secured LDAP configuration. The default port number for LDAPS is 636. You can map the repository in IDE and use the Secured LDAP. The logged-in Active Directory is displayed by default, and **Active Directory Datasource** expects **LDAP://** or **Fully Qualified Domain Name** as value.	AVEVA Work Tasks Lists or Active Directory
Custom Application Folder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	a semicolon.	
Hide Repository	Select this option to make the repository unavailable to the Farm.	-
Source path	Enter the path to the Source Elements Folder.	-
Attachment Storage Path	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an absolute path. - The default path is the **Attachments** folder in *C:\ProgramData\AVEVA Work Tasks*\. - The default path is used when no path is specified. - Ensure appropriate rights are set to the folder, where attachments are stored, for the following identities: -- The application pool identity for **Central Configuration** and **Enterprise Console(s)**. -- The account configured for the **Workflow Engine** service. - In case of mapped drives, ensure they are available to the identities. - In multiple-server environments, the path specified must be in a shared location.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Auto Log Off Settings for Enterprise Console	Automatically logs out the logged in user after the designated period of inactivity. Enter the time in minutes. This field accepts only valid integer values between 0 to 360. 0 minute means the user will not get automatically logged off. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. The default value for automatic log off is 30 minutes. You use this field to define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML.	0 to 360 (minutes)
Auto Log Off Settings for AVEVA Work Tasks Pro	Request for user credentials for AVEVA Work Tasks Pro app after the set time is elapsed. Enter the time in minutes. This field accepts only valid integer values between 0 to 2880. 0 minute means the user will not get automatically logged off. The default value is 30 minutes. If AVEVA Work Tasks Pro app is offline, the Auto Log Off will not get active, and the user will not be automatically logged off as per the defined settings.	0 to 2880 (in minutes)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Offline Support	Select this option to enable offline submission of items in AVEVA Work Tasks Pro app.	-
Refresh on Work Item submission	Select this option to enable Inbox refresh on Work Item submission in AVEVA Work Tasks Pro . Refresh option is enabled by default.	-
Timeout	Enter the time in seconds to set the timeout. If the value is set as 0, then there is no timeout. Default value is 60 seconds.	-
Label for Fill Form	Enter the label for Fill Form. The customized label will be displayed for the Fill Form in AVEVA Work Tasks Pro.	Work Tasks
Date Configuration Flag for Alternate Actors	Select the **Enable Date Settings** option to set the **From** date and **To** date as mandatory while creating alternate actors. For more information about creating alternate actors, see *Alternate Actors* topic in the *User Guide*.	-
Display Domain name with User Name	Select this checkbox to display the domain name along with the user name in the Enterprise Console and the List. The domain name will be displayed only: • If this checkbox is selected. • If the **Resource Provider** is selected as **Active Directory**.	-

User Providers

AVEVA Work Tasks supports the following User Providers by default:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Active Directory

When you select **Active Directory** as the **Resource Provider**, details such as **Active Directory datasource** and the **Active Directory search base** are populated by default.

Authentication User, **Authentication Password**, **User Log-on ID**, and **User Log-on Display Name** are optional fields. Access to the active directory might not be available for all users. In such cases, provide the active directory authentication user name and the active directory authentication password with which you can connect to the Active Directory.

AVEVA Work Tasks considers *SAM-Account-Name* as **User Log-on ID** by default. If you want to use active directory attributes other than the *SAM-Account-Name* user, then enter the desired active directory attribute details in the **User Log-on ID** field and **User Log-on Display Name** field. **User Log-on Display Name** is the user friendly display name of the active directory attribute. This **User Log-on Display Name** is referred in the User Lookup dialog box against the **User Log-on ID**.

Steps

1. When you select **Active Directory** as the **Resource Provider**, then the **Use Token Service** field appears. In this field, select one of the following:
   - If you do not want to use AIM authentication, select **Without Token Service**.
   - If you want to use AIM authentication, then select a server from the list of alias server names.
2. Enter **Active Directory datasource**.
3. Enter **Active Directory search base**.
4. Enter **Domain Name**.

   The domain name is case-sensitive.
5. Enter **Authentication User**.
6. Enter **Authentication Password**.
7. Check **Is Cross Domain Role Provider** checkbox, if cross domain users and roles mapping is required.

   Example: Users *John* and *Jim* of domain *TelematicsU* and users *Mike* and *Bob* from domain *TelematicsAU* are added to role Team HR of domain *TelematicsMain*.
8. Enter **User Log-on ID** if you want to use another Active Directory attributes instead of *SAM-Account-Name*.

   We recommend to use attributes of type *string* as input for **User Log-on ID.** However, *ObjectSID* and *objectGUID* are also supported.

   After you set the **User Log-on ID**, you cannot change it later.
9. Enter the **User Log-on Display Name**.

   After you set the **User Log-on Display Name**, you cannot change it later.

   This is the default display name of the user account of the active directory user used. **User Log-on Display Name** is used in user lookups.
10. Click **Test LDAP**.

    A message about successful LDAP test appears.

Note:
- Some of the active directory attributes or properties are not indexed. If you use an active directory attribute that is not indexed, it might impact the log-on validation performance based on the active directory entries. We recommend to use indexed attributes as **User Log-on ID** or index the attributes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- For an upgraded repository, you cannot change both **User Log-on ID** and **User Log-on Display Name**.
- You cannot localize **User Log-on ID** and **User log-on Display Name** fields.
- Disabled users will not be shown in user lookup search.

Using Forms

For Forms Resource Provider:

- If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.

1. In the **Resource Provider** drop-down list**,** select **Forms.**
2. In the **User Entity properties** field, enter XML data.

   Click **Sample Code** to get the XML Schema Template.

   Example

   ```
   <properties>
   <property name="userid" type="string" purpose="identifier" />
   <property name="username" type="string" purpose="name" />
   <property name="pwd" type="string" purpose="password" />
   </properties>
   ```
3. In the **User Provider Properties** field, enter the same XML Schema.

   There must be at least one identifier in the XML Schema. For example, UserID
4. In the **Datasource** field, enter a valid data source or specify a dll.

   The value for the data source can be specified as **Physical path of the dll** or **Physical path of the dll::param. Param** is not mandatory.

   See topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Build Business Process* section for creating the assembly.

   Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly, whereas one class can return multiple connection strings.

   Examples of data source parameter:

   - c:\DLLStore\GetConnectionString.dll
   - c:\GetConnectionString.dll::Leave

   If there are two data sources, *Leave* and *Licensing*, and only one DLL is used to return a connection string based on the data source selected. The parameter can be specified as **c:\GetConnectionString.dll::Leave** or **c:\ GetConnectionString.dll::Licensing**. The connection string can be returned from the connection dll based on this parameter.
5. Enter a proper table name or view name which holds the User Name and Password for the Authentication purpose.

   Example: Passmast (the table which is available in IMS_NEW Database)
6. Click **Test SQL Connection**.

   A message about successful test connection appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using ASP.NET Membership

This resource provider lists users defined in the asp.net membership database.

Lists

This resource provider lists users created using AVEVA Work Tasks Users Creation. If you are unable to determine a User Provider to create a repository, use AVEVA Work Tasks Lists.

To verify successful creation of the Repository:

1. Click **Enterprise Console** URL from the site path column.
   The repository created is listed.
   Add the created Enterprise Console link into "Add to Favorites" to easily navigate in future.
2. Click **Repository User Provider** link to logon to the repository.
   The Enterprise Console log-on page appears.
3. Enter the logon details, and then click **Login**.
   The Enterprise Console home page appears. This indicates successful creation of a Repository.
   All features can only be available for a user with administrative privileges.

Note: Consider you created the Repository and then deleted it. If you create the same Repository again immediately and try to log into it through Enterprise Console, the Provider does not list the Repository. You need to access IIS and recycle AVEVA Work Tasks from the Application Pool to get the Repository listed in the Provider list..

Install on Multiple Machines

Repeat the same installation steps for other servers.

- Ensure that the same Farm database should be selected when configuring AVEVA Work Tasks Farm on other servers.
- Map the same repository for all the machines and the same repository can be accessed on all machines.
- After AVEVA Work Tasks installation on the second machine, go to *Central Config* site and map the current machine URLs for the existing repository.

Verify the Installation

After you install AVEVA Work Tasks, verify that the product is installed successfully. To verify successful installation, create a simple Workflow using Process Designer. The tasks includes the following:

- Creating a Workflow
- Publishing the Workflow
- Testing the Workflow

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Creating a Workflow

1. Log on to the **Enterprise Console**.
2. If the Repository is configured for Active Directory provider, go to step 4.
3. Enter the **User Name** and **Password**.
   The **User Name** and **Password** will be validated with the user provider selected while creating the repository.
   The default authentication provider is the Repository User Provider List.
4. In the **Enterprise Console**, click menu icon.
   All features can only be available for a user with administrative privileges.
5. From the menu, select **Workflows**.
   The Workflow window appears.
6. On the Action bar, click **New**, and then click **Workflow**.
   The New Workflow window appears.
7. Enter **Title** and **Description**.
8. Click **Save & Continue**.
   The successful message appears.
9. Select **Design the Workflow now**.
   This is the default option.
10. Click **Finish**.
    The Process Designer window appears.
11. From the activities pane on the left, drag the Information activity to the Process Designer window.
12. The activity *Information1* is created.
    Click to create a link between two activities.
    The Link Action window appears.
    a. Select from the **From Action** drop-down box.
    b. Select from the **To Action** drop-down box.
    c. Click **OK**.
       The link is established.
13. Right-click *Information1*.
14. Select **Activity Properties** from the context menu.
    The Activity Properties pane appears.
    a. Enter **Name** and **Description**.
    b. Click to **Assign Actor(s)**.
       The Assign Actor(s) window appears.
    c. Select the required provider from the **Providers** drop-down list.
    d. Select values from the drop-down list.
    e. Click **Add**.
    f. Click **Update**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

15. Click to enter the **Subject** for Notification Contents.
    The window to enter subject for notification appears.
16. Click **Update**.
17. Click to enter the **Body** for Notification Contents.
    The window to enter the Body content appears.
18. Click **Update**.
19. From the Activity Properties pane, click **Save**.

Publishing the Workflow

1. On the **Publish** menu, click **Publish**.
   The Published successful message appears.
2. Close the dialog box.
   The Enterprise Console screen lists the workflow created.

Testing the Workflow

1. Select the listed workflow.
2. Right-click the selected workflow.
3. Click **Publish and Test Run**.
   The **Status Finished successfully** appears.
   If Status is **Execution Pending** in the Process Execution window, then click on the **Refresh** button to get the status.
4. In the **Enterprise Console** menu, select **Inbox**.
   The Inbox lists the work item.
5. Click to view the work item listed.
   This indicates that the installation is successful.

Web Farm Connecting to Separate Services Cluster with Database on a Different Machine

This section describes the activities involved while installing the Load Balanced Server for the scenario Web Farm Connecting to Separate AVEVA Work Tasks Services Cluster with Database on a Different Machine.

In this scenario, AVEVA Work Tasks Server is installed in a cluster. The cluster includes Machine 1, Server 1, and Server N. "N" represents the number of machines in AVEVA Work Tasks Cluster.

AVEVA Work Tasks Client is installed in a Web Farm. The Web Farm contains Machine 3 (or Server 1), Server 2 and Server N. "N" represents the number of machines in the Web Farm.

Ensure that you install the product in the following sequence:

- High-Level View of Server Installation
- High-Level View of Client Installation

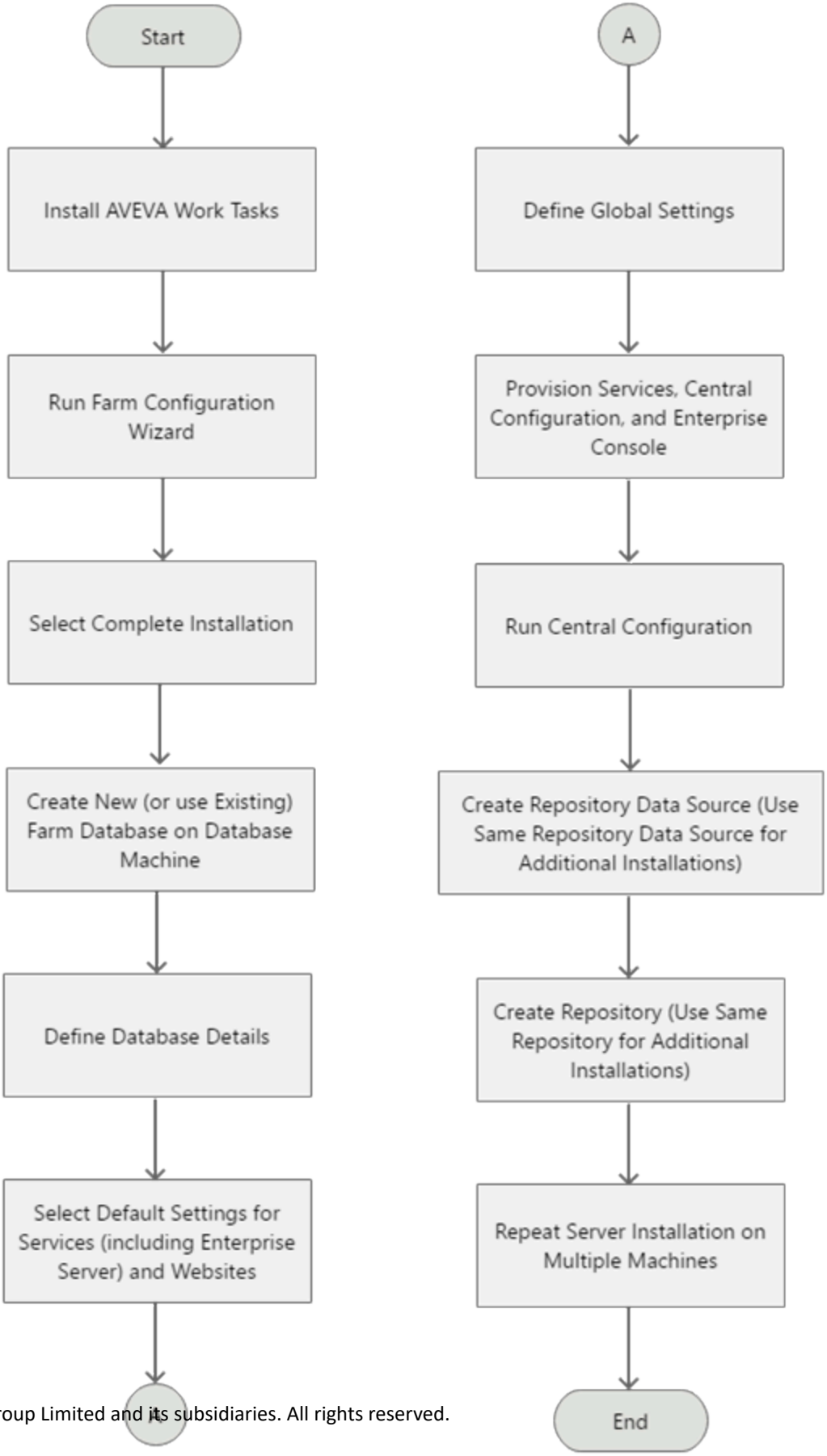
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

High-Level View of Server Installation

This section provides a high-level view of the various installation steps. The objective of this section is to provide an overview of the various activities involved while installing the AVEVA Work Tasks Server. This section also enables you to visually keep track of your activities while following the sequence of tasks.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



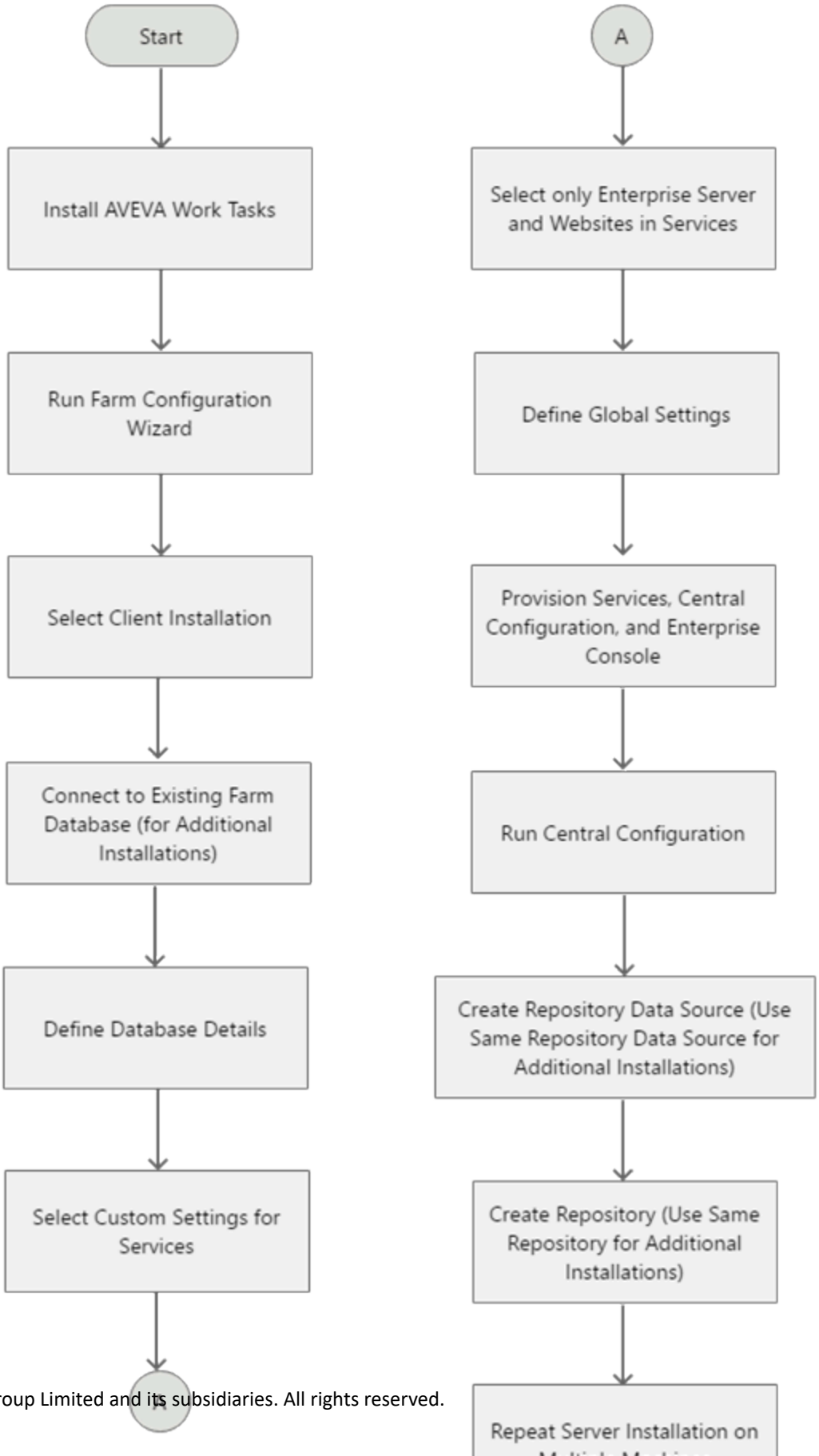
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

High-Level View of Client Installation

This section provides a high-level view of the AVEVA Work Tasks Client installation. The Client components are installed on Web Servers. For more information on this scenario, see Load Balanced Server.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation

This section provides installation steps for the scenario Web Farm Connecting to Separate AVEVA Work Tasks Services Cluster with Database on a Different Machine.

The installation includes the following:

- Installing AVEVA Work Tasks Server
- Installing AVEVA Work Tasks Client

The Server has to be installed on all the machines in a Cluster before installing the Client in a Web Farm.

AVEVA Work Tasks Server

This section describes the activities involved while installing AVEVA Work Tasks on a server machine.

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.

2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.

3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.

5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.

6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.

4. Click **Next**.

   The License Server details are configured successfully.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.
4. Click **Next**.

   The **Edit Farm Settings** window appears.
5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.
6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
9. Click **Next**.

   The License Server details are configured successfully.

Select the Type of Installation - Enterprise Edition

Following points must be considered while installing AVEVA Work Tasks on the Enterprise Server:

- Ensure that you select the **Complete Installation** and **Enable load balancing** options.
- You can either create a new Farm database or use an existing one.

1. In the Select the type of installation page, select **Complete Installation (Server)**.

   A Complete Standalone Deployment does not require a client installation.
2. Select the **Enable load balancing (This feature requires and Enterprise license.)** check box for installing AVEVA Work Tasks on the Enterprise Server.

   The **Farm Configuration Wizard** dialog box appears.
3. Click **OK**.
4. To create a new Farm database, click **Create AVEVA Work Tasks Farm**, and then select one of the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Create AVEVA Work Tasks Farm database**
- **Use existing AVEVA Work Tasks Farm database (Creates AVEVA Work Tasks Farm tables only if they do not exist)**

5. To connect to an existing Farm database, click **Connect to existing AVEVA Work Tasks Farm (if the first server of the Enterprise Server is already installed)**.
6. Click **Next**.

   The Database User Settings page appears.

Create Farm Database

Use the Create AVEVA Work Tasks Farm Database page to provide **New Database** details. Select the **Connection String** option to establish a connection with the database. This connection information is encrypted and stored.

You can enter either Windows or Microsoft SQL Server user settings for authentication.

Note:
- The Farm database password does not support **&** and **;** special characters.
- While setting up **Farm Database** with **Windows Authentication**, do not enter **FQDN** in the **Domain Name** field.

To Create Farm Database

1. Enter **Database Title**.
2. Select **Connection String**.

   See *Connection Provider* later in this topic, if you have selected the **Connection Provider** option.

   - Enter the following details:

     Server Name:

     Provide information about the database server on which the Farm database can be installed.

     If the **Database Type** is **SQL Server**, then you can specify an IPv6 address for the **Server Name**.

     If SQLExpress Instance need to be used as a Database instance of the Farm database, then **Server Name** should be <server name>\SQLExpress.

     To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

   - **Authentication Type**

     This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports Windows and SQL Server authentication process.

     Example: Windows
     If Windows Authentication is selected, the user configured for the Farm configuration will be used to access the database. This is the authentication recommended by AVEVA Work Tasks.

   - **Login**

     This field is used to enter the SQL login credential for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Password**

     This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Advanced Settings**

  This is an optional field. It is used to enter any additional connection information, such as connection Timeout. Example: Connection Timeout = 20.

  While creating a data source for repository, use the following custom settings to set properties for connection pool. These settings are case sensitive and should be mentioned only in lower case.

  ```
  encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
  ```

  encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

  enableltm: AVEVA Work Tasks internally implements light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive. This property can be used to turn off this feature. The default value is set as True.

  connectionretrywaittime: While opening a new connection, if the data base server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

  maxconnectionretries: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

  sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

3. Click **Test Connection** to ensure database connectivity.
4. Click **Next**.

   The message *AVEVA Work Tasks Farm is Created* appears.
5. Click **OK**.

AlwaysOn Availability Groups feature of Microsoft SQL Server

The **AlwaysOn Availability Groups** feature of Microsoft SQL Server is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in Microsoft SQL Server 2012, AlwaysOn Availability Groups maximizes the availability of a set of user databases for an enterprise.

After you configure the AVEVA Work Tasks, you must add the created Farm Database to the SQL Server AlwaysOn Availability Groups. Contact your Database Administrator to add the database to SQL Server AlwaysOn Availability Groups.

Notes:

- The authentication user needs different rights assigned for different scenarios. If a user wants to create database (Farm or Repository Data Source Database), Authentication User needs to have minimum 'dbcreator' access rights on Database Server Instance.

- If database is already created, user needs to have following permissions on Database Server:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- SELECT, INSERT, UPDATE, and DELETE permissions.

- EXECUTE permission for Stored Procedures.

To choose Connection Provider

Complete the following steps to choose a connection provider:

1. Enter **Database Title**.
2. In the **Database Type** box, select a database type.
3. Select **Connection Provider**.

   This option is used to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.
4. Enter the following details:
   - **Assembly Path**:

     Click the icon to browse and choose the required dll file.

     If custom properties (enableltm, connectionretrywaittime, maxconnectionretries and sqlcommandtimeout) are required to be used in Assembly connection string, then user should select **Use existing Farm database** while creating a the Farm. See Creating or Using an Existing Database.

     See the topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub chapter in the *Build Business Process* section for creating the assembly. Make sure that, there should be only one class implementing *IDBConnectionString* interface within this assembly.
5. Click **Test Connection** to ensure database connectivity.
6. Click **Next**.
7. The message *AVEVA Work Tasks Farm is created* appears.
8. Click **OK**.

Define Security Settings

You can define the Security Settings using the Farm Configuration Wizard.

For more information on Security settings, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client.

To define the Security Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Security Settings** and **Service Settings** and then, click **Next**.
4. Select **Secure Communication** to establish secured communication between the AVEVA Work Tasks services through HTTPS.
5. Select **Secure Web Sites** check box to configure secured web sites. Uncheck the **Secure Web Sites** check box

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

to use HTTP sites. By default the sites are created in HTTPS mode.

6. Enter the Port number. The default port for the HTTPS site is 8001, and for the HTTP site is 8000. You can enter any port number that IIS accepts. But if any other services or applications are already using the entered port number, AVEVA Work Tasks will not start.
7. Select the available certificate from the **Certificate** list. If your certificate is not listed, click the **Refresh** icon to repopulate the list.

   Note: If the required certificate is not listed, generate one using the System Management Server (Configurator tool). Do not use a self-signed certificate without a proper root chain.
8. Click the **Validate** button to validate the certificate before configuring the security settings.

   Note: An Internet connection is needed to validate the certificate.
9. Click **Configure**, then **Next**, and then **Finish** to apply the new security settings.

Note:
- To update the port number, certificate, and websites from unsecure to secure or vice-versa, select **Security Settings** and **Service Settings** check box in **Edit AVEVA Work Task Farm Settings** window before updating the values. After the required details are provided, these settings get updated only when you click **Configure**, then **Next**, and then **Finish** as you navigate through the Farm Configuration Wizard..
- If the Quickflow is already deployed on a machine, you need to deploy it again after you update the security settings. If secured web site is configured, then you cannot change the server configuration for Quickflow in Central Configuration. However, if the **Secure Web Sites** checkbox is not selected, you can make the changes to the server configuration.
- Advance Server communication cannot be configured on HTTP/HTTPS.
- For resources of Load Balanced Server, ensure to configure all services with the same protocol (HTTP or HTTPS).
- It is recommended to use Standard SSL certificate instead of self-signed SSL certificate.

Using ASB Certificate for Secured Communication

ASB Certificate is a PCS-managed SSL certificate and renews automatically every 30 days when the certificate is still valid. The certificate is valid for two years. The PCS watchdog service on the runtime node checks the validity of the SSL certificate within 5 minutes from when it starts and once a day. PCS also renews the SSL certificate if it is expired.

If the runtime node is offline and later back online, the PCS watchdog service regularly checks the certificate validity and renews it if it is required. Consider restarting the PCS watchdog service to enforce the SSL certificate validity check within 5 minutes from the restarting time.

Furthermore, when the SSL certificate is renewed, PCS binds it to port 443 (by default) and changes the SslThumbprint PCS registry value. If AVEVA Work Tasks sites are hosted on port 443, certificate binding will be auto-renewed when the SSL certificate is renewed. It is recommended to check the bindings in IIS with every certificate renewal to ensure the application is running seamlessly.

To disable the certificate renewal, consider disabling the **ArchestrA Certificate Renewal Service** windows service.

Note: Client Service will try to automatically bind the renewed certificate. If there are any issues in browsing the website or starting the services after certificate renewal, users need to reconfigure the secure communication in Farm Configuration. To reconfigure, see the steps mentioned in above section *To define the Security Settings*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Securing Communication between Services

To have a secure communication between the services, apply the below changes to the Windows registry and reboot the server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
```

For more information on securing communication, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client and https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2015/2960358.

Define Services and Global Settings

You can define the Services and Global Settings using the Farm Configuration Wizard or through Configuration XMLs.

To define Services

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.

   Note: You can also customize the settings for these services and websites using the **Custom Settings** option. For more information about Service-related changes and Website-related changes, see Advanced Settings and Administering Web Servers respectively.

2. Click **Next**.

   The Global Settings dialog box appears.

3. In the **Communication** tab, enter key fields in **Email Settings** and **Exception Email settings** to configure exception e-mails.

   The following are the details:

   - **Host**: Enter information related to mail server.
     Example: Enter details related to Microsoft Exchange Server (IP address of the exchange server).
   - **Port**: Enter the port number for the mail server. The default port number is 25.
   - **From email**: Enter the from e-mail address.
     Example: admin@telematics.com

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **To email**: Enter the to e-mail address.
  Example: alex@telematics.com

4. Click **OK**.

Note: To confirm the exception e-mail settings, click **Test SMTP**. An e-mail will be generated and sent to the **To email** address confirming successful configuration of the e-mail. The advanced settings like **Secure Socket Layer (SSL), Email Timeout, User Name,** and **Password** for the e-mail account holder, **Password Encrypted** flag will differ based on the mail server settings.

Define Central Configuration

1. Run **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Farm Configuration Wizard window appears.
2. Click **Next**.
3. The Edit Farm settings window appears.
4. Select **Service Settings**.
5. Click **Next**.

   The Services/Global Settings window appears.
6. Click against **Central Configuration** to edit AVEVA Work Tasks **Central Configuration settings**.

   The Central Configuration Settings page appears.
   - **http/https**: This option enables you to determine if you want a secured (https) or a non-secured site (http).
   - **Authentication Type**: This option allows you to provide the authentication type to access the Central Configuration. The available authentication types are Windows, AVEVA Connect, and Azure Active Directory.

     Note: The AVEVA Connect and Azure Active Directory authentication types are available only if these user providers are configured in the system.
     To configure AVEVA Connect user provider, see AVEVA Connect Settings.
     To configure Azure Active Directory user provider see Azure Active Directory Settings.

   - **Create at**: This field allows you to provide the Internet Information Service (IIS) path where the new site is created.
   - **Port Number**: This field is used to provide a port number for the Central Configuration.
7. Click against **Create at** to change the path of the Central Configuration.

   A message indicating that if the path has been changed the Central Configuration will no longer be accessible appears.
8. Click **Yes**.

   The IIS structure screen is displayed. Select the desired IIS path for the Central Configuration.
9. Click **OK**.

   The selected URL appears in the **Central Configuration Settings** screen.
10. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:

- The default IIS path for Central Configuration is displayed. Click the edit icon to change the site path.

- If you want to update the authentication type to AVEVA Connect or Azure Active Directory of the already configured Central Configuration, you need to reconfigure the AVEVA Connect and Azure Active Directory user providers for the Central Configuration. For information on configuring user providers, see AVEVA Connect Settings and Azure Active Directory Settings.

- Central Configuration cannot be accessed in Chrome browser when configured with unsecured access (http://) for Azure Active Directory and AVEVA Connect authentication types.

Provision Services

While the installation is in progress, you can view the services that will be installed. This provides information on the progress of installation. At the end of the installation, a status indicating the completion of installation appears.

- Click **OK** to complete the installation.

AVEVA Work Tasks Client

This section describes the activities involved while installing AVEVA Work Tasks on a client machine.

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.
   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.
   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.
   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.
   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.
   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.

4. Click **Next**.

   The License Server details are configured successfully.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.

4. Click **Next**.

   The **Edit Farm Settings** window appears.

5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.

6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.

9. Click **Next**.

   The License Server details are configured successfully.

Select Client Installation

1. In the Select the type of installation page, select **Client**.
2. Click **Next**.

   The Database User Settings page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Connect to an Existing Farm Database

You can use the Connect to existing database dialog box to connect to the existing Farm. You can select the Connection String or Connection provider for the new database.

1. Select the computer name on which the Farm Database is installed from the **Server Name** field.
2. Select **Connection String**.
3. Enter the following details:
   a. **Authentication Type**: This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports both Windows and SQL server authentication process. Example: Windows.
   b. **User ID**: This field is used to enter the SQL user ID for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   c. **Password**: This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   d. **Advanced Settings**: This is an optional field. It is used to enter any additional connection information such as connection Timeout etc. Example: Connection Timeout = 20.

      While creating a data source for repository, use the following custom settings to set properties for the connection pool.

      encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;

      encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

      enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks controls the transaction from escalating to a distributed transaction which is very expensive.

      The above property can be used to turn off this feature. The default value is set as True.

      connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

      maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

      sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

      Note: These settings are case sensitive and should be mentioned only in lower case.
   e. **Database**: This field is used to select the machine with database installed. Example: FarmDB.

      AVEVA Work Tasks will validate if at least one server is installed.
4. Click **Test Connection** to ensure database connectivity.

   You must have the following permissions on the Database Server:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SELECT, INSERT, UPDATE, and DELETE permissions.

EXECUTE permission for Stored Procedures.

5. Click **Next** to continue with the wizard.
6. Click **OK**.

Define and Provision the Enterprise Console

You can define the required AVEVA Work Tasks services.

Steps to Define Enterprise Console

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.
2. Click **Next**.

   The **Installing Services** screen appears, indicating the progress of installation.

   A message *Installation is Complete* appears after all the services are installed.
3. Click **OK**.

Create a Repository DataSource

A **Repository DataSource** is used to store all repository related data. The **Repository DataSource** must be created on a machine where the **Central Configuration** site is installed.

Example

A TelematicsDataSource is created for TelematicsHR Repository.

A Repository contains Workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating the repository.

To create a Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**.

   The Central Configuration web page appears.
2. Click **DataSource Settings**.

   The Datasource List page appears.
3. Click **Add**.

   The Manage Datasource page appears.
4. Enter the required details.
   a. **Name**: Enter a valid data source name. Example: *TelematicsDataSource*.
   b. **Description**: Enter a description for the data source. Example: *This data source is used to store connection details*.
   c. **Connection String**: Select this option to establish a connection with the database. The connection string contains fields that that defines the connection. The connection information is encrypted and stored.

      Server Name

      To use a database on the local machine, enter "localhost", ".", or the local computer's name for the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Server Name.

Note: Server Name can be machine name/IP address/SQL Server AlwaysON/SQL Server instance name. Instead of entering "local host", it is preferable to give the local machine name.

If the **Database Type** is **SQL Server**, then specify an IPV6/IPV4 address for the **Server Name**.
To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: *localhost,1455*
To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be *<server name>\SQLExpress*.

Authentication Type

Use this field to determine the process to validate a user who is trying to connect to the database. AVEVA Work Tasks uses Windows and SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. The **User ID** and **Password** field is disabled if Windows Authentication is selected. Example: *Windows*

Advanced Settings

Use this field to define advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection.
Example: *Connection Timeout = 20*
or
For defining maximum connections that can be opened in a connection pool.
Example: *max pool size = 250*
Default connection pool is set as 250. We recommend to enter a higher value on heavily loaded environment.
While creating a data source for a repository, use the following custom settings to set properties for the connection pool.
encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.
enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature, AVEVA Work Tasks controls the transaction from escalating to an expensive distributed transaction.
The above property can be used to turn off this feature. The default value is set as *True*.
connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is 30 seconds. The value should be set in seconds.
maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is infinite.
sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.
These settings are case sensitive and should be mentioned only in lower case.

Database:

Specify or select a database that can be used as a repository database. This database stores all repository related data. The repository database is created if it does not exist on the database server instance.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example: *TelematicsDB*

Enable Table Partitioning:

Select to enable database table partitioning. For more information, see Partitioning Database.

d. **Connection Provider**: Use this to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.

The selected dll must be copied to all the machines, all servers and all clients, that are connected to the current Farm.

Assembly Path

Click to browse and choose the required dll file.
To use custom properties (enableltm, connectionretrywaittime, maxconnectionretries, and sqlcommandtimeout) in the Assembly connection string, the database must be created before.
See *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Developer Guide* for creating the assembly. Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.

   A message about successful test connection appears.

6. Click **OK**.

   A message about successful creation of the data source appears.The message also states that the required tables will be created in the data source after a repository is associated with the data source.

When a data source is saved, an entry is created for the data source in the **Farm Database**. When a first repository is created for the data source, AVEVA Work Tasks tables are added to the data source. While creating a data source, the archival database is created with the same database name suffixed by *_BI*. This database is used for archival in future use.

Note: After you configure the AVEVA Work Tasks, you must add the created **Farm Database** to the **SQL Server AlwaysOn Availability Groups**. Contact your Database Administrator to add the database to **SQL Server AlwaysOn Availability Groups**.

To view the data source created, click **DataSource Settings** link.

Create a Repository

A **Repository** contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

To create a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The Central Configuration page appears.

2. Click **Repository Settings**.

   The Repository List page appears.

3. Click **Add**.

   Note: If the number of repositories exceeds the permitted limit for the license, then the message "You have exceeded the maximum number of repositories" appears, and the repository creation is stopped. Click **Close**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Manage Repository** page appears.

4. Enter the required details.
5. Click **OK**.

   The **Config Administrator** page appears.
6. Click to configure an administrator.

   The User Lookup dialog box appears.
7. Select the provider from the provider drop-down list.
8. Enter the details to search the administrator for the repository that is created.

   Note: You can also search a user based on *ObjectSID* and *objectGUID* (applicable only for Active Directory provider).
9. Click **Search**.

   The user is listed under **Available Resources**.
10. Select and click the arrow to list the user under **Selected Resource**.
11. Click **Update**.

    The selected user is displayed in **User Lookup**.
12. Click **OK**.

    A message confirming the successful creation of an administrator appears.
13. Click **OK**.

    The **Repository List** page appears.

    For more information about configuring the system user, see Configure System User.

Manage Repository Window Fields

The following table explains the fields of the Manage Repository window:

Field Name	Description	Example
Repository DataSource	Lists all the data sources created in a Farm. The selected data source will be used to store data for the current repository.	TelematicsDatasource
Attach DataSource	Select Attach DataSource checkbox and then select a repository from the selected DataSource .Data associated with the selected repository will be copied to the new repository. The version of the installed product and destination DataSource should be the same as that of the DataSource from which the repository data is to be copied. For	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	example, in an upgraded environment, from build Work Tasks version 1 to Work Tasks version 2, you cannot attach a DataSource created in build Work Tasks version 1 to build Work Tasks version 2. To match the build version of the installed product to the version of the DataSource: 1. If the DataSource was created in build Work Tasks version 1, then the entire build Work Tasks version 1 must be upgraded to Work Tasks version 2 by running the product installer in the upgrade mode. 2. All the Farm, DataSources, Repositories, and Lists of build Work Tasks version 1 must be upgraded to build Work Tasks version 2 by running the AVEVAWorkTasksUpgradeManager.exe located at the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder. 3. After upgrading all the AVEVA Work Tasks components to build Work Tasks version 2 build, that DataSource can be used in any machine that has same version by using this feature.	
Repository Name	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the repository name.	Telematics Repository
Description	Provide a description for the repository.	This stores Telematics Employee Leave Details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
From mail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might occur in the Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be sent to the mail ID defined in **Exception to mail**. The From address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To avoid this, restart the Workflow Engine Service and Communication Service.	Provide the valid email id that is specified in the **User Name** field while configuring **SMTP Settings** in the **Mail Server Settings** page. For more details, see Mail Server Setting.
Exception to mail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails is sent to the email ID specified in **Exception to mail**.	Provide the valid email id.
Alert initiator of the Workflow	Determines if the status of a workflow should be sent as a notification mail to the initiator of the workflow.	A process with a rule to check the pending leave available for the employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
Servers	Lists the servers available in the Farm. Repository will be available only to the mapped servers.	Work Tasks Server Name
Enterprise Consoles	Lists all the Enterprise Consoles for a selected AVEVA Work Tasks Server. Repository will be available only to the mapped Enterprise Console.	http://worktaskservername:8001/EnterpriseConsole

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Enable Theme	Select this checkbox to enable the defined theme.	-
Theme Name	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. It provides **Default** and **Lite** themes.	-
Resource Provider	Lists the various default resource providers supported by AVEVA Work Tasks. Resource providers are also termed as User Provider. A User Provider is a data store containing information about the user. AVEVA Work Tasks also supports custom resource provider. For more information about Resource Provider, see User Providers. AVEVA Connect and Azure Active Directory resource providers are also available after you configure them. To configure these providers, see AVEVA Connect Settings and Azure Active Directory Settings. AVEVA Work Tasks allows secured LDAP configuration. The default port number for LDAPS is 636. You can map the repository in IDE and use the Secured LDAP. The logged-in Active Directory is displayed by default, and **Active Directory Datasource** expects **LDAP://** or **Fully Qualified Domain Name** as value.	AVEVA Work Tasks Lists or Active Directory
Custom Application Folder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with a semicolon.	-
Hide Repository	Select this option to make the repository unavailable to the Farm.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Source path	Enter the path to the Source Elements Folder.	-
Attachment Storage Path	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an absolute path. - The default path is the **Attachments** folder in *C:\ProgramData\AVEVA Work Tasks*. - The default path is used when no path is specified. - Ensure appropriate rights are set to the folder, where attachments are stored, for the following identities: -- The application pool identity for **Central Configuration** and **Enterprise Console(s)**. -- The account configured for the **Workflow Engine** service. - In case of mapped drives, ensure they are available to the identities. - In multiple-server environments, the path specified must be in a shared location.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Auto Log Off Settings for Enterprise Console	Automatically logs out the logged in user after the designated period of inactivity. Enter the time in minutes. This field accepts only valid integer values between 0 to 360. 0 minute means the user will not get automatically logged off. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. The default value for automatic log off is 30 minutes. You use this field to define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML.	0 to 360 (minutes)
Auto Log Off Settings for AVEVA Work Tasks Pro	Request for user credentials for AVEVA Work Tasks Pro app after the set time is elapsed. Enter the time in minutes. This field accepts only valid integer values between 0 to 2880. 0 minute means the user will not get automatically logged off. The default value is 30 minutes. If AVEVA Work Tasks Pro app is offline, the Auto Log Off will not get active, and the user will not be automatically logged off as per the defined settings.	0 to 2880 (in minutes)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Offline Support	Select this option to enable offline submission of items in AVEVA Work Tasks Pro app.	-
Refresh on Work Item submission	Select this option to enable Inbox refresh on Work Item submission in AVEVA Work Tasks Pro . Refresh option is enabled by default.	-
Timeout	Enter the time in seconds to set the timeout. If the value is set as 0, then there is no timeout. Default value is 60 seconds.	-
Label for Fill Form	Enter the label for Fill Form. The customized label will be displayed for the Fill Form in AVEVA Work Tasks Pro.	Work Tasks
Date Configuration Flag for Alternate Actors	Select the **Enable Date Settings** option to set the **From** date and **To** date as mandatory while creating alternate actors. For more information about creating alternate actors, see *Alternate Actors* topic in the *User Guide*.	-
Display Domain name with User Name	Select this checkbox to display the domain name along with the user name in the Enterprise Console and the List. The domain name will be displayed only: • If this checkbox is selected. • If the **Resource Provider** is selected as **Active Directory**.	-

User Providers

AVEVA Work Tasks supports the following User Providers by default:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Active Directory

When you select **Active Directory** as the **Resource Provider**, details such as **Active Directory datasource** and the **Active Directory search base** are populated by default.

Authentication User, **Authentication Password**, **User Log-on ID**, and **User Log-on Display Name** are optional fields. Access to the active directory might not be available for all users. In such cases, provide the active directory authentication user name and the active directory authentication password with which you can connect to the Active Directory.

AVEVA Work Tasks considers *SAM-Account-Name* as **User Log-on ID** by default. If you want to use active directory attributes other than the *SAM-Account-Name* user, then enter the desired active directory attribute details in the **User Log-on ID** field and **User Log-on Display Name** field. **User Log-on Display Name** is the user friendly display name of the active directory attribute. This **User Log-on Display Name** is referred in the User Lookup dialog box against the **User Log-on ID**.

Steps

1. When you select **Active Directory** as the **Resource Provider**, then the **Use Token Service** field appears. In this field, select one of the following:
   - If you do not want to use AIM authentication, select **Without Token Service**.
   - If you want to use AIM authentication, then select a server from the list of alias server names.
2. Enter **Active Directory datasource**.
3. Enter **Active Directory search base**.
4. Enter **Domain Name**.

   The domain name is case-sensitive.
5. Enter **Authentication User**.
6. Enter **Authentication Password**.
7. Check **Is Cross Domain Role Provider** checkbox, if cross domain users and roles mapping is required.

   Example: Users *John* and *Jim* of domain *TelematicsU* and users *Mike* and *Bob* from domain *TelematicsAU* are added to role Team HR of domain *TelematicsMain*.
8. Enter **User Log-on ID** if you want to use another Active Directory attributes instead of *SAM-Account-Name*.

   We recommend to use attributes of type *string* as input for **User Log-on ID.** However, *ObjectSID* and *objectGUID* are also supported.

   After you set the **User Log-on ID**, you cannot change it later.
9. Enter the **User Log-on Display Name**.

   After you set the **User Log-on Display Name**, you cannot change it later.

   This is the default display name of the user account of the active directory user used. **User Log-on Display Name** is used in user lookups.
10. Click **Test LDAP**.

    A message about successful LDAP test appears.

Note:
- Some of the active directory attributes or properties are not indexed. If you use an active directory attribute that is not indexed, it might impact the log-on validation performance based on the active directory entries. We recommend to use indexed attributes as **User Log-on ID** or index the attributes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- For an upgraded repository, you cannot change both **User Log-on ID** and **User Log-on Display Name**.
- You cannot localize **User Log-on ID** and **User log-on Display Name** fields.
- Disabled users will not be shown in user lookup search.

Using Forms

For Forms Resource Provider:

- If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.

1. In the **Resource Provider** drop-down list**,** select **Forms.**
2. In the **User Entity properties** field, enter XML data.

   Click **Sample Code** to get the XML Schema Template.

   Example

   ```
   <properties>
   <property name="userid" type="string" purpose="identifier" />
   <property name="username" type="string" purpose="name" />
   <property name="pwd" type="string" purpose="password" />
   </properties>
   ```
3. In the **User Provider Properties** field, enter the same XML Schema.

   There must be at least one identifier in the XML Schema. For example, UserID
4. In the **Datasource** field, enter a valid data source or specify a dll.

   The value for the data source can be specified as **Physical path of the dll** or **Physical path of the dll::param. Param** is not mandatory.

   See topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Build Business Process* section for creating the assembly.

   Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly, whereas one class can return multiple connection strings.

   Examples of data source parameter:

   - c:\DLLStore\GetConnectionString.dll
   - c:\GetConnectionString.dll::Leave

   If there are two data sources, *Leave* and *Licensing*, and only one DLL is used to return a connection string based on the data source selected. The parameter can be specified as **c:\GetConnectionString.dll::Leave** or **c:\ GetConnectionString.dll::Licensing**. The connection string can be returned from the connection dll based on this parameter.
5. Enter a proper table name or view name which holds the User Name and Password for the Authentication purpose.

   Example: Passmast (the table which is available in IMS_NEW Database)
6. Click **Test SQL Connection**.

   A message about successful test connection appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using ASP.NET Membership

This resource provider lists users defined in the asp.net membership database.

Lists

This resource provider lists users created using AVEVA Work Tasks Users Creation. If you are unable to determine a User Provider to create a repository, use AVEVA Work Tasks Lists.

To verify successful creation of the Repository:

1. Click **Enterprise Console** URL from the site path column.
   The repository created is listed.
   Add the created Enterprise Console link into "Add to Favorites" to easily navigate in future.
2. Click **Repository User Provider** link to logon to the repository.
   The Enterprise Console log-on page appears.
3. Enter the logon details, and then click **Login**.
   The Enterprise Console home page appears. This indicates successful creation of a Repository.
   All features can only be available for a user with administrative privileges.

Note: Consider you created the Repository and then deleted it. If you create the same Repository again immediately and try to log into it through Enterprise Console, the Provider does not list the Repository. You need to access IIS and recycle AVEVA Work Tasks from the Application Pool to get the Repository listed in the Provider list..

Verify the Installation

After you install AVEVA Work Tasks, verify that the product is installed successfully. To verify successful installation, create a simple Workflow using Process Designer. The tasks includes the following:

- Creating a Workflow
- Publishing the Workflow
- Testing the Workflow

Creating a Workflow

1. Log on to the **Enterprise Console**.
2. If the Repository is configured for Active Directory provider, go to step 4.
3. Enter the **User Name** and **Password**.
   The **User Name** and **Password** will be validated with the user provider selected while creating the repository.
   The default authentication provider is the Repository User Provider List.
4. In the **Enterprise Console**, click menu icon.
   All features can only be available for a user with administrative privileges.
5. From the menu, select **Workflows**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Workflow window appears.

6. On the Action bar, click **New**, and then click **Workflow**.
   The New Workflow window appears.
7. Enter **Title** and **Description**.
8. Click **Save & Continue**.
   The successful message appears.
9. Select **Design the Workflow now**.
   This is the default option.
10. Click **Finish**.
    The Process Designer window appears.
11. From the activities pane on the left, drag the Information activity to the Process Designer window.
12. The activity *Information1* is created.
    Click to create a link between two activities.
    The Link Action window appears.
    a. Select from the **From Action** drop-down box.
    b. Select from the **To Action** drop-down box.
    c. Click **OK**.
       The link is established.
13. Right-click *Information1*.
14. Select **Activity Properties** from the context menu.
    The Activity Properties pane appears.
    a. Enter **Name** and **Description**.
    b. Click to **Assign Actor(s)**.
       The Assign Actor(s) window appears.
    c. Select the required provider from the **Providers** drop-down list.
    d. Select values from the drop-down list.
    e. Click **Add**.
    f. Click **Update**.
15. Click to enter the **Subject** for Notification Contents.
    The window to enter subject for notification appears.
16. Click **Update**.
17. Click to enter the **Body** for Notification Contents.
    The window to enter the Body content appears.
18. Click **Update**.
19. From the Activity Properties pane, click **Save**.

Publishing the Workflow

1. On the **Publish** menu, click **Publish**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Published successful message appears.

2. Close the dialog box.

   The Enterprise Console screen lists the workflow created.

Testing the Workflow

1. Select the listed workflow.
2. Right-click the selected workflow.
3. Click **Publish and Test Run**.

   The **Status Finished successfully** appears.

   If Status is **Execution Pending** in the Process Execution window, then click on the **Refresh** button to get the status.
4. In the **Enterprise Console** menu, select **Inbox**.

   The Inbox lists the work item.
5. Click to view the work item listed.

   This indicates that the installation is successful.

Single Server Capable of Adding Machines at Any Time

This section describes the activities involved while installing the Load Balanced Server for the scenario Single Server Capable of Adding Machines at any point in time.

In this scenario the Server is installed on Machine 1. This setup supports scale-out and you can add more machines depending on the number of workflows that need to be processed.

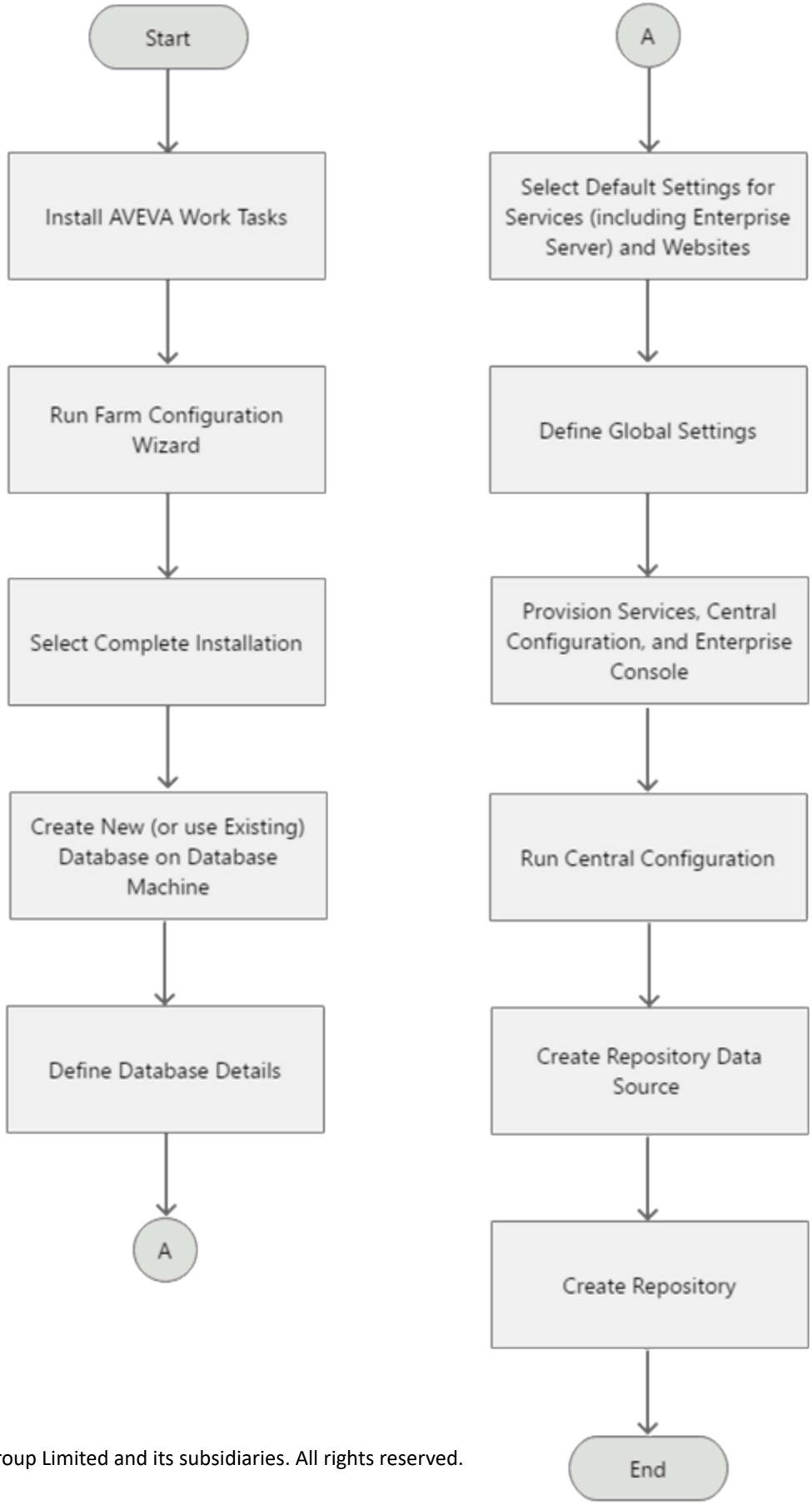
See Also: Single Server Capable of Adding Machines at Any Time

High-Level View of AVEVA Work Tasks Installation

This section provides a high-level view of the various installation steps. The objective of this section is to provide an overview of the various activities involved while installing AVEVA Work Tasks. This section also enables you to visually keep track of your activities while following the sequence of tasks.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation

This section provides installation steps for the scenario Single Server Capable of Adding Machines at any time.

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Wizard.

3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.

5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.

6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.

4. Click **Next**.

   The License Server details are configured successfully.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.

4. Click **Next**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Edit Farm Settings** window appears.

5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.

6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.

9. Click **Next**.

   The License Server details are configured successfully.

Select the Type of Installation - Enterprise Edition

Following points must be considered while installing AVEVA Work Tasks on the Enterprise Server:

- Ensure that you select the **Complete Installation** and **Enable load balancing** options.
- You can either create a new Farm database or use an existing one.

1. In the Select the type of installation page, select **Complete Installation (Server)**.

   A Complete Standalone Deployment does not require a client installation.

2. Select the **Enable load balancing (This feature requires and Enterprise license.)** check box for installing AVEVA Work Tasks on the Enterprise Server.

   The **Farm Configuration Wizard** dialog box appears.

3. Click **OK**.
4. To create a new Farm database, click **Create AVEVA Work Tasks Farm**, and then select one of the following:
   - **Create AVEVA Work Tasks Farm database**
   - **Use existing AVEVA Work Tasks Farm database (Creates AVEVA Work Tasks Farm tables only if they do not exist)**
5. To connect to an existing Farm database, click **Connect to existing AVEVA Work Tasks Farm (if the first server of the Enterprise Server is already installed)**.
6. Click **Next**.

   The Database User Settings page appears.

Create Farm Database

Use the Create AVEVA Work Tasks Farm Database page to provide **New Database** details. Select the **Connection String** option to establish a connection with the database. This connection information is encrypted and stored.

You can enter either Windows or Microsoft SQL Server user settings for authentication.

Note:
- The Farm database password does not support **&** and **;** special characters.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- While setting up **Farm Database** with **Windows Authentication**, do not enter **FQDN** in the **Domain Name** field.

To Create Farm Database

1. Enter **Database Title**.
2. Select **Connection String**.

   See *Connection Provider* later in this topic, if you have selected the **Connection Provider** option.

   - Enter the following details:

     Server Name:

     Provide information about the database server on which the Farm database can be installed.

     If the **Database Type** is **SQL Server**, then you can specify an IPv6 address for the **Server Name**.

     If SQLExpress Instance need to be used as a Database instance of the Farm database, then **Server Name** should be <server name>\SQLExpress.

     To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

   - **Authentication Type**

     This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports Windows and SQL Server authentication process.

     Example: Windows
     If Windows Authentication is selected, the user configured for the Farm configuration will be used to access the database. This is the authentication recommended by AVEVA Work Tasks.

   - **Login**

     This field is used to enter the SQL login credential for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Password**

     This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Advanced Settings**

     This is an optional field. It is used to enter any additional connection information, such as connection Timeout. Example: Connection Timeout = 20.

     While creating a data source for repository, use the following custom settings to set properties for connection pool. These settings are case sensitive and should be mentioned only in lower case.

     ```
     encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
     ```

     encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

     enableltm: AVEVA Work Tasks internally implements light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive. This property can be used to turn off this feature. The default value is set as True.

     connectionretrywaittime: While opening a new connection, if the data base server is not available, the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

maxconnectionretries: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

3. Click **Test Connection** to ensure database connectivity.
4. Click **Next**.

   The message *AVEVA Work Tasks Farm is Created* appears.
5. Click **OK**.

AlwaysOn Availability Groups feature of Microsoft SQL Server

The **AlwaysOn Availability Groups** feature of Microsoft SQL Server is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in Microsoft SQL Server 2012, AlwaysOn Availability Groups maximizes the availability of a set of user databases for an enterprise.

After you configure the AVEVA Work Tasks, you must add the created Farm Database to the SQL Server AlwaysOn Availability Groups. Contact your Database Administrator to add the database to SQL Server AlwaysOn Availability Groups.

Notes:

- The authentication user needs different rights assigned for different scenarios. If a user wants to create database (Farm or Repository Data Source Database), Authentication User needs to have minimum 'dbcreator' access rights on Database Server Instance.

- If database is already created, user needs to have following permissions on Database Server:

- SELECT, INSERT, UPDATE, and DELETE permissions.

- EXECUTE permission for Stored Procedures.

To choose Connection Provider

Complete the following steps to choose a connection provider:

1. Enter **Database Title**.
2. In the **Database Type** box, select a database type.
3. Select **Connection Provider**.

   This option is used to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.
4. Enter the following details:
   - **Assembly Path**:

     Click the icon to browse and choose the required dll file.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If custom properties (enableltm, connectionretrywaittime, maxconnectionretries and sqlcommandtimeout) are required to be used in Assembly connection string, then user should select **Use existing Farm database** while creating a the Farm. See Creating or Using an Existing Database.

See the topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub chapter in the *Build Business Process* section for creating the assembly. Make sure that, there should be only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.
6. Click **Next**.
7. The message *AVEVA Work Tasks Farm is created* appears.
8. Click **OK**.

Define Security Settings

You can define the Security Settings using the Farm Configuration Wizard.

For more information on Security settings, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client.

To define the Security Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Security Settings** and **Service Settings** and then, click **Next**.
4. Select **Secure Communication** to establish secured communication between the AVEVA Work Tasks services through HTTPS.
5. Select **Secure Web Sites** check box to configure secured web sites. Uncheck the **Secure Web Sites** check box to use HTTP sites. By default the sites are created in HTTPS mode.
6. Enter the Port number. The default port for the HTTPS site is 8001, and for the HTTP site is 8000. You can enter any port number that IIS accepts. But if any other services or applications are already using the entered port number, AVEVA Work Tasks will not start.
7. Select the available certificate from the **Certificate** list. If your certificate is not listed, click the **Refresh** icon to repopulate the list.

   Note: If the required certificate is not listed, generate one using the System Management Server (Configurator tool). Do not use a self-signed certificate without a proper root chain.
8. Click the **Validate** button to validate the certificate before configuring the security settings.

   Note: An Internet connection is needed to validate the certificate.
9. Click **Configure**, then **Next**, and then **Finish** to apply the new security settings.

Note:
- To update the port number, certificate, and websites from unsecure to secure or vice-versa, select **Security Settings** and **Service Settings** check box in **Edit AVEVA Work Task Farm Settings** window before updating the values. After the required details are provided, these settings get updated only when you click **Configure**, then **Next**, and then **Finish** as you navigate through the Farm Configuration Wizard..

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- If the Quickflow is already deployed on a machine, you need to deploy it again after you update the security settings. If secured web site is configured, then you cannot change the server configuration for Quickflow in Central Configuration. However, if the **Secure Web Sites** checkbox is not selected, you can make the changes to the server configuration.
- Advance Server communication cannot be configured on HTTP/HTTPS.
- For resources of Load Balanced Server, ensure to configure all services with the same protocol (HTTP or HTTPS).
- It is recommended to use Standard SSL certificate instead of self-signed SSL certificate.

Using ASB Certificate for Secured Communication

ASB Certificate is a PCS-managed SSL certificate and renews automatically every 30 days when the certificate is still valid. The certificate is valid for two years. The PCS watchdog service on the runtime node checks the validity of the SSL certificate within 5 minutes from when it starts and once a day. PCS also renews the SSL certificate if it is expired.

If the runtime node is offline and later back online, the PCS watchdog service regularly checks the certificate validity and renews it if it is required. Consider restarting the PCS watchdog service to enforce the SSL certificate validity check within 5 minutes from the restarting time.

Furthermore, when the SSL certificate is renewed, PCS binds it to port 443 (by default) and changes the SslThumbprint PCS registry value. If AVEVA Work Tasks sites are hosted on port 443, certificate binding will be auto-renewed when the SSL certificate is renewed. It is recommended to check the bindings in IIS with every certificate renewal to ensure the application is running seamlessly.

To disable the certificate renewal, consider disabling the **ArchestrA Certificate Renewal Service** windows service.

Note: Client Service will try to automatically bind the renewed certificate. If there are any issues in browsing the website or starting the services after certificate renewal, users need to reconfigure the secure communication in Farm Configuration. To reconfigure, see the steps mentioned in above section *To define the Security Settings*.

Securing Communication between Services

To have a secure communication between the services, apply the below changes to the Windows registry and reboot the server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
```

For more information on securing communication, see https://docs.microsoft.com/en-us/mem/configmgr/core/

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

plan-design/security/enable-tls-1-2-client and
https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2015/2960358.

Define Services and Global Settings

You can define the Services and Global Settings using the Farm Configuration Wizard or through Configuration XMLs.

To define Services

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.

   Note: You can also customize the settings for these services and websites using the **Custom Settings** option. For more information about Service-related changes and Website-related changes, see Advanced Settings and Administering Web Servers respectively.

2. Click **Next**.

   The Global Settings dialog box appears.

3. In the **Communication** tab, enter key fields in **Email Settings** and **Exception Email settings** to configure exception e-mails.

   The following are the details:

   - **Host**: Enter information related to mail server.
     Example: Enter details related to Microsoft Exchange Server (IP address of the exchange server).
   - **Port**: Enter the port number for the mail server. The default port number is 25.
   - **From email**: Enter the from e-mail address.
     Example: admin@telematics.com
   - **To email**: Enter the to e-mail address.
     Example: alex@telematics.com

4. Click **OK**.

Note: To confirm the exception e-mail settings, click **Test SMTP**. An e-mail will be generated and sent to the **To email** address confirming successful configuration of the e-mail. The advanced settings like **Secure Socket Layer (SSL), Email Timeout, User Name,** and **Password** for the e-mail account holder, **Password Encrypted** flag will differ based on the mail server settings.

Provision Services

While the installation is in progress, you can view the services that will be installed. This provides information on the progress of installation. At the end of the installation, a status indicating the completion of installation appears.

- Click **OK** to complete the installation.

Create a Repository DataSource

A **Repository DataSource** is used to store all repository related data. The **Repository DataSource** must be

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

created on a machine where the **Central Configuration** site is installed.

Example

A TelematicsDataSource is created for TelematicsHR Repository.

A Repository contains Workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating the repository.

To create a Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**.

   The Central Configuration web page appears.
2. Click **DataSource Settings**.

   The Datasource List page appears.
3. Click **Add**.

   The Manage Datasource page appears.
4. Enter the required details.
   a. **Name**: Enter a valid data source name. Example: *TelematicsDataSource*.
   b. **Description**: Enter a description for the data source. Example: *This data source is used to store connection details*.
   c. **Connection String**: Select this option to establish a connection with the database. The connection string contains fields that that defines the connection. The connection information is encrypted and stored.

      Server Name

      To use a database on the local machine, enter "localhost", ".", or the local computer's name for the **Server Name**.

      Note: Server Name can be machine name/IP address/SQL Server AlwaysON/SQL Server instance name. Instead of entering "local host", it is preferable to give the local machine name.

      If the **Database Type** is **SQL Server**, then specify an IPV6/IPV4 address for the **Server Name**.
      To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: *localhost,1455*
      To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be *<server name>\SQLExpress*.

      Authentication Type

      Use this field to determine the process to validate a user who is trying to connect to the database. AVEVA Work Tasks uses Windows and SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. The **User ID** and **Password** field is disabled if Windows Authentication is selected. Example: *Windows*

      Advanced Settings

      Use this field to define advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection.
      Example: *Connection Timeout = 20*
      or
      For defining maximum connections that can be opened in a connection pool.
      Example: *max pool size = 250*
      Default connection pool is set as 250. We recommend to enter a higher value on heavily loaded environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

While creating a data source for a repository, use the following custom settings to set properties for the connection pool.
encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.
enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature, AVEVA Work Tasks controls the transaction from escalating to an expensive distributed transaction.
The above property can be used to turn off this feature. The default value is set as *True*.
connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is 30 seconds. The value should be set in seconds.
maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is infinite.
sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.
These settings are case sensitive and should be mentioned only in lower case.

Database:

Specify or select a database that can be used as a repository database. This database stores all repository related data. The repository database is created if it does not exist on the database server instance.
Example: *TelematicsDB*

Enable Table Partitioning:

Select to enable database table partitioning. For more information, see Partitioning Database.

d. **Connection Provider**: Use this to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.

The selected dll must be copied to all the machines, all servers and all clients, that are connected to the current Farm.

Assembly Path

Click to browse and choose the required dll file.
To use custom properties (enableltm, connectionretrywaittime, maxconnectionretries, and sqlcommandtimeout) in the Assembly connection string, the database must be created before.
See *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Developer Guide* for creating the assembly. Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.

   A message about successful test connection appears.

6. Click **OK**.

   A message about successful creation of the data source appears.The message also states that the required tables will be created in the data source after a repository is associated with the data source.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

When a data source is saved, an entry is created for the data source in the **Farm Database**. When a first repository is created for the data source, AVEVA Work Tasks tables are added to the data source. While creating a data source, the archival database is created with the same database name suffixed by *_BI*. This database is used for archival in future use.

Note: After you configure the AVEVA Work Tasks, you must add the created **Farm Database** to the **SQL Server AlwaysOn Availability Groups**. Contact your Database Administrator to add the database to **SQL Server AlwaysOn Availability Groups**.

To view the data source created, click **DataSource Settings** link.

Create a Repository

A **Repository** contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

To create a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The Central Configuration page appears.
2. Click **Repository Settings**.

   The Repository List page appears.
3. Click **Add**.

   Note: If the number of repositories exceeds the permitted limit for the license, then the message "You have exceeded the maximum number of repositories" appears, and the repository creation is stopped. Click **Close**.

   The **Manage Repository** page appears.
4. Enter the required details.
5. Click **OK**.

   The **Config Administrator** page appears.
6. Click to configure an administrator.

   The User Lookup dialog box appears.
7. Select the provider from the provider drop-down list.
8. Enter the details to search the administrator for the repository that is created.

   Note: You can also search a user based on *ObjectSID* and *objectGUID* (applicable only for Active Directory provider).
9. Click **Search**.

   The user is listed under **Available Resources**.
10. Select and click the arrow to list the user under **Selected Resource**.
11. Click **Update**.

    The selected user is displayed in **User Lookup**.
12. Click **OK**.

    A message confirming the successful creation of an administrator appears.
13. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Repository List** page appears.

For more information about configuring the system user, see Configure System User.

Manage Repository Window Fields

The following table explains the fields of the Manage Repository window:

Field Name	Description	Example
Repository DataSource	Lists all the data sources created in a Farm. The selected data source will be used to store data for the current repository.	TelematicsDatasource
Attach DataSource	Select Attach DataSource checkbox and then select a repository from the selected DataSource .Data associated with the selected repository will be copied to the new repository. The version of the installed product and destination DataSource should be the same as that of the DataSource from which the repository data is to be copied. For example, in an upgraded environment, from build Work Tasks version 1 to Work Tasks version 2, you cannot attach a DataSource created in build Work Tasks version 1 to build Work Tasks version 2. To match the build version of the installed product to the version of the DataSource: 1. If the DataSource was created in build Work Tasks version 1, then the entire build Work Tasks version 1 must be upgraded to Work Tasks version 2 by running the product installer in the upgrade mode. 2. All the Farm, DataSources, Repositories, and Lists of build Work Tasks version 1 must be upgraded to build Work Tasks	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	version 2 by running the AVEVAWorkTasksUpgradeManager.exe located at the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder. 3. After upgrading all the AVEVA Work Tasks components to build Work Tasks version 2 build, that DataSource can be used in any machine that has same version by using this feature.	
Repository Name	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the repository name.	Telematics Repository
Description	Provide a description for the repository.	This stores Telematics Employee Leave Details.
From mail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might occur in the Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be sent to the mail ID defined in **Exception to mail**. The From address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To avoid this, restart the Workflow Engine Service and Communication Service.	Provide the valid email id that is specified in the **User Name** field while configuring **SMTP Settings** in the **Mail Server Settings** page. For more details, see Mail Server Setting.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Exception to mail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails is sent to the email ID specified in **Exception to mail**.	Provide the valid email id.
Alert initiator of the Workflow	Determines if the status of a workflow should be sent as a notification mail to the initiator of the workflow.	A process with a rule to check the pending leave available for the employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
Servers	Lists the servers available in the Farm. Repository will be available only to the mapped servers.	Work Tasks Server Name
Enterprise Consoles	Lists all the Enterprise Consoles for a selected AVEVA Work Tasks Server. Repository will be available only to the mapped Enterprise Console.	http://worktaskservername:8001/EnterpriseConsole
Enable Theme	Select this checkbox to enable the defined theme.	-
Theme Name	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. It provides **Default** and **Lite** themes.	-
Resource Provider	Lists the various default resource providers supported by AVEVA Work Tasks. Resource providers are also termed as User Provider. A User Provider is a data store containing information about the user. AVEVA Work Tasks also supports custom resource provider. For more information about Resource Provider, see User Providers. AVEVA Connect and Azure Active Directory resource providers are	AVEVA Work Tasks Lists or Active Directory

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	also available after you configure them. To configure these providers, see AVEVA Connect Settings and Azure Active Directory Settings. AVEVA Work Tasks allows secured LDAP configuration. The default port number for LDAPS is 636. You can map the repository in IDE and use the Secured LDAP. The logged-in Active Directory is displayed by default, and **Active Directory Datasource** expects **LDAP://** or **Fully Qualified Domain Name** as value.	
Custom Application Folder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with a semicolon.	-
Hide Repository	Select this option to make the repository unavailable to the Farm.	-
Source path	Enter the path to the Source Elements Folder.	-
Attachment Storage Path	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an absolute path. - The default path is the **Attachments** folder in *C:\ProgramData\AVEVA Work Tasks*\. - The default path is used when no path is specified. - Ensure appropriate rights are set to the folder, where attachments are stored, for the following identities: -- The application pool identity for **Central Configuration** and **Enterprise Console(s)**. -- The account configured for the	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	Workflow Engine service. - In case of mapped drives, ensure they are available to the identities. - In multiple-server environments, the path specified must be in a shared location.	
Auto Log Off Settings for Enterprise Console	Automatically logs out the logged in user after the designated period of inactivity. Enter the time in minutes. This field accepts only valid integer values between 0 to 360. 0 minute means the user will not get automatically logged off. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. The default value for automatic log off is 30 minutes. You use this field to define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML.	0 to 360 (minutes)
Auto Log Off Settings for AVEVA Work Tasks Pro	Request for user credentials for AVEVA Work Tasks Pro app after the set time is elapsed. Enter the time in minutes. This field accepts only valid integer values between 0 to 2880. 0 minute means the user will not get automatically logged off. The default value is 30 minutes.	0 to 2880 (in minutes)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	If AVEVA Work Tasks Pro app is offline, the Auto Log Off will not get active, and the user will not be automatically logged off as per the defined settings.	
Offline Support	Select this option to enable offline submission of items in AVEVA Work Tasks Pro app.	-
Refresh on Work Item submission	Select this option to enable Inbox refresh on Work Item submission in AVEVA Work Tasks Pro . Refresh option is enabled by default.	-
Timeout	Enter the time in seconds to set the timeout. If the value is set as 0, then there is no timeout. Default value is 60 seconds.	-
Label for Fill Form	Enter the label for Fill Form. The customized label will be displayed for the Fill Form in AVEVA Work Tasks Pro.	Work Tasks
Date Configuration Flag for Alternate Actors	Select the **Enable Date Settings** option to set the **From** date and **To** date as mandatory while creating alternate actors. For more information about creating alternate actors, see *Alternate Actors* topic in the *User Guide*.	-
Display Domain name with User Name	Select this checkbox to display the domain name along with the user name in the Enterprise Console and the List. The domain name will be displayed only: • If this checkbox is selected. • If the **Resource Provider** is selected as **Active Directory**.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

User Providers

AVEVA Work Tasks supports the following User Providers by default:

Active Directory

When you select **Active Directory** as the **Resource Provider**, details such as **Active Directory datasource** and the **Active Directory search base** are populated by default.

Authentication User, **Authentication Password**, **User Log-on ID**, and **User Log-on Display Name** are optional fields. Access to the active directory might not be available for all users. In such cases, provide the active directory authentication user name and the active directory authentication password with which you can connect to the Active Directory.

AVEVA Work Tasks considers *SAM-Account-Name* as **User Log-on ID** by default. If you want to use active directory attributes other than the *SAM-Account-Name* user, then enter the desired active directory attribute details in the **User Log-on ID** field and **User Log-on Display Name** field. **User Log-on Display Name** is the user friendly display name of the active directory attribute. This **User Log-on Display Name** is referred in the User Lookup dialog box against the **User Log-on ID**.

Steps

1. When you select **Active Directory** as the **Resource Provider**, then the **Use Token Service** field appears. In this field, select one of the following:
   - If you do not want to use AIM authentication, select **Without Token Service**.
   - If you want to use AIM authentication, then select a server from the list of alias server names.
2. Enter **Active Directory datasource**.
3. Enter **Active Directory search base**.
4. Enter **Domain Name**.

   The domain name is case-sensitive.
5. Enter **Authentication User**.
6. Enter **Authentication Password**.
7. Check **Is Cross Domain Role Provider** checkbox, if cross domain users and roles mapping is required.

   Example: Users *John* and *Jim* of domain *TelematicsU* and users *Mike* and *Bob* from domain *TelematicsAU* are added to role Team HR of domain *TelematicsMain*.
8. Enter **User Log-on ID** if you want to use another Active Directory attributes instead of *SAM-Account-Name*.

   We recommend to use attributes of type *string* as input for **User Log-on ID.** However, *ObjectSID* and *objectGUID* are also supported.

   After you set the **User Log-on ID**, you cannot change it later.
9. Enter the **User Log-on Display Name**.

   After you set the **User Log-on Display Name**, you cannot change it later.

   This is the default display name of the user account of the active directory user used. **User Log-on Display Name** is used in user lookups.
10. Click **Test LDAP**.

    A message about successful LDAP test appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:
- Some of the active directory attributes or properties are not indexed. If you use an active directory attribute that is not indexed, it might impact the log-on validation performance based on the active directory entries. We recommend to use indexed attributes as **User Log-on ID** or index the attributes.
- For an upgraded repository, you cannot change both **User Log-on ID** and **User Log-on Display Name**.
- You cannot localize **User Log-on ID** and **User log-on Display Name** fields.
- Disabled users will not be shown in user lookup search.

Using Forms

For Forms Resource Provider:

- If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.

1. In the **Resource Provider** drop-down list**,** select **Forms.**
2. In the **User Entity properties** field, enter XML data.

   Click **Sample Code** to get the XML Schema Template.

   Example

   ```
   <properties>
   <property name="userid" type="string" purpose="identifier" />
   <property name="username" type="string" purpose="name" />
   <property name="pwd" type="string" purpose="password" />
   </properties>
   ```
3. In the **User Provider Properties** field, enter the same XML Schema.

   There must be at least one identifier in the XML Schema. For example, UserID
4. In the **Datasource** field, enter a valid data source or specify a dll.

   The value for the data source can be specified as **Physical path of the dll** or **Physical path of the dll::param. Param** is not mandatory.

   See topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Build Business Process* section for creating the assembly.

   Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly, whereas one class can return multiple connection strings.

   Examples of data source parameter:

   - c:\DLLStore\GetConnectionString.dll
   - c:\GetConnectionString.dll::Leave

   If there are two data sources, *Leave* and *Licensing*, and only one DLL is used to return a connection string based on the data source selected. The parameter can be specified as **c:\GetConnectionString.dll::Leave** or **c:\ GetConnectionString.dll::Licensing**. The connection string can be returned from the connection dll based on this parameter.
5. Enter a proper table name or view name which holds the User Name and Password for the Authentication purpose.

   Example: Passmast (the table which is available in IMS_NEW Database)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Click **Test SQL Connection**.

   A message about successful test connection appears.

Using ASP.NET Membership

This resource provider lists users defined in the asp.net membership database.

Lists

This resource provider lists users created using AVEVA Work Tasks Users Creation. If you are unable to determine a User Provider to create a repository, use AVEVA Work Tasks Lists.

To verify successful creation of the Repository:

1. Click **Enterprise Console** URL from the site path column.

   The repository created is listed.

   Add the created Enterprise Console link into "Add to Favorites" to easily navigate in future.
2. Click **Repository User Provider** link to logon to the repository.

   The Enterprise Console log-on page appears.
3. Enter the logon details, and then click **Login**.

   The Enterprise Console home page appears. This indicates successful creation of a Repository.

   All features can only be available for a user with administrative privileges.

Note: Consider you created the Repository and then deleted it. If you create the same Repository again immediately and try to log into it through Enterprise Console, the Provider does not list the Repository. You need to access IIS and recycle AVEVA Work Tasks from the Application Pool to get the Repository listed in the Provider list..

Verify the Installation

After you install AVEVA Work Tasks, verify that the product is installed successfully. To verify successful installation, create a simple Workflow using Process Designer. The tasks includes the following:

- Creating a Workflow
- Publishing the Workflow
- Testing the Workflow

Creating a Workflow

1. Log on to the **Enterprise Console**.
2. If the Repository is configured for Active Directory provider, go to step 4.
3. Enter the **User Name** and **Password**.

   The **User Name** and **Password** will be validated with the user provider selected while creating the repository.

   The default authentication provider is the Repository User Provider List.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. In the **Enterprise Console**, click menu icon.
   All features can only be available for a user with administrative privileges.
5. From the menu, select **Workflows**.
   The Workflow window appears.
6. On the Action bar, click **New**, and then click **Workflow**.
   The New Workflow window appears.
7. Enter **Title** and **Description**.
8. Click **Save & Continue**.
   The successful message appears.
9. Select **Design the Workflow now**.
   This is the default option.
10. Click **Finish**.
    The Process Designer window appears.
11. From the activities pane on the left, drag the Information activity to the Process Designer window.
12. The activity *Information1* is created.
    Click to create a link between two activities.
    The Link Action window appears.
    a. Select from the **From Action** drop-down box.
    b. Select from the **To Action** drop-down box.
    c. Click **OK**.
       The link is established.
13. Right-click *Information1*.
14. Select **Activity Properties** from the context menu.
    The Activity Properties pane appears.
    a. Enter **Name** and **Description**.
    b. Click to **Assign Actor(s)**.
       The Assign Actor(s) window appears.
    c. Select the required provider from the **Providers** drop-down list.
    d. Select values from the drop-down list.
    e. Click **Add**.
    f. Click **Update**.
15. Click to enter the **Subject** for Notification Contents.
    The window to enter subject for notification appears.
16. Click **Update**.
17. Click to enter the **Body** for Notification Contents.
    The window to enter the Body content appears.
18. Click **Update**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

19. From the Activity Properties pane, click **Save**.

Publishing the Workflow

1. On the **Publish** menu, click **Publish**.

   The Published successful message appears.
2. Close the dialog box.

   The Enterprise Console screen lists the workflow created.

Testing the Workflow

1. Select the listed workflow.
2. Right-click the selected workflow.
3. Click **Publish and Test Run**.

   The **Status Finished successfully** appears.

   If Status is **Execution Pending** in the Process Execution window, then click on the **Refresh** button to get the status.
4. In the **Enterprise Console** menu, select **Inbox**.

   The Inbox lists the work item.
5. Click to view the work item listed.

   This indicates that the installation is successful.

Intranet and Extranet Web Servers on Different Machines

This section describes the activities involved while installing the Load Balanced Server for the scenario Intranet and Extranet Web Servers on Different Machines.

In this scenario, the Server is installed on Machine 1 and the Client is installed on multiple Web Farms namely Machine 3 (or Server 1), Server 2 and Server N in a Web Farm. "Server N" represents the number of machines in the Web Farm.

Ensure that you install the product in the following sequence:

- High-Level View of Server Installation
- High-Level View of Client Installation

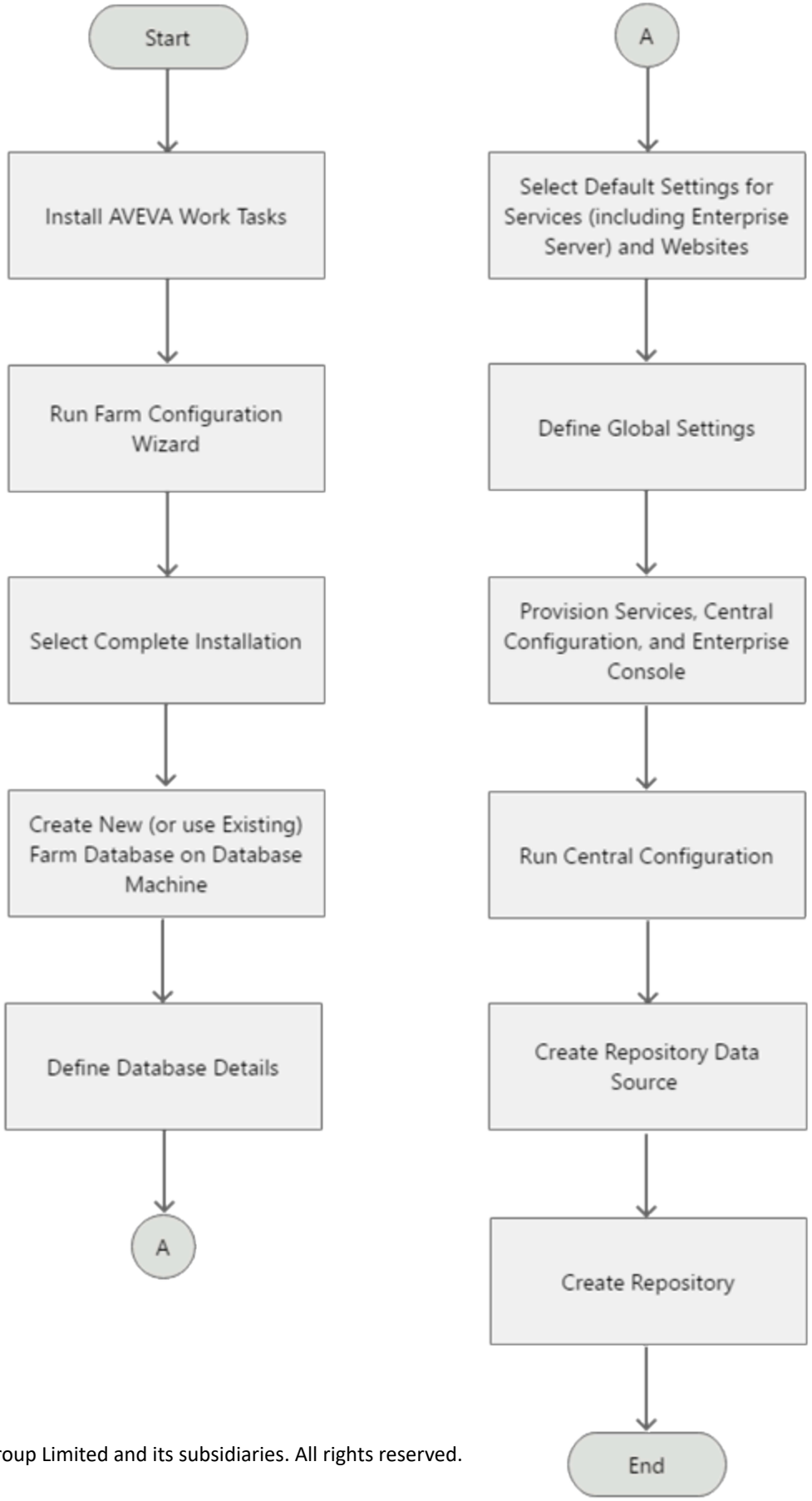
See Also: Intranet and Extranet Web Servers on Different Machines

High-Level View of Server Installation

This section provides a high-level view of the various installation steps. The objective of this section is to provide an overview of the various activities involved while installing AVEVA Work Tasks Server. This section also enables you to visually keep track of your activities while following the sequence of tasks.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



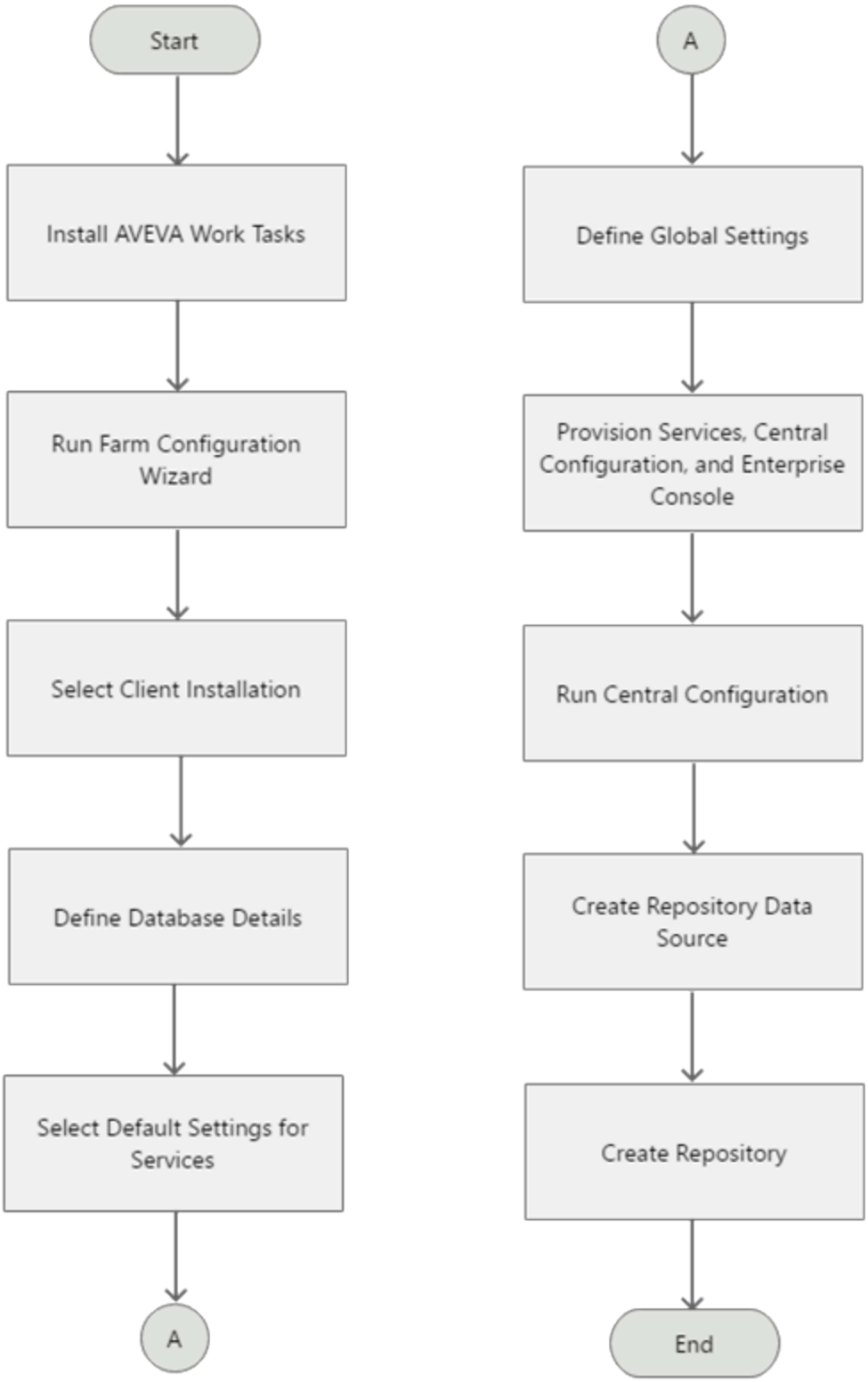
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

High-Level View of Client Installation

This section provides a high-level view of AVEVA Work Tasks Client installation. The Client components are installed on Web Servers. For more information on this scenario, see Load Balanced Server.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation

This section provides installation steps for the scenario Intranet and Extranet Web Servers on Different Machines.

The installation includes the following:

- Installing AVEVA Work Tasks Server
- Installing AVEVA Work Tasks Client

AVEVA Work Tasks Server has to be installed before installing AVEVA Work Tasks Client. In this scenario, the Client has to be installed on multiple machines.

AVEVA Work Tasks Server

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the current MSDTC settings.

2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.

3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.

5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.

6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.

4. Click **Next**.

   The License Server details are configured successfully.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.
4. Click **Next**.

   The **Edit Farm Settings** window appears.
5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.
6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
9. Click **Next**.

   The License Server details are configured successfully.

Select the Type of Installation - Enterprise Edition

Following points must be considered while installing AVEVA Work Tasks on the Enterprise Server:

- Ensure that you select the **Complete Installation** and **Enable load balancing** options.
- You can either create a new Farm database or use an existing one.

1. In the Select the type of installation page, select **Complete Installation (Server)**.

   A Complete Standalone Deployment does not require a client installation.
2. Select the **Enable load balancing (This feature requires and Enterprise license.)** check box for installing AVEVA Work Tasks on the Enterprise Server.

   The **Farm Configuration Wizard** dialog box appears.
3. Click **OK**.
4. To create a new Farm database, click **Create AVEVA Work Tasks Farm**, and then select one of the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Create AVEVA Work Tasks Farm database**
- **Use existing AVEVA Work Tasks Farm database (Creates AVEVA Work Tasks Farm tables only if they do not exist)**

5. To connect to an existing Farm database, click **Connect to existing AVEVA Work Tasks Farm (if the first server of the Enterprise Server is already installed)**.
6. Click **Next**.

   The Database User Settings page appears.

Create Farm Database

Use the Create AVEVA Work Tasks Farm Database page to provide **New Database** details. Select the **Connection String** option to establish a connection with the database. This connection information is encrypted and stored.

You can enter either Windows or Microsoft SQL Server user settings for authentication.

Note:
- The Farm database password does not support **&** and **;** special characters.
- While setting up **Farm Database** with **Windows Authentication**, do not enter **FQDN** in the **Domain Name** field.

To Create Farm Database

1. Enter **Database Title**.
2. Select **Connection String**.

   See *Connection Provider* later in this topic, if you have selected the **Connection Provider** option.

   - Enter the following details:

     Server Name:

     Provide information about the database server on which the Farm database can be installed.

     If the **Database Type** is **SQL Server**, then you can specify an IPv6 address for the **Server Name**.

     If SQLExpress Instance need to be used as a Database instance of the Farm database, then **Server Name** should be <server name>\SQLExpress.

     To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

   - **Authentication Type**

     This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports Windows and SQL Server authentication process.

     Example: Windows
     If Windows Authentication is selected, the user configured for the Farm configuration will be used to access the database. This is the authentication recommended by AVEVA Work Tasks.

   - **Login**

     This field is used to enter the SQL login credential for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Password**

     This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Advanced Settings**

  This is an optional field. It is used to enter any additional connection information, such as connection Timeout. Example: Connection Timeout = 20.

  While creating a data source for repository, use the following custom settings to set properties for connection pool. These settings are case sensitive and should be mentioned only in lower case.

  ```
  encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
  ```

  encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

  enableltm: AVEVA Work Tasks internally implements light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive. This property can be used to turn off this feature. The default value is set as True.

  connectionretrywaittime: While opening a new connection, if the data base server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

  maxconnectionretries: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

  sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

3. Click **Test Connection** to ensure database connectivity.
4. Click **Next**.

   The message *AVEVA Work Tasks Farm is Created* appears.
5. Click **OK**.

AlwaysOn Availability Groups feature of Microsoft SQL Server

The **AlwaysOn Availability Groups** feature of Microsoft SQL Server is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in Microsoft SQL Server 2012, AlwaysOn Availability Groups maximizes the availability of a set of user databases for an enterprise.

After you configure the AVEVA Work Tasks, you must add the created Farm Database to the SQL Server AlwaysOn Availability Groups. Contact your Database Administrator to add the database to SQL Server AlwaysOn Availability Groups.

Notes:

- The authentication user needs different rights assigned for different scenarios. If a user wants to create database (Farm or Repository Data Source Database), Authentication User needs to have minimum 'dbcreator' access rights on Database Server Instance.

- If database is already created, user needs to have following permissions on Database Server:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- SELECT, INSERT, UPDATE, and DELETE permissions.

- EXECUTE permission for Stored Procedures.

To choose Connection Provider

Complete the following steps to choose a connection provider:

1. Enter **Database Title**.
2. In the **Database Type** box, select a database type.
3. Select **Connection Provider**.

   This option is used to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.
4. Enter the following details:
   - **Assembly Path**:

     Click the icon to browse and choose the required dll file.

     If custom properties (enableltm, connectionretrywaittime, maxconnectionretries and sqlcommandtimeout) are required to be used in Assembly connection string, then user should select **Use existing Farm database** while creating a the Farm. See Creating or Using an Existing Database.

     See the topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub chapter in the *Build Business Process* section for creating the assembly. Make sure that, there should be only one class implementing *IDBConnectionString* interface within this assembly.
5. Click **Test Connection** to ensure database connectivity.
6. Click **Next**.
7. The message *AVEVA Work Tasks Farm is created* appears.
8. Click **OK**.

Define Security Settings

You can define the Security Settings using the Farm Configuration Wizard.

For more information on Security settings, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client.

To define the Security Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Security Settings** and **Service Settings** and then, click **Next**.
4. Select **Secure Communication** to establish secured communication between the AVEVA Work Tasks services through HTTPS.
5. Select **Secure Web Sites** check box to configure secured web sites. Uncheck the **Secure Web Sites** check box

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

to use HTTP sites. By default the sites are created in HTTPS mode.

6. Enter the Port number. The default port for the HTTPS site is 8001, and for the HTTP site is 8000. You can enter any port number that IIS accepts. But if any other services or applications are already using the entered port number, AVEVA Work Tasks will not start.
7. Select the available certificate from the **Certificate** list. If your certificate is not listed, click the **Refresh** icon to repopulate the list.

   Note: If the required certificate is not listed, generate one using the System Management Server (Configurator tool). Do not use a self-signed certificate without a proper root chain.
8. Click the **Validate** button to validate the certificate before configuring the security settings.

   Note: An Internet connection is needed to validate the certificate.
9. Click **Configure**, then **Next**, and then **Finish** to apply the new security settings.

Note:
- To update the port number, certificate, and websites from unsecure to secure or vice-versa, select **Security Settings** and **Service Settings** check box in **Edit AVEVA Work Task Farm Settings** window before updating the values. After the required details are provided, these settings get updated only when you click **Configure**, then **Next**, and then **Finish** as you navigate through the Farm Configuration Wizard..
- If the Quickflow is already deployed on a machine, you need to deploy it again after you update the security settings. If secured web site is configured, then you cannot change the server configuration for Quickflow in Central Configuration. However, if the **Secure Web Sites** checkbox is not selected, you can make the changes to the server configuration.
- Advance Server communication cannot be configured on HTTP/HTTPS.
- For resources of Load Balanced Server, ensure to configure all services with the same protocol (HTTP or HTTPS).
- It is recommended to use Standard SSL certificate instead of self-signed SSL certificate.

Using ASB Certificate for Secured Communication

ASB Certificate is a PCS-managed SSL certificate and renews automatically every 30 days when the certificate is still valid. The certificate is valid for two years. The PCS watchdog service on the runtime node checks the validity of the SSL certificate within 5 minutes from when it starts and once a day. PCS also renews the SSL certificate if it is expired.

If the runtime node is offline and later back online, the PCS watchdog service regularly checks the certificate validity and renews it if it is required. Consider restarting the PCS watchdog service to enforce the SSL certificate validity check within 5 minutes from the restarting time.

Furthermore, when the SSL certificate is renewed, PCS binds it to port 443 (by default) and changes the SslThumbprint PCS registry value. If AVEVA Work Tasks sites are hosted on port 443, certificate binding will be auto-renewed when the SSL certificate is renewed. It is recommended to check the bindings in IIS with every certificate renewal to ensure the application is running seamlessly.

To disable the certificate renewal, consider disabling the **ArchestrA Certificate Renewal Service** windows service.

Note: Client Service will try to automatically bind the renewed certificate. If there are any issues in browsing the website or starting the services after certificate renewal, users need to reconfigure the secure communication in Farm Configuration. To reconfigure, see the steps mentioned in above section *To define the Security Settings*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Securing Communication between Services

To have a secure communication between the services, apply the below changes to the Windows registry and reboot the server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
```

For more information on securing communication, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client and https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2015/2960358.

Define Services and Global Settings

You can define the Services and Global Settings using the Farm Configuration Wizard or through Configuration XMLs.

To define Services

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.

   Note: You can also customize the settings for these services and websites using the **Custom Settings** option. For more information about Service-related changes and Website-related changes, see Advanced Settings and Administering Web Servers respectively.

2. Click **Next**.

   The Global Settings dialog box appears.

3. In the **Communication** tab, enter key fields in **Email Settings** and **Exception Email settings** to configure exception e-mails.

   The following are the details:

   - **Host**: Enter information related to mail server.
     Example: Enter details related to Microsoft Exchange Server (IP address of the exchange server).
   - **Port**: Enter the port number for the mail server. The default port number is 25.
   - **From email**: Enter the from e-mail address.
     Example: admin@telematics.com

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **To email**: Enter the to e-mail address.
  Example: alex@telematics.com

4. Click **OK**.

Note: To confirm the exception e-mail settings, click **Test SMTP**. An e-mail will be generated and sent to the **To email** address confirming successful configuration of the e-mail. The advanced settings like **Secure Socket Layer (SSL), Email Timeout, User Name,** and **Password** for the e-mail account holder, **Password Encrypted** flag will differ based on the mail server settings.

Define Central Configuration

1. Run **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Farm Configuration Wizard window appears.
2. Click **Next**.
3. The Edit Farm settings window appears.
4. Select **Service Settings**.
5. Click **Next**.

   The Services/Global Settings window appears.
6. Click against **Central Configuration** to edit AVEVA Work Tasks **Central Configuration settings**.

   The Central Configuration Settings page appears.
   - **http/https**: This option enables you to determine if you want a secured (https) or a non-secured site (http).
   - **Authentication Type**: This option allows you to provide the authentication type to access the Central Configuration. The available authentication types are Windows, AVEVA Connect, and Azure Active Directory.

     Note: The AVEVA Connect and Azure Active Directory authentication types are available only if these user providers are configured in the system.
     To configure AVEVA Connect user provider, see AVEVA Connect Settings.
     To configure Azure Active Directory user provider see Azure Active Directory Settings.

   - **Create at**: This field allows you to provide the Internet Information Service (IIS) path where the new site is created.
   - **Port Number**: This field is used to provide a port number for the Central Configuration.
7. Click against **Create at** to change the path of the Central Configuration.

   A message indicating that if the path has been changed the Central Configuration will no longer be accessible appears.
8. Click **Yes**.

   The IIS structure screen is displayed. Select the desired IIS path for the Central Configuration.
9. Click **OK**.

   The selected URL appears in the **Central Configuration Settings** screen.
10. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:

- The default IIS path for Central Configuration is displayed. Click the edit icon to change the site path.

- If you want to update the authentication type to AVEVA Connect or Azure Active Directory of the already configured Central Configuration, you need to reconfigure the AVEVA Connect and Azure Active Directory user providers for the Central Configuration. For information on configuring user providers, see AVEVA Connect Settings and Azure Active Directory Settings.

- Central Configuration cannot be accessed in Chrome browser when configured with unsecured access (http://) for Azure Active Directory and AVEVA Connect authentication types.

Provision Services

While the installation is in progress, you can view the services that will be installed. This provides information on the progress of installation. At the end of the installation, a status indicating the completion of installation appears.

- Click **OK** to complete the installation.

AVEVA Work Tasks Client

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Select Client Installation

1. In the Select the type of installation page, select **Client**.
2. Click **Next**.

   The Database User Settings page appears.

Connect to an Existing Farm Database

You can use the Connect to existing database dialog box to connect to the existing Farm. You can select the Connection String or Connection provider for the new database.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Select the computer name on which the Farm Database is installed from the **Server Name** field.
2. Select **Connection String**.
3. Enter the following details:
   a. **Authentication Type**: This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports both Windows and SQL server authentication process. Example: Windows.
   b. **User ID**: This field is used to enter the SQL user ID for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   c. **Password**: This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   d. **Advanced Settings**: This is an optional field. It is used to enter any additional connection information such as connection Timeout etc. Example: Connection Timeout = 20.

      While creating a data source for repository, use the following custom settings to set properties for the connection pool.

      encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;

      encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

      enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks controls the transaction from escalating to a distributed transaction which is very expensive.

      The above property can be used to turn off this feature. The default value is set as True.

      connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

      maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

      sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

      Note: These settings are case sensitive and should be mentioned only in lower case.
   e. **Database**: This field is used to select the machine with database installed. Example: FarmDB.

      AVEVA Work Tasks will validate if at least one server is installed.
4. Click **Test Connection** to ensure database connectivity.

   You must have the following permissions on the Database Server:

   SELECT, INSERT, UPDATE, and DELETE permissions.

   EXECUTE permission for Stored Procedures.
5. Click **Next** to continue with the wizard.
6. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Define and Provision the Enterprise Console

You can define the required AVEVA Work Tasks services.

Steps to Define Enterprise Console

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.
2. Click **Next**.

   The **Installing Services** screen appears, indicating the progress of installation.

   A message *Installation is Complete* appears after all the services are installed.
3. Click **OK**.

Create a Repository DataSource

A **Repository DataSource** is used to store all repository related data. The **Repository DataSource** must be created on a machine where the **Central Configuration** site is installed.

Example

A TelematicsDataSource is created for TelematicsHR Repository.

A Repository contains Workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating the repository.

To create a Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**.

   The Central Configuration web page appears.
2. Click **DataSource Settings**.

   The Datasource List page appears.
3. Click **Add**.

   The Manage Datasource page appears.
4. Enter the required details.
   a. **Name**: Enter a valid data source name. Example: *TelematicsDataSource*.
   b. **Description**: Enter a description for the data source. Example: *This data source is used to store connection details*.
   c. **Connection String**: Select this option to establish a connection with the database. The connection string contains fields that that defines the connection. The connection information is encrypted and stored.

      Server Name

      To use a database on the local machine, enter "localhost", ".", or the local computer's name for the **Server Name**.

      Note: Server Name can be machine name/IP address/SQL Server AlwaysON/SQL Server instance name. Instead of entering "local host", it is preferable to give the local machine name.

      If the **Database Type** is **SQL Server**, then specify an IPV6/IPV4 address for the **Server Name**.
      To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: *localhost,1455*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be *<server name>\SQLExpress*.

Authentication Type

Use this field to determine the process to validate a user who is trying to connect to the database. AVEVA Work Tasks uses Windows and SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. The **User ID** and **Password** field is disabled if Windows Authentication is selected. Example: *Windows*

Advanced Settings

Use this field to define advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection.
Example: *Connection Timeout = 20*
or
For defining maximum connections that can be opened in a connection pool.
Example: *max pool size = 250*
Default connection pool is set as 250. We recommend to enter a higher value on heavily loaded environment.
While creating a data source for a repository, use the following custom settings to set properties for the connection pool.
encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.
enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature, AVEVA Work Tasks controls the transaction from escalating to an expensive distributed transaction.
The above property can be used to turn off this feature. The default value is set as *True*.
connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is 30 seconds. The value should be set in seconds.
maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is infinite.
sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.
These settings are case sensitive and should be mentioned only in lower case.

Database:

Specify or select a database that can be used as a repository database. This database stores all repository related data. The repository database is created if it does not exist on the database server instance.
Example: *TelematicsDB*

Enable Table Partitioning:

Select to enable database table partitioning. For more information, see Partitioning Database.

d. **Connection Provider**: Use this to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The selected dll must be copied to all the machines, all servers and all clients, that are connected to the current Farm.

Assembly Path

Click to browse and choose the required dll file.
To use custom properties (enableltm, connectionretrywaittime, maxconnectionretries, and sqlcommandtimeout) in the Assembly connection string, the database must be created before.
See *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Developer Guide* for creating the assembly. Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.

   A message about successful test connection appears.

6. Click **OK**.

   A message about successful creation of the data source appears.The message also states that the required tables will be created in the data source after a repository is associated with the data source.

When a data source is saved, an entry is created for the data source in the **Farm Database**. When a first repository is created for the data source, AVEVA Work Tasks tables are added to the data source. While creating a data source, the archival database is created with the same database name suffixed by *_BI*. This database is used for archival in future use.

Note: After you configure the AVEVA Work Tasks, you must add the created **Farm Database** to the **SQL Server AlwaysOn Availability Groups**. Contact your Database Administrator to add the database to **SQL Server AlwaysOn Availability Groups**.

To view the data source created, click **DataSource Settings** link.

Create a Repository

A **Repository** contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

To create a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The Central Configuration page appears.

2. Click **Repository Settings**.

   The Repository List page appears.

3. Click **Add**.

   Note: If the number of repositories exceeds the permitted limit for the license, then the message "You have exceeded the maximum number of repositories" appears, and the repository creation is stopped. Click **Close**.

   The **Manage Repository** page appears.

4. Enter the required details.
5. Click **OK**.

   The **Config Administrator** page appears.

6. Click to configure an administrator.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The User Lookup dialog box appears.

7. Select the provider from the provider drop-down list.
8. Enter the details to search the administrator for the repository that is created.

   Note: You can also search a user based on *ObjectSID* and *objectGUID* (applicable only for Active Directory provider).

9. Click **Search**.

   The user is listed under **Available Resources**.

10. Select and click the arrow to list the user under **Selected Resource**.
11. Click **Update**.

    The selected user is displayed in **User Lookup**.

12. Click **OK**.

    A message confirming the successful creation of an administrator appears.

13. Click **OK**.

    The **Repository List** page appears.

    For more information about configuring the system user, see Configure System User.

Manage Repository Window Fields

The following table explains the fields of the Manage Repository window:

Field Name	Description	Example
Repository DataSource	Lists all the data sources created in a Farm. The selected data source will be used to store data for the current repository.	TelematicsDatasource
Attach DataSource	Select Attach DataSource checkbox and then select a repository from the selected DataSource .Data associated with the selected repository will be copied to the new repository. The version of the installed product and destination DataSource should be the same as that of the DataSource from which the repository data is to be copied. For example, in an upgraded environment, from build Work Tasks version 1 to Work Tasks version 2, you cannot attach a DataSource created in build Work Tasks version 1 to build Work Tasks version 2.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	To match the build version of the installed product to the version of the DataSource: 1. If the DataSource was created in build Work Tasks version 1, then the entire build Work Tasks version 1 must be upgraded to Work Tasks version 2 by running the product installer in the upgrade mode. 2. All the Farm, DataSources, Repositories, and Lists of build Work Tasks version 1 must be upgraded to build Work Tasks version 2 by running the AVEVAWorkTasksUpgradeManager.exe located at the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder. 3. After upgrading all the AVEVA Work Tasks components to build Work Tasks version 2 build, that DataSource can be used in any machine that has same version by using this feature.	
Repository Name	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the repository name.	Telematics Repository
Description	Provide a description for the repository.	This stores Telematics Employee Leave Details.
From mail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might	Provide the valid email id that is specified in the **User Name** field while configuring **SMTP Settings** in the **Mail Server Settings** page. For more details, see Mail Server Setting.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	occur in the Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be sent to the mail ID defined in **Exception to mail**. The From address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To avoid this, restart the Workflow Engine Service and Communication Service.	
Exception to mail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails is sent to the email ID specified in **Exception to mail**.	Provide the valid email id.
Alert initiator of the Workflow	Determines if the status of a workflow should be sent as a notification mail to the initiator of the workflow.	A process with a rule to check the pending leave available for the employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
Servers	Lists the servers available in the Farm. Repository will be available only to the mapped servers.	Work Tasks Server Name
Enterprise Consoles	Lists all the Enterprise Consoles for a selected AVEVA Work Tasks Server. Repository will be available only to the mapped Enterprise Console.	http://worktaskservername:8001/EnterpriseConsole
Enable Theme	Select this checkbox to enable the defined theme.	-
Theme Name	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. It provides **Default** and **Lite** themes.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Resource Provider	Lists the various default resource providers supported by AVEVA Work Tasks. Resource providers are also termed as User Provider. A User Provider is a data store containing information about the user. AVEVA Work Tasks also supports custom resource provider. For more information about Resource Provider, see User Providers. AVEVA Connect and Azure Active Directory resource providers are also available after you configure them. To configure these providers, see AVEVA Connect Settings and Azure Active Directory Settings. AVEVA Work Tasks allows secured LDAP configuration. The default port number for LDAPS is 636. You can map the repository in IDE and use the Secured LDAP. The logged-in Active Directory is displayed by default, and **Active Directory Datasource** expects **LDAP://** or **Fully Qualified Domain Name** as value.	AVEVA Work Tasks Lists or Active Directory
Custom Application Folder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with a semicolon.	-
Hide Repository	Select this option to make the repository unavailable to the Farm.	-
Source path	Enter the path to the Source Elements Folder.	-
Attachment Storage Path	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	absolute path. - The default path is the **Attachments** folder in *C:\ProgramData\AVEVA Work Tasks*\. - The default path is used when no path is specified. - Ensure appropriate rights are set to the folder, where attachments are stored, for the following identities: -- The application pool identity for **Central Configuration** and **Enterprise Console(s)**. -- The account configured for the **Workflow Engine** service. - In case of mapped drives, ensure they are available to the identities. - In multiple-server environments, the path specified must be in a shared location.	

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Auto Log Off Settings for Enterprise Console	Automatically logs out the logged in user after the designated period of inactivity. Enter the time in minutes. This field accepts only valid integer values between 0 to 360. 0 minute means the user will not get automatically logged off. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. The default value for automatic log off is 30 minutes. You use this field to define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML.	0 to 360 (minutes)
Auto Log Off Settings for AVEVA Work Tasks Pro	Request for user credentials for AVEVA Work Tasks Pro app after the set time is elapsed. Enter the time in minutes. This field accepts only valid integer values between 0 to 2880. 0 minute means the user will not get automatically logged off. The default value is 30 minutes. If AVEVA Work Tasks Pro app is offline, the Auto Log Off will not get active, and the user will not be automatically logged off as per the defined settings.	0 to 2880 (in minutes)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Offline Support	Select this option to enable offline submission of items in AVEVA Work Tasks Pro app.	-
Refresh on Work Item submission	Select this option to enable Inbox refresh on Work Item submission in AVEVA Work Tasks Pro . Refresh option is enabled by default.	-
Timeout	Enter the time in seconds to set the timeout. If the value is set as 0, then there is no timeout. Default value is 60 seconds.	-
Label for Fill Form	Enter the label for Fill Form. The customized label will be displayed for the Fill Form in AVEVA Work Tasks Pro.	Work Tasks
Date Configuration Flag for Alternate Actors	Select the **Enable Date Settings** option to set the **From** date and **To** date as mandatory while creating alternate actors. For more information about creating alternate actors, see *Alternate Actors* topic in the *User Guide*.	-
Display Domain name with User Name	Select this checkbox to display the domain name along with the user name in the Enterprise Console and the List. The domain name will be displayed only: • If this checkbox is selected. • If the **Resource Provider** is selected as **Active Directory**.	-

User Providers

AVEVA Work Tasks supports the following User Providers by default:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Active Directory

When you select **Active Directory** as the **Resource Provider**, details such as **Active Directory datasource** and the **Active Directory search base** are populated by default.

Authentication User, **Authentication Password**, **User Log-on ID**, and **User Log-on Display Name** are optional fields. Access to the active directory might not be available for all users. In such cases, provide the active directory authentication user name and the active directory authentication password with which you can connect to the Active Directory.

AVEVA Work Tasks considers *SAM-Account-Name* as **User Log-on ID** by default. If you want to use active directory attributes other than the *SAM-Account-Name* user, then enter the desired active directory attribute details in the **User Log-on ID** field and **User Log-on Display Name** field. **User Log-on Display Name** is the user friendly display name of the active directory attribute. This **User Log-on Display Name** is referred in the User Lookup dialog box against the **User Log-on ID**.

Steps

1. When you select **Active Directory** as the **Resource Provider**, then the **Use Token Service** field appears. In this field, select one of the following:
   - If you do not want to use AIM authentication, select **Without Token Service**.
   - If you want to use AIM authentication, then select a server from the list of alias server names.
2. Enter **Active Directory datasource**.
3. Enter **Active Directory search base**.
4. Enter **Domain Name**.

   The domain name is case-sensitive.
5. Enter **Authentication User**.
6. Enter **Authentication Password**.
7. Check **Is Cross Domain Role Provider** checkbox, if cross domain users and roles mapping is required.

   Example: Users *John* and *Jim* of domain *TelematicsU* and users *Mike* and *Bob* from domain *TelematicsAU* are added to role Team HR of domain *TelematicsMain*.
8. Enter **User Log-on ID** if you want to use another Active Directory attributes instead of *SAM-Account-Name*.

   We recommend to use attributes of type *string* as input for **User Log-on ID.** However, *ObjectSID* and *objectGUID* are also supported.

   After you set the **User Log-on ID**, you cannot change it later.
9. Enter the **User Log-on Display Name**.

   After you set the **User Log-on Display Name**, you cannot change it later.

   This is the default display name of the user account of the active directory user used. **User Log-on Display Name** is used in user lookups.
10. Click **Test LDAP**.

    A message about successful LDAP test appears.

Note:
- Some of the active directory attributes or properties are not indexed. If you use an active directory attribute that is not indexed, it might impact the log-on validation performance based on the active directory entries. We recommend to use indexed attributes as **User Log-on ID** or index the attributes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- For an upgraded repository, you cannot change both **User Log-on ID** and **User Log-on Display Name**.
- You cannot localize **User Log-on ID** and **User log-on Display Name** fields.
- Disabled users will not be shown in user lookup search.

Using Forms

For Forms Resource Provider:

- If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.

1. In the **Resource Provider** drop-down list**,** select **Forms.**
2. In the **User Entity properties** field, enter XML data.

   Click **Sample Code** to get the XML Schema Template.

   Example

   ```
   <properties>
   <property name="userid" type="string" purpose="identifier" />
   <property name="username" type="string" purpose="name" />
   <property name="pwd" type="string" purpose="password" />
   </properties>
   ```
3. In the **User Provider Properties** field, enter the same XML Schema.

   There must be at least one identifier in the XML Schema. For example, UserID
4. In the **Datasource** field, enter a valid data source or specify a dll.

   The value for the data source can be specified as **Physical path of the dll** or **Physical path of the dll::param. Param** is not mandatory.

   See topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Build Business Process* section for creating the assembly.

   Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly, whereas one class can return multiple connection strings.

   Examples of data source parameter:

   - c:\DLLStore\GetConnectionString.dll
   - c:\GetConnectionString.dll::Leave

   If there are two data sources, *Leave* and *Licensing*, and only one DLL is used to return a connection string based on the data source selected. The parameter can be specified as **c:\GetConnectionString.dll::Leave** or **c:\ GetConnectionString.dll::Licensing**. The connection string can be returned from the connection dll based on this parameter.
5. Enter a proper table name or view name which holds the User Name and Password for the Authentication purpose.

   Example: Passmast (the table which is available in IMS_NEW Database)
6. Click **Test SQL Connection**.

   A message about successful test connection appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using ASP.NET Membership

This resource provider lists users defined in the asp.net membership database.

Lists

This resource provider lists users created using AVEVA Work Tasks Users Creation. If you are unable to determine a User Provider to create a repository, use AVEVA Work Tasks Lists.

To verify successful creation of the Repository:

1. Click **Enterprise Console** URL from the site path column.
   The repository created is listed.
   Add the created Enterprise Console link into "Add to Favorites" to easily navigate in future.
2. Click **Repository User Provider** link to logon to the repository.
   The Enterprise Console log-on page appears.
3. Enter the logon details, and then click **Login**.
   The Enterprise Console home page appears. This indicates successful creation of a Repository.
   All features can only be available for a user with administrative privileges.

Note: Consider you created the Repository and then deleted it. If you create the same Repository again immediately and try to log into it through Enterprise Console, the Provider does not list the Repository. You need to access IIS and recycle AVEVA Work Tasks from the Application Pool to get the Repository listed in the Provider list..

Verify the Installation

After you install AVEVA Work Tasks, verify that the product is installed successfully. To verify successful installation, create a simple Workflow using Process Designer. The tasks includes the following:

- Creating a Workflow
- Publishing the Workflow
- Testing the Workflow

Creating a Workflow

1. Log on to the **Enterprise Console**.
2. If the Repository is configured for Active Directory provider, go to step 4.
3. Enter the **User Name** and **Password**.
   The **User Name** and **Password** will be validated with the user provider selected while creating the repository.
   The default authentication provider is the Repository User Provider List.
4. In the **Enterprise Console**, click menu icon.
   All features can only be available for a user with administrative privileges.
5. From the menu, select **Workflows**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Workflow window appears.

6. On the Action bar, click **New**, and then click **Workflow**.
   The New Workflow window appears.
7. Enter **Title** and **Description**.
8. Click **Save & Continue**.
   The successful message appears.
9. Select **Design the Workflow now**.
   This is the default option.
10. Click **Finish**.
    The Process Designer window appears.
11. From the activities pane on the left, drag the Information activity to the Process Designer window.
12. The activity *Information1* is created.
    Click to create a link between two activities.
    The Link Action window appears.
    a. Select from the **From Action** drop-down box.
    b. Select from the **To Action** drop-down box.
    c. Click **OK**.
       The link is established.
13. Right-click *Information1*.
14. Select **Activity Properties** from the context menu.
    The Activity Properties pane appears.
    a. Enter **Name** and **Description**.
    b. Click to **Assign Actor(s)**.
       The Assign Actor(s) window appears.
    c. Select the required provider from the **Providers** drop-down list.
    d. Select values from the drop-down list.
    e. Click **Add**.
    f. Click **Update**.
15. Click to enter the **Subject** for Notification Contents.
    The window to enter subject for notification appears.
16. Click **Update**.
17. Click to enter the **Body** for Notification Contents.
    The window to enter the Body content appears.
18. Click **Update**.
19. From the Activity Properties pane, click **Save**.

Publishing the Workflow

1. On the **Publish** menu, click **Publish**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Published successful message appears.

2. Close the dialog box.

   The Enterprise Console screen lists the workflow created.

Testing the Workflow

1. Select the listed workflow.
2. Right-click the selected workflow.
3. Click **Publish and Test Run**.

   The **Status Finished successfully** appears.

   If Status is **Execution Pending** in the Process Execution window, then click on the **Refresh** button to get the status.
4. In the **Enterprise Console** menu, select **Inbox**.

   The Inbox lists the work item.
5. Click to view the work item listed.

   This indicates that the installation is successful.

Client and Server Installation

This section describes the activities involved while installing AVEVA Work Tasks on a AVEVA Work Tasks Client and AVEVA Work Tasks Server.

The installation for this scenario can be incorporated in any scenarios mentioned above wherein the Server and the Client is involved.

Ensure that you install the product in the following sequence:

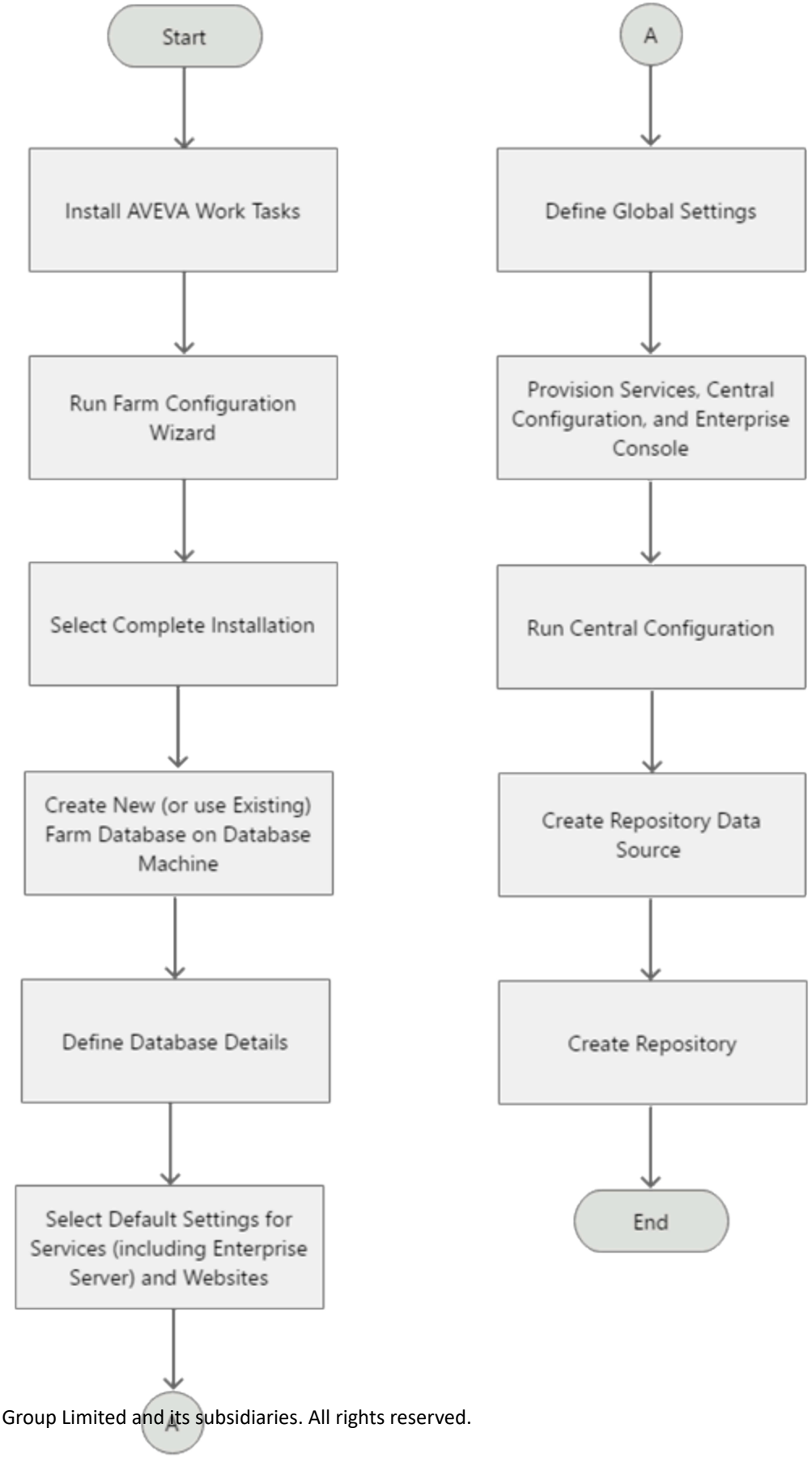
- High-Level View of Server Installation
- High-Level View of Client Installation

High-Level View of Server Installation

This section provides a high-level view of the various installation steps. The objective of this section is to provide an overview of the various activities involved while installing AVEVA Work Tasks Server. This section also enables you to visually keep track of your activities while following the sequence of tasks.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



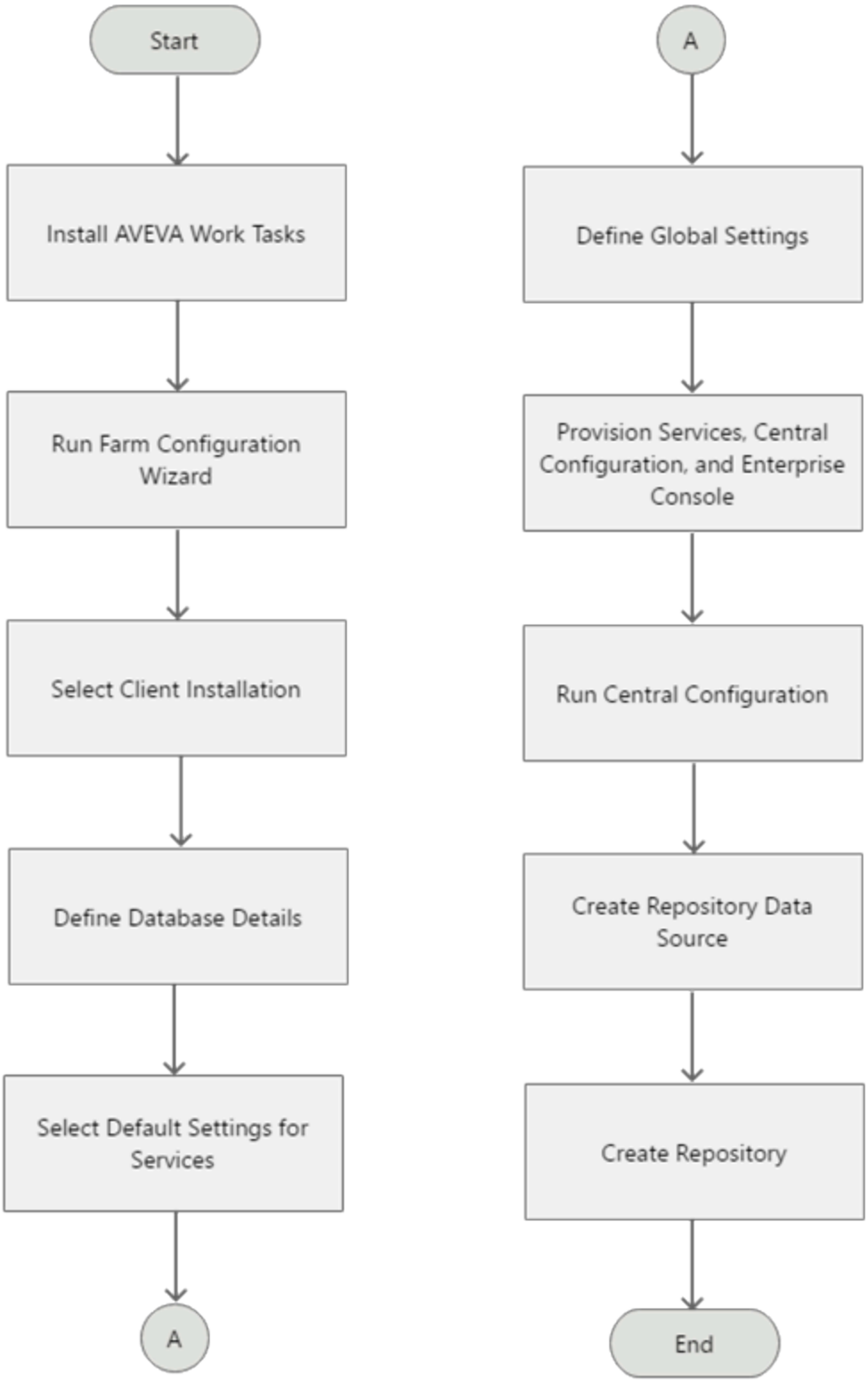
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

High-Level View of Client Installation

This section provides a high-level view of AVEVA Work Tasks Client installation. The Client components are installed on Web Servers.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation

This section provides installation steps for the scenario AVEVA Work Tasks Client and AVEVA Work Tasks Server Installation.

The installation includes the following:

- Installing AVEVA Work Tasks Server
- Installing AVEVA Work Tasks Client

The Server has to be installed before installing the Client.

AVEVA Work Tasks Server

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
4. Click **Next**.

   The License Server details are configured successfully.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.
4. Click **Next**.

   The **Edit Farm Settings** window appears.
5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.
6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
9. Click **Next**.

   The License Server details are configured successfully.

Select the Type of Installation - Enterprise Edition

Following points must be considered while installing AVEVA Work Tasks on the Enterprise Server:

- Ensure that you select the **Complete Installation** and **Enable load balancing** options.
- You can either create a new Farm database or use an existing one.

1. In the Select the type of installation page, select **Complete Installation (Server)**.

   A Complete Standalone Deployment does not require a client installation.
2. Select the **Enable load balancing (This feature requires and Enterprise license.)** check box for installing AVEVA Work Tasks on the Enterprise Server.

   The **Farm Configuration Wizard** dialog box appears.
3. Click **OK**.
4. To create a new Farm database, click **Create AVEVA Work Tasks Farm**, and then select one of the following:
   - **Create AVEVA Work Tasks Farm database**
   - **Use existing AVEVA Work Tasks Farm database (Creates AVEVA Work Tasks Farm tables only if they do not exist)**
5. To connect to an existing Farm database, click **Connect to existing AVEVA Work Tasks Farm (if the first server**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

of the Enterprise Server is already installed).

6. Click **Next**.

   The Database User Settings page appears.

Create Farm Database

Use the Create AVEVA Work Tasks Farm Database page to provide **New Database** details. Select the **Connection String** option to establish a connection with the database. This connection information is encrypted and stored.

You can enter either Windows or Microsoft SQL Server user settings for authentication.

Note:
- The Farm database password does not support **&** and **;** special characters.
- While setting up **Farm Database** with **Windows Authentication**, do not enter **FQDN** in the **Domain Name** field.

To Create Farm Database

1. Enter **Database Title**.
2. Select **Connection String**.

   See *Connection Provider* later in this topic, if you have selected the **Connection Provider** option.

   - Enter the following details:

     Server Name:

     Provide information about the database server on which the Farm database can be installed.

     If the **Database Type** is **SQL Server**, then you can specify an IPv6 address for the **Server Name**.

     If SQLExpress Instance need to be used as a Database instance of the Farm database, then **Server Name** should be <server name>\SQLExpress.

     To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

   - **Authentication Type**

     This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports Windows and SQL Server authentication process.

     Example: Windows
     If Windows Authentication is selected, the user configured for the Farm configuration will be used to access the database. This is the authentication recommended by AVEVA Work Tasks.

   - **Login**

     This field is used to enter the SQL login credential for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Password**

     This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.

   - **Advanced Settings**

     This is an optional field. It is used to enter any additional connection information, such as connection Timeout. Example: Connection Timeout = 20.

     While creating a data source for repository, use the following custom settings to set properties for

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

connection pool. These settings are case sensitive and should be mentioned only in lower case.

```
encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10;
sqlcommandtimeout=300;
```

encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

enableltm: AVEVA Work Tasks internally implements light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive. This property can be used to turn off this feature. The default value is set as True.

connectionretrywaittime: While opening a new connection, if the data base server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

maxconnectionretries: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

3. Click **Test Connection** to ensure database connectivity.
4. Click **Next**.

   The message *AVEVA Work Tasks Farm is Created* appears.
5. Click **OK**.

AlwaysOn Availability Groups feature of Microsoft SQL Server

The **AlwaysOn Availability Groups** feature of Microsoft SQL Server is a high-availability and disaster-recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in Microsoft SQL Server 2012, AlwaysOn Availability Groups maximizes the availability of a set of user databases for an enterprise.

After you configure the AVEVA Work Tasks, you must add the created Farm Database to the SQL Server AlwaysOn Availability Groups. Contact your Database Administrator to add the database to SQL Server AlwaysOn Availability Groups.

Notes:

- The authentication user needs different rights assigned for different scenarios. If a user wants to create database (Farm or Repository Data Source Database), Authentication User needs to have minimum 'dbcreator' access rights on Database Server Instance.

- If database is already created, user needs to have following permissions on Database Server:

- SELECT, INSERT, UPDATE, and DELETE permissions.

- EXECUTE permission for Stored Procedures.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To choose Connection Provider

Complete the following steps to choose a connection provider:

1. Enter **Database Title**.
2. In the **Database Type** box, select a database type.
3. Select **Connection Provider**.

   This option is used to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.
4. Enter the following details:
   - **Assembly Path**:

     Click the icon to browse and choose the required dll file.

     If custom properties (enableltm, connectionretrywaittime, maxconnectionretries and sqlcommandtimeout) are required to be used in Assembly connection string, then user should select **Use existing Farm database** while creating a the Farm. See Creating or Using an Existing Database.

     See the topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub chapter in the *Build Business Process* section for creating the assembly. Make sure that, there should be only one class implementing *IDBConnectionString* interface within this assembly.
5. Click **Test Connection** to ensure database connectivity.
6. Click **Next**.
7. The message *AVEVA Work Tasks Farm is created* appears.
8. Click **OK**.

Define Security Settings

You can define the Security Settings using the Farm Configuration Wizard.

For more information on Security settings, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client.

To define the Security Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Security Settings** and **Service Settings** and then, click **Next**.
4. Select **Secure Communication** to establish secured communication between the AVEVA Work Tasks services through HTTPS.
5. Select **Secure Web Sites** check box to configure secured web sites. Uncheck the **Secure Web Sites** check box to use HTTP sites. By default the sites are created in HTTPS mode.
6. Enter the Port number. The default port for the HTTPS site is 8001, and for the HTTP site is 8000. You can enter any port number that IIS accepts. But if any other services or applications are already using the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

entered port number, AVEVA Work Tasks will not start.

7. Select the available certificate from the **Certificate** list. If your certificate is not listed, click the **Refresh** icon to repopulate the list.

   Note: If the required certificate is not listed, generate one using the System Management Server (Configurator tool). Do not use a self-signed certificate without a proper root chain.

8. Click the **Validate** button to validate the certificate before configuring the security settings.

   Note: An Internet connection is needed to validate the certificate.

9. Click **Configure**, then **Next**, and then **Finish** to apply the new security settings.

Note:
- To update the port number, certificate, and websites from unsecure to secure or vice-versa, select **Security Settings** and **Service Settings** check box in **Edit AVEVA Work Task Farm Settings** window before updating the values. After the required details are provided, these settings get updated only when you click **Configure**, then **Next**, and then **Finish** as you navigate through the Farm Configuration Wizard..
- If the Quickflow is already deployed on a machine, you need to deploy it again after you update the security settings. If secured web site is configured, then you cannot change the server configuration for Quickflow in Central Configuration. However, if the **Secure Web Sites** checkbox is not selected, you can make the changes to the server configuration.
- Advance Server communication cannot be configured on HTTP/HTTPS.
- For resources of Load Balanced Server, ensure to configure all services with the same protocol (HTTP or HTTPS).
- It is recommended to use Standard SSL certificate instead of self-signed SSL certificate.

Using ASB Certificate for Secured Communication

ASB Certificate is a PCS-managed SSL certificate and renews automatically every 30 days when the certificate is still valid. The certificate is valid for two years. The PCS watchdog service on the runtime node checks the validity of the SSL certificate within 5 minutes from when it starts and once a day. PCS also renews the SSL certificate if it is expired.

If the runtime node is offline and later back online, the PCS watchdog service regularly checks the certificate validity and renews it if it is required. Consider restarting the PCS watchdog service to enforce the SSL certificate validity check within 5 minutes from the restarting time.

Furthermore, when the SSL certificate is renewed, PCS binds it to port 443 (by default) and changes the SslThumbprint PCS registry value. If AVEVA Work Tasks sites are hosted on port 443, certificate binding will be auto-renewed when the SSL certificate is renewed. It is recommended to check the bindings in IIS with every certificate renewal to ensure the application is running seamlessly.

To disable the certificate renewal, consider disabling the **ArchestrA Certificate Renewal Service** windows service.

Note: Client Service will try to automatically bind the renewed certificate. If there are any issues in browsing the website or starting the services after certificate renewal, users need to reconfigure the secure communication in Farm Configuration. To reconfigure, see the steps mentioned in above section *To define the Security Settings*.

Securing Communication between Services

To have a secure communication between the services, apply the below changes to the Windows registry and reboot the server:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v2.0.50727]
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v2.0.50727]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
[HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]
"SystemDefaultTlsVersions" = dword:00000001
"SchUseStrongCrypto" = dword:00000001
```

For more information on securing communication, see https://docs.microsoft.com/en-us/mem/configmgr/core/plan-design/security/enable-tls-1-2-client and https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2015/2960358.

Define Services and Global Settings

You can define the Services and Global Settings using the Farm Configuration Wizard or through Configuration XMLs.

To define Services

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.

   Note: You can also customize the settings for these services and websites using the **Custom Settings** option. For more information about Service-related changes and Website-related changes, see Advanced Settings and Administering Web Servers respectively.

2. Click **Next**.

   The Global Settings dialog box appears.

3. In the **Communication** tab, enter key fields in **Email Settings** and **Exception Email settings** to configure exception e-mails.

   The following are the details:

   - **Host**: Enter information related to mail server.
     Example: Enter details related to Microsoft Exchange Server (IP address of the exchange server).
   - **Port**: Enter the port number for the mail server. The default port number is 25.
   - **From email**: Enter the from e-mail address.
     Example: admin@telematics.com
   - **To email**: Enter the to e-mail address.
     Example: alex@telematics.com

4. Click **OK**.

Note: To confirm the exception e-mail settings, click **Test SMTP**. An e-mail will be generated and sent to the **To**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

email address confirming successful configuration of the e-mail. The advanced settings like **Secure Socket Layer (SSL), Email Timeout, User Name,** and **Password** for the e-mail account holder, **Password Encrypted** flag will differ based on the mail server settings.

Define Central Configuration

1. Run **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Farm Configuration Wizard window appears.
2. Click **Next**.
3. The Edit Farm settings window appears.
4. Select **Service Settings**.
5. Click **Next**.

   The Services/Global Settings window appears.
6. Click against **Central Configuration** to edit AVEVA Work Tasks **Central Configuration settings**.

   The Central Configuration Settings page appears.
   - **http/https**: This option enables you to determine if you want a secured (https) or a non-secured site (http).
   - **Authentication Type**: This option allows you to provide the authentication type to access the Central Configuration. The available authentication types are Windows, AVEVA Connect, and Azure Active Directory.

     Note: The AVEVA Connect and Azure Active Directory authentication types are available only if these user providers are configured in the system.
     To configure AVEVA Connect user provider, see AVEVA Connect Settings.
     To configure Azure Active Directory user provider see Azure Active Directory Settings.
   - **Create at**: This field allows you to provide the Internet Information Service (IIS) path where the new site is created.
   - **Port Number**: This field is used to provide a port number for the Central Configuration.
7. Click against **Create at** to change the path of the Central Configuration.

   A message indicating that if the path has been changed the Central Configuration will no longer be accessible appears.
8. Click **Yes**.

   The IIS structure screen is displayed. Select the desired IIS path for the Central Configuration.
9. Click **OK**.

   The selected URL appears in the **Central Configuration Settings** screen.
10. Click **OK**.

Note:

- The default IIS path for Central Configuration is displayed. Click the edit icon to change the site path.

- If you want to update the authentication type to AVEVA Connect or Azure Active Directory of the already

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

configured Central Configuration, you need to reconfigure the AVEVA Connect and Azure Active Directory user providers for the Central Configuration. For information on configuring user providers, see AVEVA Connect Settings and Azure Active Directory Settings.

- Central Configuration cannot be accessed in Chrome browser when configured with unsecured access (http://) for Azure Active Directory and AVEVA Connect authentication types.

Provision Services

While the installation is in progress, you can view the services that will be installed. This provides information on the progress of installation. At the end of the installation, a status indicating the completion of installation appears.

- Click **OK** to complete the installation.

AVEVA Work Tasks Client

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Configure the License Server

You can configure the License Server details, such as License Server name and port.

To configure License Server

1. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
2. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
3. Click **Configure**.

   The confirmation messages appears in the **Configuration Message Log** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
4. Click **Next**.

   The License Server details are configured successfully.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Editing the License Server Details

- If the Farm Configuration Wizard is already open, click **Next** to continue with License Server configuration.

However, if the **Farm Configuration Wizard** is not open due to any reason, you can manually start the wizard. To manually start the wizard and then configure the License Server, complete the following steps:

1. Close the Farm Configuration Wizard window that is automatically opened.
2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then, select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.

   Note: The Farm Configuration Wizard should be run with administrative privileges.
4. Click **Next**.

   The **Edit Farm Settings** window appears.
5. In the **Edit Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box.
6. In the **License Server Name** box, type the name of the AVEVA Enterprise License Server. If you have already configured the license server, then the name of configured License Server is displayed.
7. In the **License Server Port** box, type the port number used by the AVEVA Enterprise License Server. Ensure that you enter a valid port number.
8. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the License Server details entered.
9. Click **Next**.

   The License Server details are configured successfully.

Select Client Installation

1. In the Select the type of installation page, select **Client**.
2. Click **Next**.

   The Database User Settings page appears.

Connect to an Existing Farm Database

You can use the Connect to existing database dialog box to connect to the existing Farm. You can select the Connection String or Connection provider for the new database.

1. Select the computer name on which the Farm Database is installed from the **Server Name** field.
2. Select **Connection String**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Enter the following details:
   a. **Authentication Type**: This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports both Windows and SQL server authentication process. Example: Windows.
   b. **User ID**: This field is used to enter the SQL user ID for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   c. **Password**: This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   d. **Advanced Settings**: This is an optional field. It is used to enter any additional connection information such as connection Timeout etc. Example: Connection Timeout = 20.

      While creating a data source for repository, use the following custom settings to set properties for the connection pool.

      encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;

      encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

      enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks controls the transaction from escalating to a distributed transaction which is very expensive.

      The above property can be used to turn off this feature. The default value is set as True.

      connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

      maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

      sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

      Note: These settings are case sensitive and should be mentioned only in lower case.
   e. **Database**: This field is used to select the machine with database installed. Example: FarmDB.

      AVEVA Work Tasks will validate if at least one server is installed.
4. Click **Test Connection** to ensure database connectivity.

   You must have the following permissions on the Database Server:

   SELECT, INSERT, UPDATE, and DELETE permissions.

   EXECUTE permission for Stored Procedures.
5. Click **Next** to continue with the wizard.
6. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Define and Provision the Enterprise Console

You can define the required AVEVA Work Tasks services.

Steps to Define Enterprise Console

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.
2. Click **Next**.

   The **Installing Services** screen appears, indicating the progress of installation.

   A message *Installation is Complete* appears after all the services are installed.
3. Click **OK**.

Create a Repository DataSource

A **Repository DataSource** is used to store all repository related data. The **Repository DataSource** must be created on a machine where the **Central Configuration** site is installed.

Example

A TelematicsDataSource is created for TelematicsHR Repository.

A Repository contains Workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating the repository.

To create a Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**.

   The Central Configuration web page appears.
2. Click **DataSource Settings**.

   The Datasource List page appears.
3. Click **Add**.

   The Manage Datasource page appears.
4. Enter the required details.
   a. **Name**: Enter a valid data source name. Example: *TelematicsDataSource*.
   b. **Description**: Enter a description for the data source. Example: *This data source is used to store connection details*.
   c. **Connection String**: Select this option to establish a connection with the database. The connection string contains fields that that defines the connection. The connection information is encrypted and stored.

      Server Name

      To use a database on the local machine, enter "localhost", ".", or the local computer's name for the **Server Name**.

      Note: Server Name can be machine name/IP address/SQL Server AlwaysON/SQL Server instance name. Instead of entering "local host", it is preferable to give the local machine name.

      If the **Database Type** is **SQL Server**, then specify an IPV6/IPV4 address for the **Server Name**.
      To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: *localhost,1455*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be *<server name>\SQLExpress*.

Authentication Type

Use this field to determine the process to validate a user who is trying to connect to the database. AVEVA Work Tasks uses Windows and SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. The **User ID** and **Password** field is disabled if Windows Authentication is selected. Example: *Windows*

Advanced Settings

Use this field to define advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection.
Example: *Connection Timeout = 20*
or
For defining maximum connections that can be opened in a connection pool.
Example: *max pool size = 250*
Default connection pool is set as 250. We recommend to enter a higher value on heavily loaded environment.
While creating a data source for a repository, use the following custom settings to set properties for the connection pool.
encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.
enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature, AVEVA Work Tasks controls the transaction from escalating to an expensive distributed transaction.
The above property can be used to turn off this feature. The default value is set as *True*.
connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is 30 seconds. The value should be set in seconds.
maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is infinite.
sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.
These settings are case sensitive and should be mentioned only in lower case.

Database:

Specify or select a database that can be used as a repository database. This database stores all repository related data. The repository database is created if it does not exist on the database server instance.
Example: *TelematicsDB*

Enable Table Partitioning:

Select to enable database table partitioning. For more information, see Partitioning Database.

d. **Connection Provider**: Use this to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The selected dll must be copied to all the machines, all servers and all clients, that are connected to the current Farm.

Assembly Path

Click to browse and choose the required dll file.
To use custom properties (enableltm, connectionretrywaittime, maxconnectionretries, and sqlcommandtimeout) in the Assembly connection string, the database must be created before.
See *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Developer Guide* for creating the assembly. Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.

   A message about successful test connection appears.

6. Click **OK**.

   A message about successful creation of the data source appears.The message also states that the required tables will be created in the data source after a repository is associated with the data source.

When a data source is saved, an entry is created for the data source in the **Farm Database**. When a first repository is created for the data source, AVEVA Work Tasks tables are added to the data source. While creating a data source, the archival database is created with the same database name suffixed by *_BI*. This database is used for archival in future use.

Note: After you configure the AVEVA Work Tasks, you must add the created **Farm Database** to the **SQL Server AlwaysOn Availability Groups**. Contact your Database Administrator to add the database to **SQL Server AlwaysOn Availability Groups**.

To view the data source created, click **DataSource Settings** link.

Create a Repository

A **Repository** contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

To create a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The Central Configuration page appears.

2. Click **Repository Settings**.

   The Repository List page appears.

3. Click **Add**.

   Note: If the number of repositories exceeds the permitted limit for the license, then the message "You have exceeded the maximum number of repositories" appears, and the repository creation is stopped. Click **Close**.

   The **Manage Repository** page appears.

4. Enter the required details.
5. Click **OK**.

   The **Config Administrator** page appears.

6. Click to configure an administrator.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The User Lookup dialog box appears.

7. Select the provider from the provider drop-down list.
8. Enter the details to search the administrator for the repository that is created.

   Note: You can also search a user based on *ObjectSID* and *objectGUID* (applicable only for Active Directory provider).

9. Click **Search**.

   The user is listed under **Available Resources**.

10. Select and click the arrow to list the user under **Selected Resource**.
11. Click **Update**.

    The selected user is displayed in **User Lookup**.

12. Click **OK**.

    A message confirming the successful creation of an administrator appears.

13. Click **OK**.

    The **Repository List** page appears.

    For more information about configuring the system user, see Configure System User.

Manage Repository Window Fields

The following table explains the fields of the Manage Repository window:

Field Name	Description	Example
Repository DataSource	Lists all the data sources created in a Farm. The selected data source will be used to store data for the current repository.	TelematicsDatasource
Attach DataSource	Select Attach DataSource checkbox and then select a repository from the selected DataSource .Data associated with the selected repository will be copied to the new repository. The version of the installed product and destination DataSource should be the same as that of the DataSource from which the repository data is to be copied. For example, in an upgraded environment, from build Work Tasks version 1 to Work Tasks version 2, you cannot attach a DataSource created in build Work Tasks version 1 to build Work Tasks version 2.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	To match the build version of the installed product to the version of the DataSource: 1. If the DataSource was created in build Work Tasks version 1, then the entire build Work Tasks version 1 must be upgraded to Work Tasks version 2 by running the product installer in the upgrade mode. 2. All the Farm, DataSources, Repositories, and Lists of build Work Tasks version 1 must be upgraded to build Work Tasks version 2 by running the AVEVAWorkTasksUpgradeManager.exe located at the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder. 3. After upgrading all the AVEVA Work Tasks components to build Work Tasks version 2 build, that DataSource can be used in any machine that has same version by using this feature.	
Repository Name	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the repository name.	Telematics Repository
Description	Provide a description for the repository.	This stores Telematics Employee Leave Details.
From mail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might	Provide the valid email id that is specified in the **User Name** field while configuring **SMTP Settings** in the **Mail Server Settings** page. For more details, see Mail Server Setting.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	occur in the Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be sent to the mail ID defined in **Exception to mail**. The From address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To avoid this, restart the Workflow Engine Service and Communication Service.	
Exception to mail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails is sent to the email ID specified in **Exception to mail**.	Provide the valid email id.
Alert initiator of the Workflow	Determines if the status of a workflow should be sent as a notification mail to the initiator of the workflow.	A process with a rule to check the pending leave available for the employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
Servers	Lists the servers available in the Farm. Repository will be available only to the mapped servers.	Work Tasks Server Name
Enterprise Consoles	Lists all the Enterprise Consoles for a selected AVEVA Work Tasks Server. Repository will be available only to the mapped Enterprise Console.	http://worktaskservername:8001/EnterpriseConsole
Enable Theme	Select this checkbox to enable the defined theme.	-
Theme Name	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. It provides **Default** and **Lite** themes.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Resource Provider	Lists the various default resource providers supported by AVEVA Work Tasks. Resource providers are also termed as User Provider. A User Provider is a data store containing information about the user. AVEVA Work Tasks also supports custom resource provider. For more information about Resource Provider, see User Providers. AVEVA Connect and Azure Active Directory resource providers are also available after you configure them. To configure these providers, see AVEVA Connect Settings and Azure Active Directory Settings. AVEVA Work Tasks allows secured LDAP configuration. The default port number for LDAPS is 636. You can map the repository in IDE and use the Secured LDAP. The logged-in Active Directory is displayed by default, and **Active Directory Datasource** expects **LDAP://** or **Fully Qualified Domain Name** as value.	AVEVA Work Tasks Lists or Active Directory
Custom Application Folder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with a semicolon.	-
Hide Repository	Select this option to make the repository unavailable to the Farm.	-
Source path	Enter the path to the Source Elements Folder.	-
Attachment Storage Path	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	absolute path. - The default path is the **Attachments** folder in *C:\ProgramData\AVEVA Work Tasks*. - The default path is used when no path is specified. - Ensure appropriate rights are set to the folder, where attachments are stored, for the following identities: -- The application pool identity for **Central Configuration** and **Enterprise Console(s)**. -- The account configured for the **Workflow Engine** service. - In case of mapped drives, ensure they are available to the identities. - In multiple-server environments, the path specified must be in a shared location.	

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Auto Log Off Settings for Enterprise Console	Automatically logs out the logged in user after the designated period of inactivity. Enter the time in minutes. This field accepts only valid integer values between 0 to 360. 0 minute means the user will not get automatically logged off. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. The default value for automatic log off is 30 minutes. You use this field to define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML.	0 to 360 (minutes)
Auto Log Off Settings for AVEVA Work Tasks Pro	Request for user credentials for AVEVA Work Tasks Pro app after the set time is elapsed. Enter the time in minutes. This field accepts only valid integer values between 0 to 2880. 0 minute means the user will not get automatically logged off. The default value is 30 minutes. If AVEVA Work Tasks Pro app is offline, the Auto Log Off will not get active, and the user will not be automatically logged off as per the defined settings.	0 to 2880 (in minutes)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
Offline Support	Select this option to enable offline submission of items in AVEVA Work Tasks Pro app.	-
Refresh on Work Item submission	Select this option to enable Inbox refresh on Work Item submission in AVEVA Work Tasks Pro . Refresh option is enabled by default.	-
Timeout	Enter the time in seconds to set the timeout. If the value is set as 0, then there is no timeout. Default value is 60 seconds.	-
Label for Fill Form	Enter the label for Fill Form. The customized label will be displayed for the Fill Form in AVEVA Work Tasks Pro.	Work Tasks
Date Configuration Flag for Alternate Actors	Select the **Enable Date Settings** option to set the **From** date and **To** date as mandatory while creating alternate actors. For more information about creating alternate actors, see *Alternate Actors* topic in the *User Guide*.	-
Display Domain name with User Name	Select this checkbox to display the domain name along with the user name in the Enterprise Console and the List. The domain name will be displayed only: • If this checkbox is selected. • If the **Resource Provider** is selected as **Active Directory**.	-

User Providers

AVEVA Work Tasks supports the following User Providers by default:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Active Directory

When you select **Active Directory** as the **Resource Provider**, details such as **Active Directory datasource** and the **Active Directory search base** are populated by default.

Authentication User, **Authentication Password**, **User Log-on ID**, and **User Log-on Display Name** are optional fields. Access to the active directory might not be available for all users. In such cases, provide the active directory authentication user name and the active directory authentication password with which you can connect to the Active Directory.

AVEVA Work Tasks considers *SAM-Account-Name* as **User Log-on ID** by default. If you want to use active directory attributes other than the *SAM-Account-Name* user, then enter the desired active directory attribute details in the **User Log-on ID** field and **User Log-on Display Name** field. **User Log-on Display Name** is the user friendly display name of the active directory attribute. This **User Log-on Display Name** is referred in the User Lookup dialog box against the **User Log-on ID**.

Steps

1. When you select **Active Directory** as the **Resource Provider**, then the **Use Token Service** field appears. In this field, select one of the following:
   - If you do not want to use AIM authentication, select **Without Token Service**.
   - If you want to use AIM authentication, then select a server from the list of alias server names.
2. Enter **Active Directory datasource**.
3. Enter **Active Directory search base**.
4. Enter **Domain Name**.

   The domain name is case-sensitive.
5. Enter **Authentication User**.
6. Enter **Authentication Password**.
7. Check **Is Cross Domain Role Provider** checkbox, if cross domain users and roles mapping is required.

   Example: Users *John* and *Jim* of domain *TelematicsU* and users *Mike* and *Bob* from domain *TelematicsAU* are added to role Team HR of domain *TelematicsMain*.
8. Enter **User Log-on ID** if you want to use another Active Directory attributes instead of *SAM-Account-Name*.

   We recommend to use attributes of type *string* as input for **User Log-on ID.** However, *ObjectSID* and *objectGUID* are also supported.

   After you set the **User Log-on ID**, you cannot change it later.
9. Enter the **User Log-on Display Name**.

   After you set the **User Log-on Display Name**, you cannot change it later.

   This is the default display name of the user account of the active directory user used. **User Log-on Display Name** is used in user lookups.
10. Click **Test LDAP**.

    A message about successful LDAP test appears.

Note:
- Some of the active directory attributes or properties are not indexed. If you use an active directory attribute that is not indexed, it might impact the log-on validation performance based on the active directory entries. We recommend to use indexed attributes as **User Log-on ID** or index the attributes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- For an upgraded repository, you cannot change both **User Log-on ID** and **User Log-on Display Name**.
- You cannot localize **User Log-on ID** and **User log-on Display Name** fields.
- Disabled users will not be shown in user lookup search.

Using Forms

For Forms Resource Provider:

- If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.

1. In the **Resource Provider** drop-down list**,** select **Forms.**
2. In the **User Entity properties** field, enter XML data.

   Click **Sample Code** to get the XML Schema Template.

   Example

   ```
   <properties>
   <property name="userid" type="string" purpose="identifier" />
   <property name="username" type="string" purpose="name" />
   <property name="pwd" type="string" purpose="password" />
   </properties>
   ```
3. In the **User Provider Properties** field, enter the same XML Schema.

   There must be at least one identifier in the XML Schema. For example, UserID
4. In the **Datasource** field, enter a valid data source or specify a dll.

   The value for the data source can be specified as **Physical path of the dll** or **Physical path of the dll::param. Param** is not mandatory.

   See topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Build Business Process* section for creating the assembly.

   Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly, whereas one class can return multiple connection strings.

   Examples of data source parameter:

   - c:\DLLStore\GetConnectionString.dll
   - c:\GetConnectionString.dll::Leave

   If there are two data sources, *Leave* and *Licensing*, and only one DLL is used to return a connection string based on the data source selected. The parameter can be specified as **c:\GetConnectionString.dll::Leave** or **c:\ GetConnectionString.dll::Licensing**. The connection string can be returned from the connection dll based on this parameter.
5. Enter a proper table name or view name which holds the User Name and Password for the Authentication purpose.

   Example: Passmast (the table which is available in IMS_NEW Database)
6. Click **Test SQL Connection**.

   A message about successful test connection appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using ASP.NET Membership

This resource provider lists users defined in the asp.net membership database.

Lists

This resource provider lists users created using AVEVA Work Tasks Users Creation. If you are unable to determine a User Provider to create a repository, use AVEVA Work Tasks Lists.

To verify successful creation of the Repository:

1. Click **Enterprise Console** URL from the site path column.
   The repository created is listed.
   Add the created Enterprise Console link into "Add to Favorites" to easily navigate in future.
2. Click **Repository User Provider** link to logon to the repository.
   The Enterprise Console log-on page appears.
3. Enter the logon details, and then click **Login**.
   The Enterprise Console home page appears. This indicates successful creation of a Repository.
   All features can only be available for a user with administrative privileges.

Note: Consider you created the Repository and then deleted it. If you create the same Repository again immediately and try to log into it through Enterprise Console, the Provider does not list the Repository. You need to access IIS and recycle AVEVA Work Tasks from the Application Pool to get the Repository listed in the Provider list..

Verify the Installation

After you install AVEVA Work Tasks, verify that the product is installed successfully. To verify successful installation, create a simple Workflow using Process Designer. The tasks includes the following:

- Creating a Workflow
- Publishing the Workflow
- Testing the Workflow

Creating a Workflow

1. Log on to the **Enterprise Console**.
2. If the Repository is configured for Active Directory provider, go to step 4.
3. Enter the **User Name** and **Password**.
   The **User Name** and **Password** will be validated with the user provider selected while creating the repository.
   The default authentication provider is the Repository User Provider List.
4. In the **Enterprise Console**, click menu icon.
   All features can only be available for a user with administrative privileges.
5. From the menu, select **Workflows**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Workflow window appears.

6. On the Action bar, click **New**, and then click **Workflow**.
   The New Workflow window appears.
7. Enter **Title** and **Description**.
8. Click **Save & Continue**.
   The successful message appears.
9. Select **Design the Workflow now**.
   This is the default option.
10. Click **Finish**.
    The Process Designer window appears.
11. From the activities pane on the left, drag the Information activity to the Process Designer window.
12. The activity *Information1* is created.
    Click to create a link between two activities.
    The Link Action window appears.
    a. Select from the **From Action** drop-down box.
    b. Select from the **To Action** drop-down box.
    c. Click **OK**.
       The link is established.
13. Right-click *Information1*.
14. Select **Activity Properties** from the context menu.
    The Activity Properties pane appears.
    a. Enter **Name** and **Description**.
    b. Click to **Assign Actor(s)**.
       The Assign Actor(s) window appears.
    c. Select the required provider from the **Providers** drop-down list.
    d. Select values from the drop-down list.
    e. Click **Add**.
    f. Click **Update**.
15. Click to enter the **Subject** for Notification Contents.
    The window to enter subject for notification appears.
16. Click **Update**.
17. Click to enter the **Body** for Notification Contents.
    The window to enter the Body content appears.
18. Click **Update**.
19. From the Activity Properties pane, click **Save**.

Publishing the Workflow

1. On the **Publish** menu, click **Publish**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Published successful message appears.

2. Close the dialog box.

   The Enterprise Console screen lists the workflow created.

Testing the Workflow

1. Select the listed workflow.
2. Right-click the selected workflow.
3. Click **Publish and Test Run**.

   The **Status Finished successfully** appears.

   If Status is **Execution Pending** in the Process Execution window, then click on the **Refresh** button to get the status.
4. In the **Enterprise Console** menu, select **Inbox**.

   The Inbox lists the work item.
5. Click to view the work item listed.

   This indicates that the installation is successful.

Install and Configure AVEVA Work Tasks in Silent Mode

You can install and configure AVEVA Work Tasks in silent mode. This is useful when the product is installed and configured as part of another product installation.

For example, while installing an OEM product, AVEVA Work Tasks can be installed as a part of the OEM product installation. AVEVA Work Tasks provides extension, integration, and embedding capabilities in different form. To leverage these capabilities with another product, AVEVA Work Tasks product can be installed as part of that product.

Prerequisites for Installing and Configuring in Silent Mode

- Before installing AVEVA Work Tasks in silent mode, ensure that all the prerequisites are installed. For more information about installation prerequisites, see Installation Prerequisites.
- Before configuring AVEVA Work Tasks in silent mode, ensure that the **WWSKBPMConfig.xml** and the **WWSKBPMConfig.xsd** files are available in the "Bin" folder.

Install in Silent Mode

You can install AVEVA Work Tasks in silent mode using AVEVA Work Tasks Installer with switches for silent mode.

The installation of AVEVA Work Tasks in silent mode considers user inputs from command-line parameters.

The full syntax for running the installer from the command line is as follows:

<Installer EXE Name>.exe /S /v/qn

For example: AVEVA Work Tasks 2023 SP1.exe /S /v/qn

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

where,

- /S denotes silent install
- /v denotes pass arguments to Msiexec (Basic MSI and InstallScript MSI projects). The /v option is used to pass command line options and values of public properties through to Msiexec.exe.
- /qn denotes an msiexec switch indicating to run with no user interface (UI).

This command string takes the default values for parameters, such as user name, installation location, and company name.

Silent installation command with custom values for parameters

If you want to provide custom values for these parameters, then enter the syntax for silent installation command as follows:

<Installer EXE Name>.exe /S /v"/qn INSTALLDIR=\"<Custom Path>\""

For example: AVEVA Work Tasks 2023 SP1.exe /S /v"/qn INSTALLDIR=\"C:\InstallationTest\""

Note:
- After you run the silent installer, wait for a minimum of 30 minutes until the installation is complete. Check the AVEVA Work Tasks installation folder to ensure that all the required files and folders are copied to the installation folder.
- Silent Installation of Work Tasks may exit after installing the .Net Framework 4.8 prerequisite. Machine restart is required for the .Net Framework to be considered as installed and only then can our installer proceed to the next pre- requisite.

You can also call the Installer from a batch file. For more information about sample batch file script for installing and configuring AVEVA Work Tasks in silent mode, see Sample Batch File Script for Installing and Configuring Workflow Management in Silent Mode.

Configure in Silent Mode

You can configure AVEVA Work Tasks in silent mode using the **AVEVAWorkTasksConfigurator.exe** console application, which is provided as part of the installation. The console application takes the required input from command line parameters or from the parameters configured in the Configuration file **WWSKBPMConfig.xml**.

Note: Only HTTP communication can be configured when AVEVA Work Tasks is installed using Silent Installation.

Configure in Silent Mode Using XML

You can configure AVEVA Work Tasks by defining values for parameters in the **WWSKBPMConfig.xml** configuration file.

1. Browse to the "Bin" folder, which is available in the path where the product is installed.
2. Open the **WWSKBPMConfig.xml** file.
3. Pass the values for the parameters available in the **WWSKBPMConfig.xml** file as per the requirements.

   You can individually configure the following settings in silent mode by specifying appropriate values for the **ConfigurationType** parameter, as listed in the table below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration Tasks	Value for the ConfigurationType parameter
Configuring License Server	AVEVALicensing
Configuring Farm	FarmDatabase
Configuring Data Source	DataSource
Configuring Repository	Repository
Configuring Client	Client

If you want to configure all the above settings together, then pass the value for the **ConfigurationType** parameter as **All**.

All the parameters in the configuration file are listed in the following table.

4. Save the XML file.
5. Open the Command Prompt in the Administrative mode.
6. Enter the path for the "Bin" folder, which is available in the path where the product is installed, and then press **Enter**.

   For example: CD <AVEVA Work Tasks Install Path>\Bin
7. On the Command Prompt, type **AVEVAWorkTasksConfigurator.exe**.

   The configuration progress is displayed in the **WTLogger Console**.

Settings Category	Node	Description
ConfigurationType	ConfigurationType **Note:** This is a mandatory parameter.	Enter the type of configuration you want to perform in silent mode. The values available are: • All • AVEVALicensing • FarmDatabase • DataSource • Repository • Client
FarmConfiguration > AVEVALicensing > LicenseServer	LicenseServerName **Note:** This is a mandatory parameter.	Enter the name of the License Server.
	LicenseServerPort **Note:** This is a mandatory	Enter the port number used by the License Server. Ensure that the port number is valid.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	parameter.	
FarmConfiguration > FarmDatabase	EnableLoadBalancing	Enter either True or False. If the value is set to True, load balancing is enabled on the server. If the the value is set to False, load balancing is disabled on the server.
Farm Configuration > FarmDatabase > ConnectionString	DatabaseType	Enter the database type. The default value is SQL Server.
	FarmDatabaseName **Note:** This is a mandatory parameter.	Enter the name of the Farm database.
	ServerName **Note:** This is a mandatory parameter.	Provide information about the database server on which the Farm database can be installed. If SQLExpress Instance need to be used as a Database instance of Farm database, then **Server Name** should be <server name>\SQLExpress. To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455
	AuthenticationType **Note:** This is a mandatory parameter.	Provide information on the process used to validate a user. AVEVA Work Tasks solution supports both Windows and SQL Server authentication process. Example: Windows

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	UserID	Enter the SQL User ID for SQL if the authentication type is SQLServer. Enter the Windows User ID along with the domain or host name, if the authentication type is Windows.
	UserPassword	Enter the password for the specified user ID.
	IsPasswordEncrypted	Enter either True or False. If the value is set to True, then enter the password in encrypted format. **Note**: To encrypt a password string, - Use SE.BPM.Converters.dll reference in the project. - Use Encrypt(3) to encrypt password string. Sample code: String password = “$kelta001”; String encryptedPassword = password.Encrypt(3); If the value is set to False, then enter the password in plain text format.
	DatabaseAdvanceSettings	Enter any additional connection information, such as connection Timeout. Example: Connection Timeout = 20. While creating a data source for repository, use the following custom settings to set properties

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description

		for connection pool. These settings are case sensitive and should be mentioned only in lower case.
		
		`encrypt=true;`
		`enableltm=true;`
		`connectionretrywaittime=30;`
		`maxconnectionretries=10;`
		`sqlcommandtimeout=300;`
		
		• **encrypt**: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption..
		
		• **enableltm**: AVEVA Work Tasks internally implements light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive. This property can be used to turn off this feature. The default value is set as True.
		
		• **connectionretrywaittime**: While opening a new

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		connection, if the data base server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds. • **maxconnectionretries**: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite. • **sqlcommandtimeout**: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.
GlobalSettings > Communication > EmailSettings	Host	Enter the name or IP address of the mail host.
	SMTPPort	Enter the port number for sending e-mails. The default port is 25.
	SSL	Define whether the e-mail has to be secured by SSL before it is sent. The default value is False.
	EmailTimeout	Enter timeout for e-mails in seconds. The default timeout is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		10000.
	SMTPUserName	Enter the user name for logging in to the mail server.
	SMTPPassword	Enter the password.
	IsSMTPPasswordEncrypted	Enter either True or False. If the value is set to True, then enter the password in encrypted format. If the value is set to False, then enter the password in plain text.
GlobalSettings > Communication > EmailSettings > ExceptionEmailSettings	FromEmail	Enter the e-mail Id from emails detailing exceptions in workflow execution need to be sent.
	ToEmail	Enter the e-mail Id to which exceptions must be sent.
Global Settings > Communication > EmailSettings > ExceptionEmailSettings > IMSettings	UserName	Enter user name in which IMs must be sent.
	Password	Enter the password.
Global Settings > Communication > Service Monitor Setting	Service Monitor Port	Enter the port number. This is the default port used by the AVEVA Work Tasks Service Monitor EXE.
Global Settings > HTTP > URLEncryptionSettings	UseHTTPContext	Enter True to use HTTP context. Enter False if you do not want to include http context info while building the secure query string parameter used internally in AVEVA Work Tasks controls to pass information from one page to another.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	HardcodedCorrectionFactor	This field is used to set a flag primarily used for load testing. The date correction factor used by AVEVA Work Tasks is retrieved through a cookie from client browser. This value cannot be derived when testing using ACT or any other load test tool. If this factor is set to True, AVEVA Work Tasks always uses the date correction factor for converting UTC date time stored in AVEVA Work Tasks tables to the local time. The default value is False.
Global Settings > HTTP > HTTPProxy	HTTPProxyURL	Enter the URL for HTTP Proxy.
	HTTPProxyUser	Enter a user name for HTTP Proxy.
	HTTPProxyPassword	Enter a password.
	HTTPProxyDomain	Enter the domain name.
	BypassLocalRequest	Enter True to bypass local requests.
	BypassList	Enter list of URLs in the bypass list; requests from a URL in this list will be ignored.
	IsHTTPProxyPasswordEncrypted	Enter either True or False. If the value is set to True, then enter the password in encrypted format. If the value is set to False, then enter the password in plain text.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
Global Settings > LogTransaction	DebugLevel	Select from three debug levels. • Level 1 shows minimal information in the AVEVA Work Tasks Logger Console, giving only information on the start and end of the workflow. Error messages are also shown in this mode. This is the default mode. • Level 2 shows detailed information of what is happening in the background. It will show all the messages logged from the actions. • Level 3 shows only messages indicating whether the action is executed. All error messages logged in the logger console will be sent to receiver email specified in Exception Emails.

	PersistExceptionInDatabase	Enter either True or False.
		• If the value is set to True, you will be allowed to persist the exceptions which are logged in AVEVA Work Tasks logger console to the repository database. This in turn allows you to view exceptions at the following levels:
		  • Workflow level
		  • Workflow Instance level
		  • Activity level
		• The default value is False. If the value is set to False, you will not be allowed to persist the exceptions which are

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		logged in the logger console to the repository database.

	PersistWarningInDatabase	Enter either True or False.
		• If the value is set to True, you will be allowed to persist the warnings which are logged in AVEVA Work Tasks logger console to the repository database. Which in turn allows you to view warnings at the following levels:
		  • Workflow level
		  • Workflow Instance level
		  • Activity level
		• The default value is False. If the value is set to False, you will not be allowed to persist the warnings which are logged in the logger console to the repository database.

GlobalSettings > FileWatcher	IterationCount	Set the iteration value for a file copy process. The default value is 200.
	FirstAttempt	The value in this field indicates the wait time after a file copy command is executed for the first time. This wait time is provided for an event to be generated after the file copy process is complete. The value specified is in milliseconds. The default value is 2000.
	NextAttempt	The value in this field indicates the wait time for subsequent attempts. The value specified is in

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		milliseconds. The default value is 4000. **Example** The value 4000 indicates that the wait time for subsequent attempts is four seconds.
GlobalSettings > MachineSettings	MachineAddress	Enter the address of the machine on which AVEVA Work Tasks is installed.
WorkflowEngineSettings > EngineSettings	WEServerPort	Enter the port number used by the Workflow Engine Service. The Workflow Engine communicates with other services using this port. The default value is 8855.
	WELoadBalanceServerPort	Enter the port number for the Load Balanced Server. This field is disabled if Load Balanced value is set to False.
	WELoadBalanceClientPort	Enter the port number for the Load Balanced Client. This field is disabled if Load Balanced value is set to False.
WorkflowEngineSettings > AutoResumeSetting	RepositoryNames	Enter the name of the repository.
	EnableLoggingAutoResumeMessage	It enables you to control the log messages in the logger console whenever the engine tries to resume the respective workflow instances.
WorkflowEngineSettings >	NoOfRetriesToResumeFEWorkflo	Enter the number of attempts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
AutoResumeSetting > FEWorkflows	wInstances	that the Workflow Engine should make to resume the FE workflow instances. For example, if this value is set to 5 then the workflow engine would make 5 attempts to resume each of the FE workflow instance. If this value is set to 0 (zero) then, then auto resume of FE workflow instances does not occur.
	RemoveResumeHistoryOnFEWorkflowInstanceCompletion	If the value is set to True, the history of resumed FE workflow instances which have completed successfully would be deleted from the SWFECOUNT table. If the value is set to False, the history of resumed FE workflow instances which have completed successfully would be retained in the SWFECOUNT table.
	NoOfFEWorkflowInstanceToFetchSpecifiedTimeInterval	Set the number of workflow instances to be fetched for auto resume in the specified time interval.
	FETimeSpan	Set the time interval for resuming FE workflow instances. The value entered should be greater than 5 minutes.
WorkflowEngineSettings > AutoResumeSetting > EPWorkflows	EnableAutoResumeEPWorkflowInstance	Enable or disable auto resume of EP workflow instances.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	FetchWorkflowinstanceWhichHaveBeeninEPStateFor	Set the minimum time for which the workflow instances should have been in EP state. The value should be entered in days.hr:min:sec format. If this value is set as 5, the workflow engine tries to fetch the workflow instances which have been in EP state for at least 5 hours. The workflow engine calculates the time for which the workflow instances have been in EP state by comparing the LastUpdatedDateTime value from SWExecute table with the current time. The default time for which workflow instances have been in EP state is 5 hours. If the time set is less than 5 hours, then by default, the engine considers 5 hours as the minimum time.
	TimeIntervaltoFetchEPWorkflowInstances	Set the time interval for resuming EP workflow instances. Minimum value should be greater than 5 minutes.
	NoofEPWorkflowInstancesToBeFetchedInSpecifiedTimeInterval	Set the number of EP workflow instances to be fetched for auto resume in the specified time interval.
WorkflowEngineSettings > AutoResumeSetting > PendingWorkflows	EnableAutoResumeOfPendingWorkflowEvents	Enable or disable auto resume of pending workflow events.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	MinimumTimeForWhichTheWorkflowEventsAreinPendingState	Set the minimum time for which the workflow events should be in Pending state. If this value is set as 0.5:0:0, the workflow engine will fetch the events which are in Pending state for at least 5 hours. The workflow engine calculates the time for which the workflow events have been pending by comparing the LastUpdatedDateTime value from SWExecute table with the current time. The default time for which workflow events should be in pending state is 5 hours. If the time set is less than 5 hours, then by default, the engine considers 5 hours as the minimum time.
	TimeIntervalToFetchWorkflowInstancesWithPendingEvents	Set the time interval for resuming workflow instances with pending events. Minimum value should be greater than 5 minutes.
	NoOfWorkflowInstancesWithPendingEventsToBeFetchedIn SpecifiedTimeInterval	Set the number of workflow instances with pending events to be fetched for auto resume in the specified time.
AdvancedServerSettings	MultiCastPort	Use this port to check the connectivity between all the machines while running the Load Balanced Server. This is not used if the **Enterprise Server Mode** is **Database**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	HeartbeatInterval	Specify the heartbeat interval for the Enterprise Server. The default value is 3000 ms.
	HeartBeatTimeout	Specify the heartbeat timeout value for the Enterprise Server. The default value is 10 seconds.
	AverageCPUCount	Specify the average CPU count for the Enterprise Server. The default value is 10.
	MultiCastIP	Displays the multicast IP address. The default values are: • IPv4: 225.0.0.10 • IPv6: FF02::1
	TCPPort	Communication between the Workflow Engine and the Advance Server service is established through this port. This port is used to get the status of the least used server, and also analyze server performance. The default value is 8853.
	ASMode	Select the mode as Multicast or Database. • If you select the mode as Multicast, then it uses the configured Cluster Port, Cluster IP, and Multicast Local IP to share its server status with other servers. • If you select the mode as Database, the server status is stored in the database and is shared with other servers.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		The default value is Multicast.
	LocalIP	Displays the local server name or IP address. IP address supports both IPv4 and IPv6 address types.
	MultiCastLocalIP	Displays the multicast server name or IP address. Local IP address supports both IPv4 and IPv6 address types. This field is not used if you select the **Enterprise Server Mode** as **Database**.
TaskSchedulerSettings	TSServerPort	Enter the port number used by Task Scheduler service. This value must be modified if the default port value is in use. The default value is 8856.
	TSLoadBalancedServerPort	Enter the port number for the Load Balanced Server. This field is disabled if Load Balanced is set to False. The default value is 8853.
	TSLoadBalancedClientPort	Enter the port number for the Load Balanced Client. The default value is 8860.
	TaskFetchInterval	Enter the interval in seconds at which tasks have to be retrieved. The default value is 60.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	TSFileWatcher	Define a folder for File Watcher operation.
	TSThreadPoolSize	Define a thread pool size for the Workflow Engine. A thread pool size determines the number of threads reserved by a server for providing service requests. For the Workflow Engine deployed in a heavy load scenario, a higher thread pool size needs to be set. The default value is 25.
CommunicationServiceSettings > CommunicationServer	ListenerIPAddress	Enter the IP address to be monitored for incoming emails. This field defaults to the name of the logged in machine.
	ListenerPort	Enter the port number for the Listener IP Address. The default value is 8770.
	WaitTaskHandlerPath	Enter the assembly navigation path for handling timeouts and e-mail received.
	TaskHandlerClass	Enter the Class name for handling timeout tasks and incoming emails.
	BlockMailAddresses	Enter the e-mail addresses that should be blocked from processing.
CommunicationServiceSettings > CommunicationServer > AdvancedServerSettings	LoadBalancedServerPort	Enter the port number used by the Enterprise Server. This value must be modified if the default port value is in use. The default value is 8853.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	LoadBalancedClientPort	Enter the port number for the Client. The default value is 8864.
CommunicationServiceSettings > MailServer > Default	CommunicationServiceServerIP	Enter the IP address of the server where the Communication Service is installed.
	CommunicationServiceServerPort	Enter the Port number used by the Communication Service. The default value is 25.
CommunicationServiceSettings > MailServer > POP3	POPMailHost	Enter the IP address of the POP Mail Host.
	POPTimeInterval	Set the time interval to check the mail server for emails. The format for this field is day.hour.minute.seconds. The default value is 0.00:00:15.
	POPMailPort	Enter the port number for receiving e-mails. The default value is 110.
	IsSPAEnabled	Specify if a Secure Password Authentication is provided. The default value is False.
	IsPOPProvideEncryptedPwd	Enter either True or False. If the value is True, then enter the password in encrypted format. If the value is set to False, then enter the password in plain text.
	IsPOP3PasswordEncrypted	Enter either True or False.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		If the value is True, then enter the password in encrypted format. If the value is set to False, then enter the password in plain text.
	POPUsers	Enter the User Name and password of POP3 user.
CommunicationServiceSettings > SMSReceiver	EnableSMSReceiver	This field enables the Communication Engine to check for SMS. The default value is False.
	SMSTimeInterval	Enter the time interval at which the AVEVA Work Tasks Engine must check for SMS in the phone device. Example: 0.00:01:30 = 1 minute 30 secs The default value is 0.00:01:00.
	SMSTaskPath	Enter the Assembly path for handling timeout actions and warnings.
	SMSTaskClass	Enter the class name for handling timeout actions and warnings.
	NotificationSMSProvider	Enter the Assembly path for handling timeout actions and warnings.
CommunicationServiceSettings > SMSReceiver > ProviderConfigurations	ProviderConfiguration	These parameters are used for configuring the SMS infrastructure (sending and receiving SMS).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		• SMSProviderDeviceSettings: 8 databits, no parity, 1 stopbit • SMSProviderDialString • SMSProviderProviderType: Standard TAP provider You can specify 0 for UCP Provider, 2 for TAP provider, no request for LineFeed, and 3 for TAP provider, no request for EOT. • SMSEmailProviderAssembly • SMSEmailProviderClass • SMSEmailProviderEmailServer Enter the details specific to your service provider. AVEVA Work Tasks currently supports only GSM protocols for receiving SMS. • **Name**: Enter the name of the service provider. • **Assembly**: Enter the path. • **Class**: Enter the class name for receiving SMS. • **Device**: Enter device details.
CommunicationServiceSettings > SMSConfig	ConnectionSequences	Enter the sequence in which the Engine must choose the connection type for sending SMS. • **Connection Type**: This column is used to list the various connection types such as GSM, SMSPatternMatch, EmailProvider, SMSServiceProvider, and Hayes. • **Sequence Number**: This column is used to list the sequence of each connection type.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
CommunicationServiceSettings > SMSConfig > ResourceField	SMSResourceFieldsEmail	This field is mapped to the connection sequence table for sending SMS. The fields are mapped in the user provider for sending SMS. The three fields - Email, Mobile Phone Provider, and SMS Number are required to be mapped to the table. Some providers allow SMS to be sent through email. Enter your email address in the email field to send SMS to these providers.
	SMSResourceFieldMobilePhoneProvider	This field is mapped to the table for sending SMS. Enter the name of the Mobile Phone Provider in this field. Example: Hutch
	SMSResourceFieldSMSNumber	This field is mapped to the table for sending SMS. Enter the phone number to which the SMS has to be sent.
CommunicationServiceSettings > SMSConfigCustom > SMSPatternMatch	SMSPatternMatchAssembly	Define an assembly which handles sending and receiving SMS.
	SMSMessengerAssemblyClass	Enter the class name which handles sending and receiving SMS.
CommunicationServiceSettings > SMSConfigCustom > SMSPattern	SMSProvider	• **SMS Provider**: Provider name such as, Hutch or Airtel. • **Connection Type**: Connection type like EmailProvider, GSM, SMSServiceProvider or Hayes. • **SMS Pattern**: Pattern like 9845*, 98*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
CommunicationServiceSettings > SMSSendAction > CustomSMSProviders	CustomSMSProvider	• **Name**: Enter the name of the SMS Provider. • **SMSProviderSMSMessengerAssembly**: Enter the path to the assembly. • **SMSProviderSMSMessengerClass**: Enter the class information. • **SMSProviderDevice**: Enter the device for the custom SMS provider. • **SMSProviderDeviceSpeed**
NELoadBalance	NELoadBalanceServerPort	Enter the port number for the Load Balanced Server. This field is disabled if Load Balanced is set to False.
	NELoadBalanceClientPort	Enter the port number for the Load Balanced Client.
MobileNotificationServiceSettings	MNLoadBalanceServerPort	Enter the port number for the Load Balanced Server. This field is disabled if Load Balanced is set to False.
	MNLoadBalanceClientPort	Enter the port number for the Load Balanced Client.
	SchedulePeriod	If the notification is not delivered at the first attempt, you can set the interval between delivery attempts. The system tries to resend the notification at the interval defined in this field. The time entered should be in days:hr:min:sec format only. The

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		maximum limit is 10 minutes. The default value is 30 seconds.
	MessageRetryCount	If the notification is not delivered at the first attempt, you can set the number of retry attempts to deliver the notification. After the defined limit is reached, the notification expires. If the Retry Count value is set as 0, the failed notifications will not be resent. The default value is 5.
MobileNotificationServiceSettings > Register > RegistrationDetails	RegistrationRetryCount	A device needs to be registered in the Azure notification hub before it can receive notifications. If the device registration is not successful at the first attempt, you can set the number of retry attempts to register the device. If the Retry Count value is set as 0, the failed device registrations do not happen. The default value is 5.
	ScheduleDeviceRegistrationPeriod	If the device registration is not done at the first attempt, you can set the interval between device registration attempts. The system tries to re-register the device at the interval defined in this field. The time entered should be in days:hr:min:sec format only. The default value is 2 minutes.
	ClientId	Not applicable. The parameter is retained for backward compatibility.
	MachineId	Not applicable. The parameter is retained for backward compatibility.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
DataSource	DataSourceName **Note:** This is a mandatory parameter.	Enter the name of the datasource to be created.
	Description	Enter the description for the datasource.
	ServerName **Note:** This is a mandatory parameter.	Enter the database server or instance name. To use a database on the local machine, enter "localhost", ".", or the local computer's name for the Server Name. For SQL Server database type, specify IPv4 and IPv6 addresses for the Server Name. To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455 To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be <server name>\<instance name>.
	AuthenticationType **Note:** This is a mandatory parameter.	Use this field to determine the process to validate a user who is trying to connect to the database. AVEVA Work Tasks uses Windows or SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. For Example: Windows or SQLServer
	UserId	Enter the ID of the user who is trying to connect to the database.
	Password	Enter the password of the user.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	PasswordEncrypted	Enter either "True" or "False". If the value is set to "True", then the user has to enter the password in encrypted format. **Note**: To encrypt a password string, - Use SE.BPM.Converters.dll reference in the project. - Use Encrypt(3) to encrypt password string. Sample code: String password = “$kelta001”; String encryptedPassword = password.Encrypt(3); If the value is set to "False", then the user has to enter the password in plain text.
	AdvanceSettings	Provide the advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection. For Example: Connection Timeout = 20
	DatabaseName **Note:** This is a mandatory parameter.	Enter the name of the database that can be used as a datasource.
RepositorySettings	ResourceProvider	Enter one of the following values: • ActiveDirectory • Forms • ASPNETMembership • RepositoryUserProvider • ConnectUserProvider
	RepositoryDataSource **Note:** This is a mandatory parameter.	Select the data source that will be used to store data for the current repository.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	RepositoryName **Note:** This is a mandatory parameter.	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the repository name. For Example: Telematics Repository
	Description **Note:** This is a mandatory parameter.	Provide a description for the repository. For example: This stores Telematics Employee Leave Details.
	FromMail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might occur in the In this scenario, you have a Web Server and Services installed on the same machine. You can host this set up on multiple computers to form a Web Farm with clustered the Services. The Databases are installed on a machine by itself. While scaling out, ensure that the machine has both the Web Server and the Services installed. See: Web Server Farm with Clustered Work Tasks Services and Databases on Different Machines In this scenario, you can maintain

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		a Web Farm with the Web Servers and Services cluster on separate machines while the Databases are installed on a different machine. This deployment provides scalability and high availability of the Workflow Engine and the Web Servers. This is achieved by distributing the load equally among all the machines in the cluster and the Web Farm. Multiple servers in the Web Farm or the cluster run the same services. Failure of a server does not impact other servers. You can take the affected server offline while the other servers maintain the service. This type of installation provides Failover. Thus allowing you to easily add or remove machines from the Web Farm or from the cluster without affecting the execution of workflows. This is the optimal scenario for heavy load conditions and can be considered if you anticipate heavy load on the Workflow Engine or the Web Server. See: Web Farm Connecting to Separate Services Cluster with Database on a Different Machine Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be sent to the mail ID defined in **Exception to mail**. The from address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		avoid this, restart the Workflow Engine Service and the Communication Service. For example: admin@domain.com
	ExceptionToMail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails are sent to the email ID specified in Exception to mail. For example: admin@domain.com
	AlertInitiatorOfWorkflow	Determines if the status of a workflow should be sent as a notification mail to the initiator of the workflow. For example: A process with a rule to check the pending leave available for the employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
	EnableTheme	Set the value to enable the defined theme.
	ThemeName	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. AVEVA Work Tasks provides Default and Lite themes. Do not use the Default folder for any style sheet customizations.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	CustomApplicationFolder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with a semicolon.
	HideRepository	Set the value to make the repository unavailable to the Farm.
	SourceElements	Enter the path to the Source Elements Folder.
	AttachmentDefaultPhysicalStoragePath	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an absolute path. - The default path is the **Attachments** folder in the installation path. - The default path is used when no path is specified.
	AutoLogOffTime	Enter the time after which the logged in user must be automatically logged off after the designated period of inactivity. Enter the time in minutes. Ensure that you enter only valid integer values between 1 to 360. 0 minutes is an invalid value. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. If you do not specify a value, then

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		the log off time is not set for the repository, and the session does not log off automatically. You use also define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML. For example: 1 to 360 (in minutes)
	UserCredentialsTimeoutForApp	Enter the time in hours to request for user credentials for AVEVA Work Tasks Pro after the set time is elapsed. For example: 1 to 48 (in hours) The default value is 0, which disables this setting.
	IsWorkTasksProOfflineSupportEnabled	Enter either True or False for enabling/disabling offline support in AVEVA Work Tasks Pro app. If the value set is True, the offline support is enabled. If the value set is False, the offline support is disabled. The default value is True.
	IsWorkTasksProRefreshOnWorkItemSubmissionEnabled	Enter either True or False to allow/disallow refreshing the Inbox in AVEVA Work Tasks Pro app. If the value set is True, the Inbox is automatically refreshed on submitting the work item. If the value set is False, the Inbox is not refreshed on submitting the work item.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		The default value is True.
	TimeoutForWorkTasksPro	Set the time out value in seconds for AVEVA Work Tasks Pro app. The default value is 60.
	FillFormLabelForWorkTasksPro	Specify the label for the Fill Form for AVEVA Work Tasks Pro.
	DateConfigAlternateActors	Select the Enable Date Settings option to set the From date and To date as mandatory while creating alternate actors. For more information about creating alternate actors, see Alternate Actors topic in the *User Guide*.
	IsDomainNameToShowWithUser	Enter the value to determine whether the domain name must be displayed along with the user name in the Enterprise Console header. The domain name will be displayed only if the **Resource Provider** is selected as **Active Directory**.
	ActiveDirectoryDatasource **Note:** This is a mandatory parameter.	Enter the name of the Active Directory datasource.
	ActiveDirectorySearchBase **Note:** This is a mandatory parameter.	Enter the Active Directory Search Base to search active directories for users.
	DomainName **Note:** This is a mandatory parameter.	Enter the domain name.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	AuthenticationUser	Enter the authentication user.
	AuthenticationPassword	Enter the password.
	IsCrossDomainRoleProvider	Specify whether cross domain users and roles mapping is required.
	UserLogonID	Enter the value if you want to use another Active Directory attributes instead of SAM-Account-Name. It is recommended to use attributes of type string as input for User Log-on ID. After you set the User Log-on ID, you cannot change it later.
	UserLogonDisplayName	Enter the User Log-on Display Name. This is the default display name of the user account of the active directory user used. User Log-on Display Name is used in user lookups. After you set the User Log-on Display Name, you cannot change it later.
	AspnetMembershipDataSource	Enter the data source for the user table.
	ApplicationName	Enter the application name for the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		ASP.NET Membership provider.
	UserEntityProperties	Enter the XML code that identifies user properties.
	UserProviderProperties	Enter the XML Schema. There must be at least one identifier in the XML Schema. For example: UserID
	FormsProviderDatasource	If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.
	DataTable	Enter a valid data source or specify a dll.
	UserLoginProperty	Enter the name of the User Table or View in the Datasource. This property value will be used as User ID when logging in to the Enterprise console.
	ProvisionMessageQueue	Enter the value as True to provision the repository for Message Queue.

You need not configure the ConnectConfiguration and Https settings in the XML:

```
<ConnectConfiguration enabled="false">
<ConnectEnvironment></ConnectEnvironment>
<ApplicationClientId></ApplicationClientId>
<ApplicationClientSecret></ApplicationClientSecret>
<ClientCredentialsClientId></ClientCredentialsClientId>
<ClientCredentialsClientSecret></ClientCredentialsClientSecret>
<WorkTasksClientId></WorkTasksClientId>
<WorkTasksClientSecret></WorkTasksClientSecret>
<WorkTasksProClientId></WorkTasksProClientId>
<AccountId></AccountId>
<SolutionDefinitionId></SolutionDefinitionId>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<SolutionId></SolutionId>
<AssetName></AssetName>
<Quickflow enabled="false">
<AzureServiceBus>
<SubscriptionId></SubscriptionId>
<TenantId></TenantId>
<ClientId></ClientId>
<ClientSecret></ClientSecret>
<ResourceGroupName></ResourceGroupName>
<ServiceBusNamespaceName></ServiceBusNamespaceName>
<ServiceBusUserName></ServiceBusUserName>
<Location></Location>
</AzureServiceBus>
</Quickflow>
</ConnectConfiguration>

<SiteSuffix></SiteSuffix>

<Https enabled="false">
<Certificatepath></Certificatepath>
<Certificatepassword></Certificatepassword>
<Port>8001</Port>
</Https>
```

Configure DataSource in Silent Mode Using XML

A Repository DataSource is used to store all repository related data. The Repository DataSource must be created on a machine where the Central Configuration is installed.

You can configure a Data Source using the **WWSKBPMConfig.xml** configuration file.

For more information about the parameters in the **WWSKBPMConfig.xml** configuration file, see Configure in Silent Mode Using XML.

1. Open the **WTLogger Console**.
2. Browse to the "Bin" folder, which is available in the path where the product is installed.
3. Open the **WWSKBPMConfig.xml** file.
4. In the XML file, pass the values for the following mandatory parameters.
   - **ConfigurationType**: Enter the value as **Datasource**.
   - **DataSourceName**: Enter the name of the datasource to be created.
   - **Description**: Enter the description for the datasource.
   - **ServerName**: Enter the database server or instance name.

     To use a database on the local machine, enter "localhost", ".", or the local computer's name for the Server Name.

     For SQL Server database type, specify IPv4 and IPv6 addresses for the Server Name.

     To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

     To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be <server name>\<instance name>.
   - **AuthenticationType**: Enter the type of process to validate a user who is trying to connect to the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

database.

AVEVA Work Tasks uses Windows or SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database.

For Example: Windows or SQLServer

- **UserId**: Enter the ID of the user who is trying to connect to the database.
- **Password**: Enter the password of the user.
- **PasswordEncrypted**: Enter either "true" or "false".

  If the value is set to "true", then the user has to enter the password in encrypted format.

  If the value is set to "false", then the user has to enter the password in plain text.
- **AdvanceSettings**: Enter the advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection.

  Example: Connection Timeout = 20;
- **DatabaseName**: Enter the name of the database that can be used as a datasource.
- **EnableTablePartition**: Select to enable database table partitioning by entering either "true" or "false".
- **DatabaseSecondaryDataFilePath**: Enter the database secondary data file path, which should be on a Microsoft SQL Server machine with sufficient rights to create files on the specified path. Ensure, you provide a valid path as it is not validated by the business application. If the path is not specified, then the secondary database file will be created at the default path, where the primary database file exists.

Note: If **EnableTablePartition** is set to true for a SQL Server version that does not support table partition, then the data source is created without a table partition.

5. Save the XML file.
6. Run the **AVEVAWorkTasksConfigurator.exe**.

   The configuration progress is displayed in the **WTLogger Console**.

Configure License Server Details in Silent Mode Using XML

After installing or upgrading AVEVA Work Tasks in silent mode, you have to configure the license server.

For more information about parameters in the **WWSKBPMConfig.xml** configuration file, see Configure in Silent Mode Using XML.

1. Open the **WTLogger Console**.
2. Browse to the "Bin" folder, which is available in the path where the product is installed.
3. Open the **WWSKBPMConfig.xml** file.
4. In the XML file, pass the values for the following parameters.
   - **ConfigurationType**: Enter the value as **AVEVALicensing**.
   - **LicenseServerName**: Enter the name of the License Server.
   - **LicenseServerPort**: Enter the port number used by the License Server.
5. Save the XML file.
6. Run the **AVEVAWorkTasksConfigurator.exe**.

   The configuration progress is displayed in the **WTLogger Console**.

Note: If there are any errors while updating the license server details in silent mode, these exceptions are

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

not logged in the **WTLogger Console**.

Configure Farm in Silent Mode Using XML

You can configure the Farm settings using the **WWSKBPMConfig.xml** configuration file.

For more information about the parameters in the **WWSKBPMConfig.xml** configuration file, see Configure in Silent Mode Using XML.

1. Open the **WTLogger Console**.
2. Browse to the "Bin" folder, which is available in the path where the product is installed.
3. Open the **WWSKBPMConfig.xml** file.
4. In the XML file, pass the values for the following mandatory parameters.
   - **ConfigurationType**: Enter the value as **FarmDatabase**.
   - **FarmDatabaseName**: Enter the name of the Farm database.
   - **DatabaseType**: Enter the type of database. For example: SQLServer.

     Note: Ensure that you enter the database type value without any space.
   - **DatabaseAuthenticationType**: Enter the type of authentication. For example: Windows or SQLServer

     Note: Ensure that you enter the authentication type value without any space.
   - **DatabaseUserName**: Enter the name of the database user along with the domain name.
   - **DatabasePassword**: Enter the password for the database user.
   - **AdvancedSettings**: Enter the advanced settings for database. For example: Connection Timeout=20;
5. Save the XML file.
6. Run the **AVEVAWorkTasksConfigurator.exe**.

   The configuration progress is displayed in the **WTLogger Console**.

Configure Repository in Silent Mode Using XML

A Repository contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

You can configure a repository using the **WWSKBPMConfig.xml** configuration file.

For more information about the parameters in the **WWSKBPMConfig.xml** configuration file, see Configure in Silent Mode Using XML.

1. Open the **WTLogger Console**.
2. Browse to the "Bin" folder, which is available in the path where the product is installed.
3. Open the **WWSKBPMConfig.xml** file.
4. In the XML file, pass the values for the following mandatory parameters.
   - **ConfigurationType**: Enter the value as **Repository**.
   - **RepositoryDataSource**: Enter the name of the datasource on which the repository has to be created. Ensure that the specified datasource is already configured in the system.
   - **RepositoryName**: Enter the name of the repository.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Description**: Enter the description for the repository.

5. Save the XML file.
6. Run the **AVEVAWorkTasksConfigurator.exe**.

   The configuration progress is displayed in the **WTLogger Console**.

Configure Quickflow in Silent Mode Using XML

You can configure Quickflow using the **WWSKBPMConfig.xml** configuration file.

1. Open the **WTLogger Console**.
2. Browse to the "Bin" folder, which is available in the path where the product is installed.
3. Open the **WWSKBPMConfig.xml** file.
4. In the XML file, pass the values for the following mandatory parameters.
   - **ConfigurationType**: Enter the value as **Azure**.
   - **RepositoryDataSource**: Enter the name of the datasource on which the repository has to be created. Ensure that the specified datasource is already configured in the system.
   - **RepositoryName**: Enter the name of the repository.
   - **Description**: Enter the description for the repository.
5. Pass the following mandatory parameters for deploying the Quickflow using Azure service bus:
   - **SubscriptionId**: Enter the subscription ID of the Azure service bus.
   - **TenantId**: Enter the tenant ID of the Azure service bus.
   - **ClientId**: Enter the client ID of the Azure service bus
   - **ResourceGroupName**: Enter the resource group name of the Azure service bus
   - **ServiceBusNamespaceName**: Enter the name of the the Azure service bus
   - **ServiceBusUserName**: Enter the user name of the Azure service bus
   - **Location**: Enter the location where the Azure service bus is installed.
6. Save the XML file.
7. Run the **AVEVAWorkTasksConfigurator.exe**.

   The configuration progress is displayed in the **WTLogger Console**.

All the parameters to be configured in the configuration file are listed in the following table.

Settings Category	Nodes	Description
FarmConfiguration > ConnectConfiguration	ConnectEnvironment	Enter the name of the connect environment to be enabled.
FarmConfiguration > ConnectConfiguration > CentralConfig	ClientId	Not applicable. The parameter is retained for backward compatibility.
	ClientSecret	Not applicable. The parameter is retained for backward

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Nodes	Description
		compatibility.
	RedirectUris	Enter the CentalConfig URI to which the user is to be redirected.
FarmConfiguration > ConnectConfiguration > EnterpriseConsole	ClientId	Not applicable. The parameter is retained for backward compatibility.
	ClientSecret	Not applicable. The parameter is retained for backward compatibility.
	RedirectUris	Enter the EnterpriseConsole URI to which the user is to be redirected.
FarmConfiguration > ConnectConfiguration > Quickflow	ClientId	Not applicable. The parameter is retained for backward compatibility.
	ClientSecret	Not applicable. The parameter is retained for backward compatibility.
FarmConfiguration > ConnectConfiguration > Quickflow > AzureServiceBus	SubscriptionId	Enter the subscription id of the Azure service bus.
	TenantId	Enter the tenant id of the Azure service bus.
	ClientId	Not applicable. The parameter is retained for backward compatibility.
	ClientSecret	Not applicable. The parameter is retained for backward compatibility.
	ResourceGroupName	Enter the resource group name of the Azure service bus.
	ServiceBusNamespaceName	Enter the service bus name of the Azure service bus.
	ServiceBusUserName	Enter the service bus user name of the Azure service bus.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Nodes	Description
	Location	Enter the location where the Azure service bus is installed.

Configure Cryptography Settings in Silent Mode using XML

With Cryptography settings, you can use encryption algorithm in AVEVA Work Tasks.

You can configure Cryptography settings using the **WWSKBPMConfig.xml** configuration file.

For more information about the parameters in the **WWSKBPMConfig.xml** configuration file, see Configure in Silent Mode Using XML.

1. Open the **WTLogger Console**.
2. Browse to the "Bin" folder, which is available in the path where the product is installed.
3. Open the **WWSKBPMConfig.xml** file.
4. In the XML file, pass the value for the **Passphrase** parameter.

   Note: The value should have characters between 8 and 20, consisting of lowercase, uppercase, number, and special characters. Use the same passphrase for all servers in Load Balanced Server of AVEVA Work Tasks.
5. Save the XML file.
6. Run the **AVEVAWorkTasksConfigurator.exe**.

   The configuration progress is displayed in the **WTLogger Console**.

   Note:

   - While upgrading the product in silent mode, pass the argument as **Passphrase=Wonderware@123** to configure Cryptography Settings and use the encryption algorithm in the product.

   - For the Configuration Type = CLIENT, the Passphrase should be the same as that of the Server Passphrase.

Configure in Silent Mode using Command-Line Parameters

You can configure AVEVA Work Tasks in Silent mode using Command-Line parameters.

- Configure Farm in Silent Mode Using Command-Line Parameters
- Configure DataSource in Silent Mode Using Command-Line Parameters
- Configure Repository in Silent Mode Using Command-Line Parameters

Configure Farm in Silent Mode Using Command-Line Parameters

You can configure the Farm settings using the **AVEVAWorkTasksConfigurator.exe**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To configure the Farm on a Server

1. Open the Command Prompt in the Administrative mode.
2. Enter the path for the "Bin" folder, which is available in the path where the product is installed, and then press **Enter**.

   For example: CD <AVEVA Work Tasks Install Path>\Bin
3. On the Command Prompt, type **AVEVAWorkTasksConfigurator.exe**.
4. Pass the values for the following command line arguments:
   - **ConfigurationType** = **FARMDATABASE**.
   - **FarmDatabaseName** = <The name of the FARM database.>
   - **DatabaseServerName** = <The name of the database server.>
   - **DatabaseType** = <The type of database. For example: SQLServer>

     Note: Ensure that you enter the database type value without any space.
   - **DatabaseAuthenticationType =** <The type of authentication. For example: Windows or SQLServer>

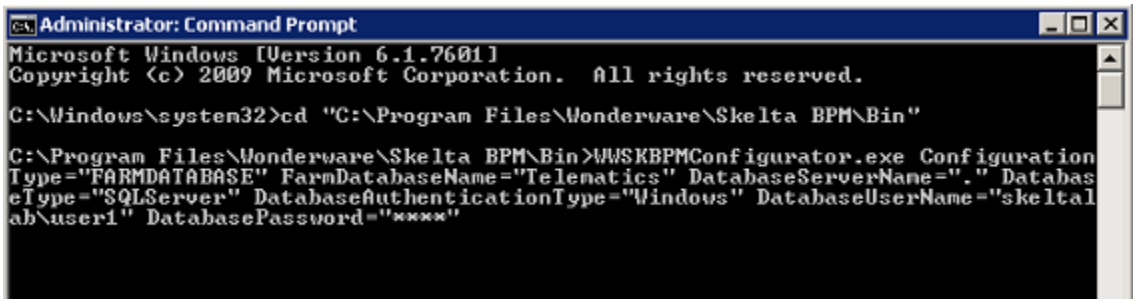
     Note: Ensure that you enter the authentication type value without any space.
   - **DatabaseUserName** = <The name of the database user along with the domain name.>
   - **DatabasePassword** = <The password for the database user.>
   - **AdvancedSettings** = <The advanced settings for database. For example: Connection Timeout=20;>

     Consider the following sample command.

     ```
     AVEVAWorkTasksConfigurator.exe ConfigurationType="FARMDATABASE"
     FarmDatabaseName="Telematics" DatabaseServerName="." DatabaseType="SQLServer"
     DatabaseAuthenticationType="Windows" DatabaseUserName="skeltalab\user1"
     DatabasePassword="****"
     ```

     Note: Before you copy the sample command to the Command Prompt, it is recommended that you verify this command in a Text Editor.

     After passing the above sample command, the Command Prompt appears as follows.



The progress of the Farm configuration is displayed in the **WTLogger Console**.

Note: The following settings are not configured as part of the Farm configuration in silent mode:
- Azure Settings
- ArchestrA Services Settings
- Enterprise Console Settings
- Central Configuration Settings

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To configure Farm on a Client

1. Open the Command Prompt in the Administrative mode.
2. Enter the path for the "Bin" folder, which is available in the path where the product is installed, and then press **Enter**.

   For example: CD <AVEVA Work Tasks Install Path>\Bin
3. On the Command Prompt, type **AVEVAWorkTasksConfigurator.exe**.
4. Pass the values for the following command line arguments:
   - **ConfigurationType: CLIENT**.
   - **FarmDatabaseName**: <The name of the FARM database.>
   - **DatabaseServerName**: <The name of the database server.>
   - **DatabaseType**: <The type of database. For example: SQLServer>

     Note: Ensure that you enter the database type value without any space.
   - **DatabaseAuthenticationType**: <The type of authentication. For example: Windows or SQLServer>

     Note: Ensure that you enter the authentication type value without any space.
   - **DatabaseUserName**: <The name of the database user along with the domain name.>
   - **DatabasePassword**: <The password for the database user.>
   - **AdvancedSettings**: <The advanced settings for database. For example: Connection Timeout=20;>

   The progress of the Client configuration is displayed in the **WTLogger Console**.

Configure DataSource in Silent Mode Using Command-Line Parameters

A Repository DataSource is used to store all repository related data. The Repository DataSource must be created on a machine where the Central Configuration is installed.

You can configure a Data Source using the **AVEVAWorkTasksConfigurator** Console Application.

1. Open the Command Prompt in the Administrative mode.
2. Enter the path for the "Bin" folder, which is available in the path where the product is installed, and then press **Enter**.

   For example: CD <AVEVA Work Tasks Install Path>\Bin
3. On the Command Prompt, type **AVEVAWorkTasksConfigurator.exe**.
4. Pass the values for the following command line arguments:
   - **ConfigurationType** = **DATASOURCE**.

     The parameter used to configure datasource.
   - **DataSourceName** = <The name of the datasource to be created.>
   - **Description** = <The description for the datasource.>
   - **ServerName** = <The database server or instance name.>

     To use a database on the local machine, enter "localhost", ".", or the local computer’s name for the **Server Name**.

     For SQL Server database type, specify IPv4 and IPv6 addresses for the Server Name.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: localhost,1455

To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be <server name>\<instance name>.

- **AuthenticationType** = <The process to validate a user who is trying to connect to the database.>

  AVEVA Work Tasks uses Windows or SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database.

  For Example: Windows or SQLServer
- **UserId** = <The ID of the user who is trying to connect to the database.>
- **Password** = <The password of the user.>
- **PasswordEncrypted** = <Enter either "True" or "False".>

  If the value is set to "True", then the user has to enter the password in encrypted format.

  If the value is set to "False", then the user has to enter the password in plain text.
- **AdvanceSettings** = <The advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection. Example: Connection Timeout = 20;>
- **DatabaseName** = <The name of the database that can be used as a datasource.>

  Consider the following sample command.

```
AVEVAWorkTasksConfigurator.exe ConfigurationType="DATASOURCE"
DataSourceName="TelematicsDataSource" Description="Telematics Datasource"
ServerName="Server1" AuthenticationType="SQLServer" UserId="skeltalab\user1"
Password="****" PasswordEncrypted="false" DatabaseName="TelematicsDB"
```

Note: Before you copy the sample command to the Command Prompt, it is recommended that you verify this command in a Text Editor.

The progress of the Data Source configuration is displayed in the **WTLogger Console**.

For more information about creating a datasource, see Creating a repository datasource.

Configure Repository in Silent Mode Using Command-Line Parameters

A Repository contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

You can configure a repository using the **AVEVAWorkTasksConfigurator** Console Application.

1. Open the Command Prompt in the Administrative mode.
2. Enter the path for the "Bin" folder, which is available in the path where the product is installed, and then press **Enter**.

   For example: CD <AVEVA Work Tasks Install Path>\Bin
3. On the Command Prompt, type **AVEVAWorkTasksConfigurator.exe**.
4. Pass the values for the following command line arguments:
   - **ConfigurationType** = **REPOSITORY.**

     The parameter used to configure a repository.
   - **RepositoryDataSource** = <The name of the datasource on which the repository has to be created. Ensure

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

that the specified datasource is already configured in the system.>

- **RepositoryName** = <The name of the repository.>
- **Description** = <The description for the repository.>

Consider the following sample command.

```
AVEVAWorkTasksConfigurator.exe ConfigurationType="REPOSITORY"
RepositoryDataSource="TelematicsDataSource" RepositoryName="TelematicsRepo"
Description="Telematics Repository"
```

Note: Before you copy the sample command to the Command Prompt, it is recommended that you verify this command in a Text Editor.

5. Enter the values for other parameters in the **WWSKBPMConfig.xml** configuration file. For more information about the parameters in the **WWSKBPMConfig.xml** configuration file, see Configure in Silent Mode Using XML.

Note:
- If you enter the password for the **Resource Provider** fields in plain text format in the XML, the password will be updated in the XML in an encrypted format, after the repository creation.
- If you create a repository in silent mode, the administrator for this repository is not configured. You must manually configure the Administrator for this repository and then perform an "iis reset".
- The maximum number of repositories, which can be created, depends on the permitted limit for the license.

The progress of the repository configuration is displayed in the **WTLogger Console**.

For more information about creating a repository and creating an Administrator for the repository, see Creating a repository.

Return Results for Silent Configuration

The return results are based on the outcome of the execution of **AVEVAWorkTasksConfigurator.exe**.

For example: If the value of the **ConfigurationType** parameter in the **WWSKBPMConfig.xml** configuration file is defined as **DataSource**, then the return result is displayed as "0" if the DataSource configuration is successful.

Similarly, based on the outcome for the selected configuration type, the return result is displayed. For more information about the **ConfigurationType** parameter in the **WWSKBPMConfig.xml** configuration file, see Configure in Silent Mode Using XML.

1. Open a Notepad file and add the following script.

```
echo off
cd <AVEVA Work Tasks Install path>\Bin AVEVAWorkTasksConfigurator.exe
echo %ERRORLEVEL%
```

2. Enter the appropriate value for **<AVEVA Work Tasks Install path>**.

   For example: *C:\Program files\AVEVA\Work Tasks\Bin*

3. Save the file with **.bat** extension.
4. Open the Command Prompt in the Administrative mode.
5. Type the path where the Batch file is saved, and then press **Enter**.
6. Type the name of the Batch file, and then press **Enter**.

   The return result of the **AVEVAWorkTasksConfigurator.exe** is stored in the **%ERRORLEVEL%** variable. The return result will be one of the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **0** - If the configuration is successful.
- **1** - If the configuration is not successful.
- **2** - If multiple instances of **AVEVAWorkTasksConfigurator.exe** are running.

Sample Batch File Script for Installing and Configuring in Silent Mode

You can install AVEVA Work Tasks in silent mode using the following sample Batch File script.

If you want to execute installation and configuration commands sequentially, then perform the following:

1. Open a Notepad file and add the following script.

```
Echo off
REM Script to execute silent AVEVA Work Tasks installation and configuration.
REM Configuration like creation of farm database, Datasource and repository.
REM
.....................................................................................
.............................
REM Below Command to execute AVEVA Work Tasks installer Silently in the default path
REM
.....................................................................................
.............................
echo AVEVA Work Tasks silent installation started......
REM.......................STEP 1 - BPM Installation
Silently.....................................................
<Path> AVEVA Work Tasks 2023 SP1.exe /S /v/qn
REM If above command is successful %ERRORLEVEL% will be zero. Non zero number else.
IF %ERRORLEVEL% EQU 0 (
   echo AVEVA Work Tasks silent installation completed successfully.
   REM The silent farm configuration executable required a configuration xml file.
   (WWSKBPMConfig.xml)
   REM The configuration file will be available in BPM installation "Bin" folder.
   (WWSKBPMConfig.xml)
   REM Get the configuration file from the installation location updated the
   mandatory nodes with relevant data.
   REM After the BPM installer installation is done silently, copy updated
   "WWSKBPMConfig.xml" AVEVA Work Tasks installation Bin folder.
   REM Below Command to copy updated "WWSKBPMConfig.xml" BPM installation Bin
   folder.
   REM ..................STEP 2 - Copy updated configuration file to installation
   Bin folder..........
   Copy /y <Path> WWSKBPMConfig.xml <AVEVA Work Tasks Install path>\Bin
   REM Below command to execute silent farm configuration executable.
   echo AVEVA Work Tasks silent farm configuration started....
   REM ..................STEP 3 - Move to the installation folder and call
   AVEVAWorkTasksConfigurator.exe ..........
   REM cd C:\Program files\AVEVA\Work Tasks\Bin
   cd <AVEVA Work Tasks Install path>\Bin AVEVAWorkTasksConfigurator.exe
   ConfigurationType="ALL"
   IF %ERRORLEVEL% EQU 0 (
   echo Silent farm configuration completed successfully.
   ) ELSE (
   echo Silent farm configuration failed. Please verify mandatory parameters in
   "WWSKBPMConfig.xml" and see WWSKLogger for details.
   )
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
) ELSE (
   echo AVEVA Work Tasks silent installation failed.
)
   REM AVEVA Work Tasks Installation and configuration completed.
```

2. Enter the appropriate value for **<path>**.

   For example: *C:\EXEPath\AVEVA Work Tasks 2023 SP1.exe /S /v/qn*

3. Enter the appropriate value for **<AVEVA Work Tasks Install path>**.

   For example: *Copy /y C:\ConfigFilePath\WSKBPMConfig.xml C:\Program files\AVEVA\Work Tasks*\\Bin

4. Save the file with **.bat** extension.
5. Open the Command Prompt in the Administrative mode.
6. Type the path where the Batch file is saved, and then press **Enter**.
7. Type the name of the Batch file, and then press **Enter**.

   The installation and configurations are performed sequentially.

Note:
- You can configure the other parameters for Farm, Data Source, and Repository either in the Batch file or in the **WWSKBPMConfig.xml** file.
- If the AVEVA Work Tasks installation is successful, then the value for **%ERRORLEVEL%** is zero; else, the value is any number except zero.

Limitations and Exceptions - Silent Installation and Configuration

- Creation of AVEVA Work Tasks Farm and Data Source through Connection Provider Assemblies is not supported.
- If you enter the password for the **Resource Provider** fields in plain text format in the Configuration XML, the password will be updated in the XML in an encrypted format, after the repository creation.
- If you create a repository in silent mode, the administrator for this repository is not configured. You must manually configure the administrator for this repository and then perform an "iis reset".
- The maximum number of repositories, which can be created, depends on the permitted limit for the license.

Installing AVEVA Work Tasks Add-in for Microsoft Outlook

The installation of AVEVA Work Tasks Add-in for Microsoft Outlook is supported only on machines where Microsoft Outlook 2016, Microsoft Outlook 2019, and Office 365 are installed.

Based on the version of Microsoft Outlook installed in your environment, the corresponding AVEVA Work Tasks Add-in for Microsoft Outlook is installed.

Example: If Microsoft Outlook 2016 is installed in your environment, the AVEVA Work Tasks Add-in for Microsoft Outlook 2016 will be installed.

Note: To install the latest AVEVA Work Tasks Add-in for Microsoft Outlook, uninstall the previous version and perform a fresh installation. Upgrade to latest version of AVEVA Work Tasks Add-in for Microsoft Outlook is not supported.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation Prerequisites

Before you install the **AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook**, ensure that you have installed **AVEVA Work Tasks** on your environment.

For more information about installing **AVEVA Work Tasks**, see Installing Different Editions.

Install AVEVA Work Tasks Add-in for Microsoft Office Outlook

Use the **AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook** installer to install AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook

1. Run the **AVEVA Work Tasks 2023 SP1-Outlook Addins.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook** screen appears.
2. In the **Welcome to the Installshield Wizard for AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook** page, click **Next**.

   The Customer Information dialog box appears.
3. Enter the **User Name** and **Organization** name.
4. Under **Install this application for**, select one of the following:
   - **Anyone who uses this computer (all users):** If the application can be accessed by all the users using this computer.
   - **Only for me (localadm):** If the application can be accessed by only the local administrator of the computer.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. In the InstallShield Wizard Completed dialog box, click **Finish**.

Uninstalling AVEVA Work Tasks Add-in for Microsoft Office Outlook

You can remove AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook in one of the following ways:

- Using the Add or Remove Program Option
- Using the AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook Installer

Using the Add or Remove Program Option

1. On the Windows **Start** menu, click **Control Panel**.
2. In the All Control Panel Items dialog box, click **Programs and Features**.
3. In the Uninstall or change a program dialog box, click **AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook**.
4. Click **Uninstall**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

A confirmation message appears.

5. Click **Yes** to remove **AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook**.

   A warning message appears.

6. Click **OK** to continue.

   A confirmation message appears.

7. Click **OK**.

Using AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook Installer

1. Run the **AVEVA Work Tasks 2023 SP1 Add-in for Microsoft Office Outlook** installer.

   The splash screen, and then the Welcome page appears.

2. Click **Next**.

   The Program Maintenance dialog box appears.

3. Select **Remove**, and then click **Next**.

   The Remove the Program dialog box appears.

4. Click **Remove**.
5. On the InstallShield Wizard Completed dialog box, click **Finish**.

Install Client on Windows Form Based Application

Client Installation on Windows

You can install the client and enable AVEVA Work Tasks functionality on a Windows Form-based application.

In a typical Server and Client installation, the Server Controls are used from the Enterprise Console Web Site. These Server Controls are by default implemented within the Enterprise Console Web site. For more information about the Server Controls, see Architecture.

Windows Form-based applications run on standalone systems and do not require Web browsers. In a Windows Form-based environment, AVEVA Work Tasks functionality can be implemented using two approaches.

The first approach is to embed the Server Controls through a browser control within the Windows form. This is similar to using a Web application in which AVEVA Work Tasks is implemented. The alternative approach is to use AVEVA Work Tasks run-time functionality using AVEVA Work Tasks APIs. However, the APIs can be used only with run-time functionalities. The design time functionalities such as defining a Queue, a Calendar and designing the process using process designer will not be possible with the APIs. The run-time APIs include:

- Getting information of a task or a work item getting assigned to a user, based on the defined process
- Taking action on a task or a work item. For example, approving or rejecting a task.

The following scenarios describe two distinct ways of integrating Windows form-based clients using AVEVA Work Tasks run-time API.

Scenario 1

In this scenario, the Client is installed on the same machine where the Windows form-based client is running. The number of the Client installation required will be equal to the number of Windows form-based applications. The Windows form-based applications make use of AVEVA Work Tasks run-time APIs to communicate with the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Client.

Example

A machine with 10 windows form-based applications will require 10 Client installations.

Scenario 2

In this scenario, Windows form-based application will interact or communicate with a single Client using Web service calls. The AVEVA Work Tasks run-time APIs will be implemented within this Web service calls. This Web service is not a part of the Client installation. This is the most preferred approach as the the Client is installed only on one machine.

Example

*A machine with 10 w*indows form-based applications will require a single Client installation.

Installation Planning Checklist

The installation planning checklist summarizes the tasks that you must perform to complete an installation. If you install the AVEVA Work Tasks products on more than one machine, complete the first installation using the detailed instructions in this guide. You can use this checklist to perform subsequent installations. Before you install the AVEVA Work Tasks products, prepare the machines where you plan to install the AVEVA Work Tasks products.

Complete the following pre-installation tasks:

- Integrate email server
- Identify server roles
- Select the database type
- Configure database user settings
- Identify user source
- Set up the communication protocol
- Set up the default ports

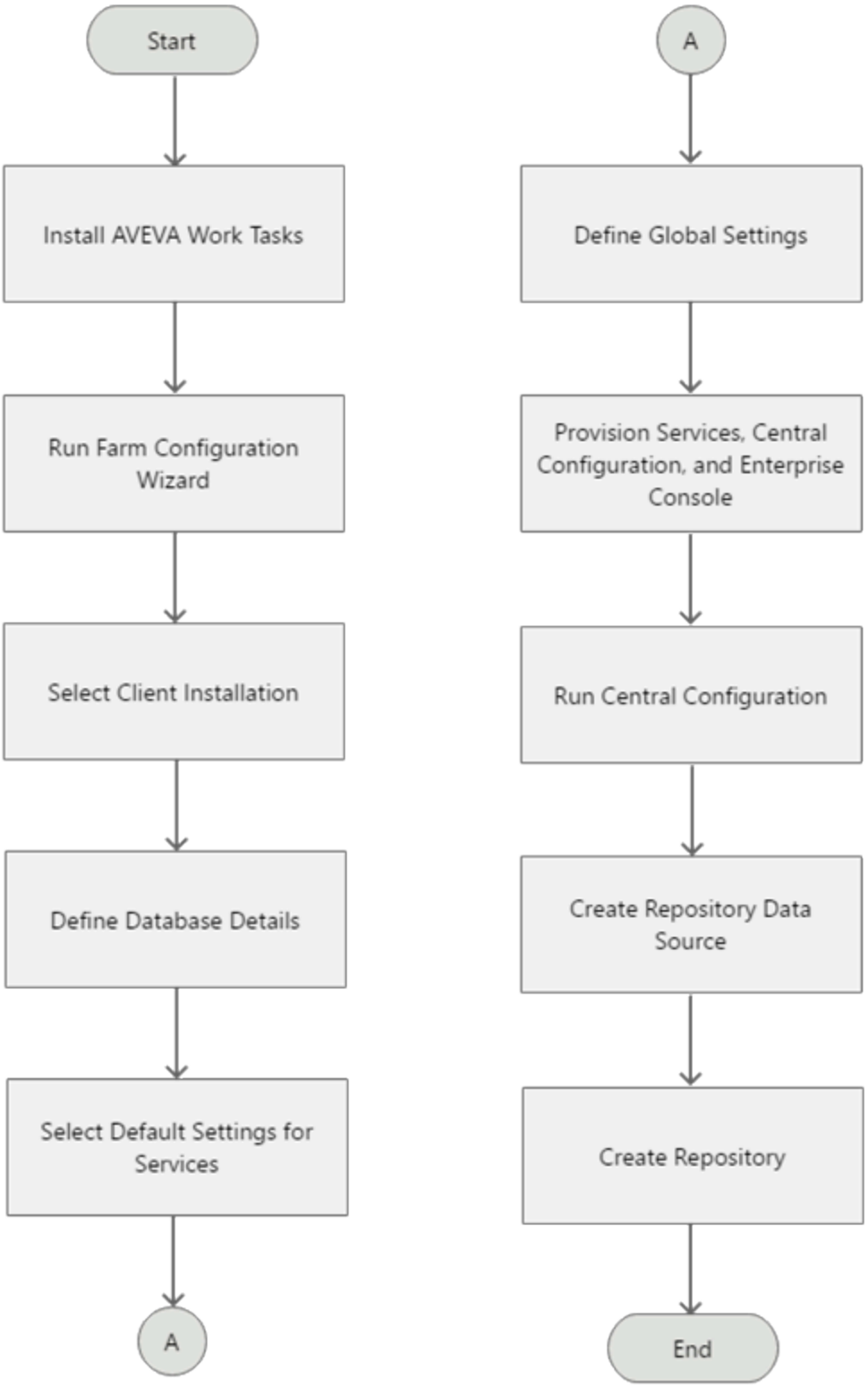
Before installing ensure all Prerequisites are taken care.

High-Level Installation View

This section provides a high-level view of the Client installation. The Client components are installed on Web Servers. For more information about this scenario, see Wonderware Skelta BPM Professional Edition.

Before continuing with the installation it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation

This section provides installation steps for the scenario **Installing AVEVA Work Tasks on a Windows Forms Based Application**.

Install AVEVA Work Tasks

You can use the AVEVA Work Tasks installer to install the AVEVA Work Tasks product.

Note: Ensure that the **ISSetupPrerequisites** folder is available in the same folder as the AVEVA Work Tasks installer.

1. Run **AVEVA Work Tasks 2023 SP1.exe**.

   The installer splash screen and then the **AVEVA Work Tasks 2023 SP1 - InstallShield Wizard** screen appears.
2. In the **Welcome to the InstallShield Wizard for AVEVA Work Tasks 2023 SP1** page, click **Next**.

   The License Agreement dialog box appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.
4. Select the **Destination Folder**.

   By default, the product is installed at **C:\Program Files\AVEVA\Work Tasks**. You can use the **Change** button to install the product at a custom location.

   In case the product is installed in a custom path, the installation folder will inherit the access control restrictions of the Program Files folder.
5. Click **Next**.

   The Ready to Install the Program dialog box appears.
6. Click **Install** to begin the installation.
7. On the InstallShield Wizard Completed dialog box, click **Finish**.

Start the Farm Configuration Wizard

The AVEVA Work Tasks Farm Configuration Wizard starts immediately after the AVEVA Work Tasks installation is complete.

1. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
2. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
3. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

If the Farm Configuration Wizard is not started automatically due to any reason, you can manually start the wizard by performing the following:

1. Close the Farm Configuration Wizard window that is automatically opened.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
3. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The Welcome to the Farm Configuration Wizard screen appears.
4. Click **Next**.

   A dialog box appears listing the unchecked MSDTC settings and asking your confirmation to continue with the current MSDTC settings.
5. If you want to continue with the current MSDTC settings, click **Yes**.

   The Cryptography Settings window appears.
6. Configure cryptography settings and click **Next**.

   The Configure License Server window appears.

Select Client Installation

1. In the Select the type of installation window, select **Client**.
2. Click **Next**.

Connect to an Existing Farm Database

You can use the Connect to existing database dialog box to connect to the existing Farm. You can select the Connection String or Connection provider for the new database.

1. Select the computer name on which the Farm Database is installed from the **Server Name** field.
2. Select **Connection String**.
3. Enter the following details:
   a. **Authentication Type**: This field is used to provide information on the process used to validate a user. AVEVA Work Tasks solution supports both Windows and SQL server authentication process. Example: Windows.
   b. **User ID**: This field is used to enter the SQL user ID for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   c. **Password**: This field is used to enter the SQL password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
   d. **Advanced Settings**: This is an optional field. It is used to enter any additional connection information such as connection Timeout etc. Example: Connection Timeout = 20.

      While creating a data source for repository, use the following custom settings to set properties for the connection pool.

      encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;

      encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

      enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation a connection which is used from a transaction is returned back to the connection pool only when the transaction is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks controls the transaction from escalating to a distributed transaction which is very expensive.

The above property can be used to turn off this feature. The default value is set as True.

connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds.

maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

Note: These settings are case sensitive and should be mentioned only in lower case.

e. **Database**: This field is used to select the machine with database installed. Example: FarmDB.

AVEVA Work Tasks will validate if at least one server is installed.

4. Click **Test Connection** to ensure database connectivity.

   You must have the following permissions on the Database Server:

   SELECT, INSERT, UPDATE, and DELETE permissions.

   EXECUTE permission for Stored Procedures.
5. Click **Next** to continue with the wizard.
6. Click **OK**.

Define and Provision the Enterprise Console

You can define the required AVEVA Work Tasks services.

Steps to Define Enterprise Console

1. In the **Services/Global Settings** page, the **Use default settings for services** option is selected by default.
2. Click **Next**.

   The **Installing Services** screen appears, indicating the progress of installation.

   A message *Installation is Complete* appears after all the services are installed.
3. Click **OK**.

Verify the Installation

After you install AVEVA Work Tasks, verify that the product is installed successfully. To verify successful installation, create a simple Workflow using Process Designer. The tasks includes the following:

- Creating a Workflow
- Publishing the Workflow
- Testing the Workflow

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Creating a Workflow

1. Log on to the **Enterprise Console**.
2. If the Repository is configured for Active Directory provider, go to step 4.
3. Enter the **User Name** and **Password**.
   The **User Name** and **Password** will be validated with the user provider selected while creating the repository.
   The default authentication provider is the Repository User Provider List.
4. In the **Enterprise Console**, click menu icon.
   All features can only be available for a user with administrative privileges.
5. From the menu, select **Workflows**.
   The Workflow window appears.
6. On the Action bar, click **New**, and then click **Workflow**.
   The New Workflow window appears.
7. Enter **Title** and **Description**.
8. Click **Save & Continue**.
   The successful message appears.
9. Select **Design the Workflow now**.
   This is the default option.
10. Click **Finish**.
    The Process Designer window appears.
11. From the activities pane on the left, drag the Information activity to the Process Designer window.
12. The activity *Information1* is created.
    Click to create a link between two activities.
    The Link Action window appears.
    a. Select from the **From Action** drop-down box.
    b. Select from the **To Action** drop-down box.
    c. Click **OK**.
       The link is established.
13. Right-click *Information1*.
14. Select **Activity Properties** from the context menu.
    The Activity Properties pane appears.
    a. Enter **Name** and **Description**.
    b. Click to **Assign Actor(s)**.
       The Assign Actor(s) window appears.
    c. Select the required provider from the **Providers** drop-down list.
    d. Select values from the drop-down list.
    e. Click **Add**.
    f. Click **Update**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

15. Click to enter the **Subject** for Notification Contents.
    The window to enter subject for notification appears.
16. Click **Update**.
17. Click to enter the **Body** for Notification Contents.
    The window to enter the Body content appears.
18. Click **Update**.
19. From the Activity Properties pane, click **Save**.

Publishing the Workflow

1. On the **Publish** menu, click **Publish**.
   The Published successful message appears.
2. Close the dialog box.
   The Enterprise Console screen lists the workflow created.

Testing the Workflow

1. Select the listed workflow.
2. Right-click the selected workflow.
3. Click **Publish and Test Run**.
   The **Status Finished successfully** appears.
   If Status is **Execution Pending** in the Process Execution window, then click on the **Refresh** button to get the status.
4. In the **Enterprise Console** menu, select **Inbox**.
   The Inbox lists the work item.
5. Click to view the work item listed.
   This indicates that the installation is successful.

Install AVEVA Work Tasks Locales for Brazilian Portuguese

AVEVA Work Tasks supports Brazilian Portuguese operating system out of the box. A separate installer for Brazilian Portuguese locales is available which can be installed after installing **AVEVA Work Tasks 2023 SP1**. Before you run the Brazilian Portuguese installer, we recommend you to back up the existing files and folders of the previous installation.

Prerequisites

- Install **AVEVA Work Tasks 2023 SP1**. For information about installing AVEVA Work Tasks, see Installing Different Editions.
- Back up the following folders before you run the locales installer:
  - *C:\Program files\AVEVA\Work Tasks\\WorkflowElements\Default\en-US\Actions\XML*
  - *C:\Program files\AVEVA\Work Tasks\\WorkflowElements\Default\en-US\Templates*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to install

You can use the **AVEVA Work Tasks Brazilian Portuguese Localization** installer to install the **AVEVA Work Tasks** product in Brazilian Portuguese.

1. Run the **AVEVA Work Tasks 2023 SP1-pt-BRLocale.exe**.

   A message to overwrite the files appear.
2. Click **Yes** to confirm the overwrite and continue the installation.

   The **Welcome to the Installation Wizard for AVEVA Work Tasks Brazilian Portuguese Localization** page appears.
3. Click **Next**.

   The install page appears.
4. Click **Install** to begin the installation.
5. On the InstallShield Wizard Completed dialog box, click **Finish**.

   AVEVA Work Tasks is successfully installed in Brazilian Portuguese.

After you install

Add the following entry within the **<system.web>** tag in the **web.config** file of the **Enterprise Console**:

```
<globalization culture="pt-BR" uiCulture="pt-BR" requestEncoding="utf-8"
responseEncoding="utf-8"/>
```

Set a Repository to Support Multiple Locales

AVEVA Work Tasks supports setting a repository supporting multiple locales. You can set this up both in a new environment as well as in an existing environment.

New Installation

1. Install **AVEVA Work Tasks**.

   - Follow the Installation.
2. Create or use an existing **Farm Database.**

   - **New Farm Database:** Follow the steps to create a new farm.

   - **Existing Farm Database**: Follow the steps to use an existing farm.
3. Create a **Data Source**.

   - Follow the steps to create a data source.
4. Create a **Repository**.

   - Follow the steps to create a repository.
5. Create multiple **Enterprise Console** for different locales.

   - Each **Enterprise Console** support a locale, therefore create a console for each locale. Also, ensure to setup the appropriate authentication for the console based on the environment.

   - Follow the steps to create an **Enterprise Console**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Map **Enterprise Console** to the repository.

   - The consoles created for the different locales must be mapped to the same repository.

   - Follow the steps to map **Enterprise Console**.

7. Set up locales for **Enterprise Console**.

   - Follow the steps to set up locales for **Enterprise Console**. See *Steps to Localize AVEVA Work Tasks* in the *Build Business Process* section.

Existing Installation

1. Create multiple **Enterprise Console** for different locales.

   - Each **Enterprise Console** support a locale, therefore create a console for each locale. Also, ensure to setup the appropriate authentication for the console based on the environment.

   - Follow the steps to create an **Enterprise Console**.

2. Map **Enterprise Console** to the repository.

   - The consoles created for the different locales must be mapped to the same repository.

   - Follow the steps to map **Enterprise Console**.

3. Set up locales for **Enterprise Console**.

   - Follow the steps to set up locales for **Enterprise Console**. See *Steps to Localize AVEVA Work Tasks* in the *Build Business Process* section.

AVEVA Licensing

The AVEVA Enterprise Licensing system is a common platform that allows you to effectively and efficiently manage AVEVA Software product licenses. AVEVA Enterprise licensing system is composed of a browser-based License Manager and a License Server that together allow you to share and deliver licenses for your installed products.

More information about AVEVA Enterprise Licensing and how it works is available in following topics:

- About AVEVA Enterprise Licensing
- How does Licensing work?
- Licensing for AVEVA Work Tasks

About AVEVA Enterprise Licensing

License Manager

License Manager allows you to quickly access and maintain licenses for all AVEVA Software products in your different environments by means of its scalable, flexible design features:

- Browser-based for scalability and ease of use; can be remotely accessed by any web browser.
- Light-weight standalone software you can install on the same node as the License Server computer, or on any other node based on your deployment needs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Manage one or multiple License Servers to best organize and manage the licenses in your unique environment.

License Server

License Server provides all the functionality to acquire, store, maintain, and serve licenses to your installed AVEVA software.

- Where all licenses and related information is stored and maintained.
- Securely serve any types of software application being licensed, including Windows browsers, tablets, and mobile devices.
- Provide current license usage information.

Installed Products

After the AVEVA Software product is installed it relies on the licensing system to enable its functionality. During product installation you decide from which License Server you want the product to get its license. If a License Server is installed on the same computer, it will use that one as the default. You can decide to install the License Server on a different computer to point to a single License Server for all installed products, or different License Servers installed for different groups of installed products. All the products installed on the same computer would point to the same License Server.

For more information about AVEVA Enterprise Licensing, refer the AVEVA Enterprise Licensing documentation available in *<InstallPath>\Program Files (x86)\Common Files\ArchestrA\Licensing Framework\License Manager*.

How does Licensing work?

This section describes the licensing workflow. This example assumes that both License Manager and License Server are installed on the same computer, and that the computer can connect to the Internet.

1. **Purchase one or more AVEVA Software products**: Along with your installation media or downloads, you will receive one or more entitlement files containing licenses details.
2. **Start License Manager**: It opens in a browser window and accesses the installed License Manager components.
3. **Select the License Server**: Use the License Manager to select an available server to host the licenses you want to activate.
4. **Import the entitlement file**: Import the entitlement file. After importing, the License Manager displays the available licenses.

   The Entitlement File contains information for specific product licenses that is imported into the License Manager to activate purchased licenses.
5. **Activate licenses**: License Manager contacts the AVEVA Activation Server over the Internet to activate the selected licenses on the License Server.

   The Activation Server activated licenses on the License Server.
6. **Manage your licenses**: Licenses are now available for the products you want to use. You can now:
   - activate new licenses

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- deactivate current licenses
- reserve and unreserve licenses

7. **Install the purchased products**: You can go ahead and install the products following the individual product installation instructions. Also make sure you have at least one license server and one license manager installed as well
8. **Configure and Run your products**: The installed products, when running, will acquire their licenses from the License Server and will release them when no longer needed.

For more information about Licensing, refer the AVEVA Enterprise Licensing documentation available in *<InstallPath>\Program Files (x86)\Common Files\ArchestrA\Licensing Framework\License Manager*.

Licensing for AVEVA Work Tasks

Prerequisites

- Ensure that the AVEVA License Manager and AVEVA License Server is installed on a machine, which is on the local network and can be accessed by all the AVEVA Work Tasks servers and clients.
- In the AVEVA License Configurator, ensure that you provide the details of the machine where the License Manager and License Server are installed.
- Ensure that the AVEVA LICAPI is installed as a prerequisite during the AVEVA Work Tasks installation.

Types of Licenses

The following are the types of licenses and their precedence, when all these licenses are available together:

1. Single Server
2. Load Balanced Server

Scenarios

- If load balance is enabled during Farm configuration, then the Load Balanced Server license is acquired, based on its availability. The Advance Server Service is installed and is enabled.
- If load balance is disabled during Farm configuration, then either the Single Server license is acquired, based on the availability. The Advance Server Service is installed and is disabled.
- To use Forms Designer, the Forms Server license has to be acquired. The Forms Server license is released when the Client service is stopped. The Forms license should be activated on all Web server(s) rendering the Forms.
- If the **corecount** license is activated and acquired along with any of the Standard licenses, then the client connection license is not checked, and there is no limit for the number of user connections.
- If "CPU Cores" count license is activated, then the "CPU Cores" count must depend on the system CPU Core count.

  For example: If the system has a 4 Core Processor, then the "CPU Cores" count in the license must be 4 or above.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Services

Workflow Engine Service

- The Workflow Engine service starts only when the following conditions are met:
  - The license is activated.
  - The **concworkflow** and/or **developer** features are available.
- When you start the Workflow Engine Service, the **concworkflow** and **corecount** license features will get acquired based on the License type. However, after this service is stopped, the license features acquired above are released.
- Before starting the Workflow Engine Service, the license availability is checked. If any server license is not activated and available, then the Workflow Engine Service is not started.
- The Workflow Engine Service checks the license periodically i.e. at an interval of 24 hours.
  - Consider a scenario where the user deactivates the server license when the license has been activated and the Workflow Engine service is running. After 24 hours, if the Workflow Engine Service is still running but the license is deactivated, then the grace period for the license is activated. A warning message is displayed to the Administrator through an exception email. The Workflow Engine Service service checks for license activation every day. If the license is not yet activated after the 7th day, then the Workflow Engine Service service is stopped. If the license is activated, then the Workflow Engine Service service will continue to run. .
- The licenses acquired by the Advance Server service and the Workflow Engine Service will be released when the Client Service is stopped.
- You can log on to the Enterprise Console only if the Client service and the Workflow Engine Service are running.

Grace Period for AVEVA License

The Workflow Engine Service checks the license periodically i.e. at an interval of 24 hours.

Consider a scenario where the user deactivates the server license when the license has been activated and the Workflow Engine service is running. After 24 hours, if the Workflow Engine Service is still running but the license is deactivated, then the grace period (of 7 days) for the license is activated.

In this scenario, a warning message is displayed to the Administrator through an exception email. Also, the message "Your system is currently running on a grace period" is displayed on the following:

- Enterprise Console
- Central Configuration
- Farm Configuration wizard
- Service Monitor > Workflow Engine Service

The Workflow Engine service checks for the license activation every day. If license is not yet activated after the 7th day, then the Workflow Engine Service is stopped. However, if the license is activated, then the Workflow Engine Service will continue to run.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Change Load Balance Settings

You can change load balance settings for AVEVA Work Tasks Server installation.

To Change Load Balance Settings

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
2. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** screen appears.
3. Click **Next**.
4. In the **Edit AVEVA Work Tasks Farm Settings** window, select the **Load Balance Settings** check box.
5. Click **Next**.

   The **Load Balance Settings** window appears.
6. Perform one of the following:
   - If load balancing is not enabled on the server, select the **Enable load balancing** check box, to enable load balancing.
   - If load balancing is enabled on the server, clear the **Enable load balancing** check box, to disable load balancing.

   Note:
   - Before you disable load balancing, ensure that you manually stop all the AVEVA Work Tasks Services.

   - If an Enterprise License is not available in the License Server, then a warning message appears indicating that the Enterprise License is not available.
7. Click **Next**.

   The load balance settings are configured successfully.

Client Connection License for System Platform IDE

In System Platform IDE, the Workflow Gateway object must be selected to browse the Enterprise Console menu. After the Enterprise Console menu is loaded in the System Platform IDE, any page can be loaded. Before the page is loaded, the Client Connection License is acquired. If the license is available, then the selected page is loaded successfully. However, if the license is not available, a blank page is displayed.

For multiple users to browse the AVEVA Work Tasks controls in System Platform IDE from the same device, separate Client Connection Licenses should be acquired..

Note:
- You can view the details of the consumed licenses from the **Usage Summary** tab in the **AVEVA License Manager Portal**.
- If you have acquired the license to access the controls and then you log out or exit, the Client Connection License is not released. The acquired license is available only after 20 minutes for the other user to use.

The Device ID maintained for the connection is a combination of the following:

- IP address (IPV6 or IPV4) of the machine from which the user logs on to the Enterprise Console
- IDE

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For example: If user "John" logs in to the Enterprise Console in the System Platform IDE, then the Device ID is displayed as *10.13.200.120$IDE*.

Where:

- IP Address is 10.13.200.120
- IDE
- $ is the separator

Releasing Client Connection License for System Platform IDE

The connection license for System Platform IDE gets released when the session timeout occurs.

Client Connection License for InTouch

When you browse the AVEVA Work Tasks controls as graphics in InTouch, the Client Connection License is acquired before the page is loaded. If the license is available, then the selected page is loaded successfully. However, if the license is not available, a blank page with an appropriate message is displayed.

For multiple users to browse the AVEVA Work Tasks controls in InTouch from the same device, separate Client Connection Licenses should be acquired.

Note:
- You can view the details of the consumed licenses from the **Usage Summary** tab in the **AVEVA License Manager Portal**.
- If you have acquired the license to access the controls and then you log out or exit, the Client Connection License is not released. The acquired license is available only after 20 minutes for the other user to use.

The Device ID maintained for the connection is a combination of the following:

- IP address (IPV6 or IPV4) of the machine from which the user logs on to the Enterprise Console
- IDE

For example: If user "John" logs in to the Enterprise Console in the System Platform IDE, then the Device ID is displayed as *10.13.200.120$IDE*.

Where:

- IP Address is 10.13.200.120
- IDE
- $ is the separator

Releasing Client Connection License for InTouch

The connection license for InTouch gets released when the session timeout occurs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Session-based Client Connection License

Acquiring Client Connection License

- The maximum number of connections which can be active at a time depends on the value set for the AVEVA Concurrent Client Connection license limit. Each user logging in to an Enterprise Console will consume a license. If a user logs in through different browsers or repositories, then each login consumes one license.

  For example: If you have purchased the Client Connection License with a Quantity of 20 Pack, a maximum of 20 connections can be active at a time.

  Note:
  - You can view the details of the consumed licenses from the **Usage Summary** tab in the **AVEVA License Manager Portal**.
  - If the AVEVA Core Count license is activated, then the client connection license will not be consumed when the user logs on to Enterprise Console.
  - If the license limit is exceeded when logging in with the AIM based Active Directory provider, an alert message will be displayed. After clicking the **OK** button, the user will remain on the same page.

- The Unique Device ID maintained for the connection is the Host name or IP address of the machine from which the user logs on to the Enterprise Console. The user name for the connection is the user ID string used for logging on to the repository. For example: admin, corp\user1

  For example: If user "admin" logs in to the Enterprise Console in the Google Chrome browser, then the **Device** is displayed as *<Unique Device ID>_<Appended Unique Characters>* and **User** is displayed as *admin*.

- After a license is acquired and the user logs on to Enterprise console, the license gets acquired again after every 5 minutes for an active session.

Releasing Client Connection License

The connection license gets released in the following scenarios:

- When a user logs out from the Enterprise Console by clicking the **Log Out** menu option. The license is released based on the Device ID and the user, as explained in the Acquiring Client Connection License section.
- If the user has configured the **Auto Log Off** setting while creating the repository, and if the user does not perform any activity in the Enterprise Console for the time period specified in this setting.

  For example: If the time period in the **Auto Log Off** setting is configured as 5 minutes. The user is logged off if the user session is inactive for 5 minutes.
- If the user has logged in to the Enterprise Console and if the user closes the browser window without clicking the **Log Out** menu option.

Note: In some cases, when a user closes the browser window without logging out, the Client Connection License is not released immediately. In such cases, the Client Connection License is released after 40 minutes.

Releasing Client Connection License – Customized Log out

If a user has customized the logout option instead of using the Enterprise Console logout, then the client

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

connection license consumed after logging in to the Enterprise Console is not released.

The user has to release the client connection license from their logout custom code.

The following code can be used to release the client connection license.

```
Skelta.Entity.UserContext UserCtx = new Skelta.Entity.UserContext();
string userName = UserCtx.GetUserName();
string deviceId = string.Empty;
if (HttpContext.Current.Request.Cookies[userName + "_skdeviceid"] != null)
{
deviceId = HttpContext.Current.Request.Cookies[userName + "_skdeviceid"].Value;
}
Workflow.NET.Security.Client.SELicensing.LicenseConnectionManager licManager
= new Workflow.NET.Security.Client.SELicensing.LicenseConnectionManager();
bool licenseReleased = licManager.ReleaseClientConnectionLicense(deviceId, userName);
if (licenseReleased)
{
HttpRuntime.Cache.Remove(deviceId);
Response.Cookies[userName + "_skdeviceid"].Expires = DateTime.Now.AddDays(-1);
}
```

AVEVA Work Tasks Pro

Work Tasks Pro is a premium mobile application for AVEVA Work Tasks. It is designed for business professionals who use a mobile phone or a tablet device to access work items, while on the move.

If you are an administrator, complete the following tasks to enable the users to use the Work Tasks Pro application in a mobile device:

1. Configure Azure Setting
2. Set MobileWorkList Delivery Channel

The product versions that Work Tasks Pro support are:

- AVEVA Work Tasks 2023 SP1
- AVEVA Work Tasks 2023
- AVEVA Work Tasks 2020 Update 2
- AVEVA Work Tasks 2020 Update 1
- AVEVA Work Tasks 2020
- Workflow Management 2019 Update 1
- Workflow Management 2019

Note: AVEVA Work Tasks supports JavaScript based on the ES5 (ES2014) standard, applicable for Forms.

AVEVA License Utilization

AVEVA Work Tasks Pro app acquires license from the Enterprise Console (EC) server to which it gets connected. When you log on to the app and get connected to the EC server, the EC server acquires a license for the app. The license gets released only when you log out from the app.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

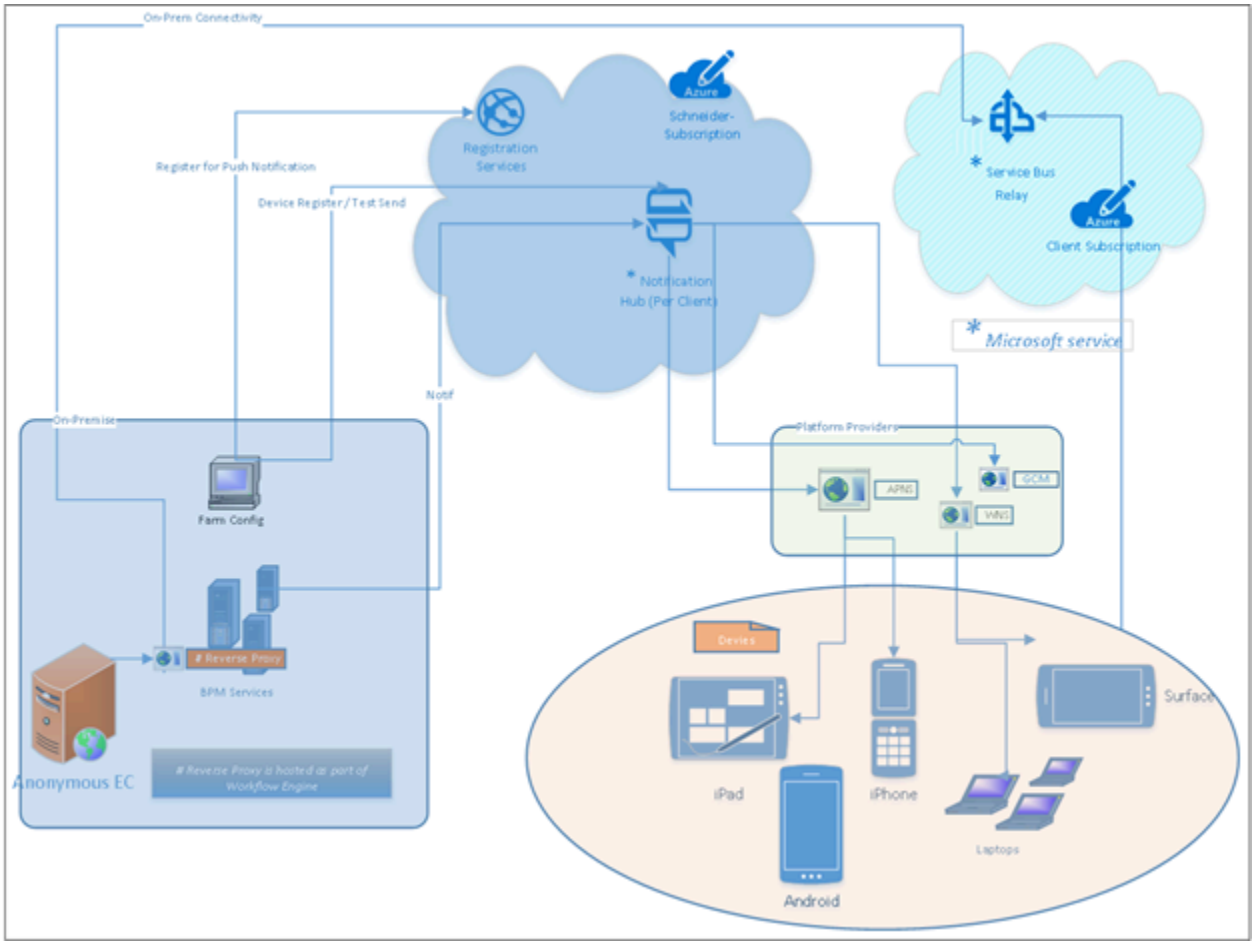
If the app is killed by the system (in case of iOS device system process based on factors listed in the Background Processing in iOS or in other devices due to system based behaviors) or done manually, while the user is logged-in, then the background process would not get triggered in case of iOS. So, the license would still be consumed and not released. But if you open the app, it will check if the auto-logoff needs to be triggered, then it would perform the same and release the license.

Also note that when performing logoff or auto-logoff is triggered (done by app when it is active or by the iOS background process), if the EC Server is not available, it would not be able to release the license. But if you log in with same or another user in the same device (which you can only when the server connectivity is available), then the license would get updated accordingly, rather than creating a new license.

Similar to auto-logoff, there is a session timeout for AIM, AVEVA Connect, and AAD Token Based repositories. The app internally refreshes the session tokens. But in case the EC server connectivity is not available at that moment, then it would show a session timeout message. In such cases, the user would be logged-out, but the license would not be released, as the EC Server connectivity is not available. As in the case of earlier behavior with auto-logoff, if you login with the same or another user in the same device (which you can only when the server connectivity is available), then the license would get updated accordingly, rather than creating a new license.

Push Notification Architecture for AVEVA Work Tasks Pro

The Push Notification of AVEVA Work Tasks Pro App comprises of Azure Notification Hub wherein, the mobile application communicates with the AVEVA Work Tasks Server through Azure Service Bus.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The following section explains how to achieve the desired functionality for mobile applications.

- **Push Notification**

  When a task is assigned to a mobile application user, a notification alert has to be sent to their mobile device.

  The mobile channel which is a part of the workflow engine, tries to communicate with a push notification engine called Notification Hub. This is a multi-platform, scalable, push notification engine meant for quickly sending millions of messages or alerts from any back-end to any mobile device.

  With a single cross-platform API call, you can easily send targeted and personalized push notifications to any mobile platforms from on-premise back-end.

  The mobile channel which is a part of the workflow engine, communicates with the notification hub, provisioned for each client. However, if the workflow engine fails to establish the communication, the notification alert information gets stored in the database. If notifications are not sent, the notification service will retry based on settings configured in Retry Settings.

  The local Mobile Notification Service, which gets deployed with the AVEVA Work Tasks installation, ensures reliable delivery of the stored messages. The local Mobile Notification Service polls the information and schedules it for delivery on the basis of Retry Settings. The reliable delivery mechanism ensures that all the stored messages are delivered and no messages are lost during the transmission.

- **Registration Services**

  Registration service enables AVEVA Work Tasks to send notifications. Hosted REST service for registration enables AVEVA Work Tasks to send notifications. For registration process, see Register to Mobile Notification Service.

  After successful registration and validation, the notification hub will be provisioned for each client and a key is assigned for each user. Based on the authentication, notification messages are sent.

  After the notification hub is provisioned, whenever AVEVA Work Tasks tries to send notification to mobile devices, communication happens only through notification hub. This hub is unique to each client and is responsible in communicating with the relevant platform provider and delivers the notification.

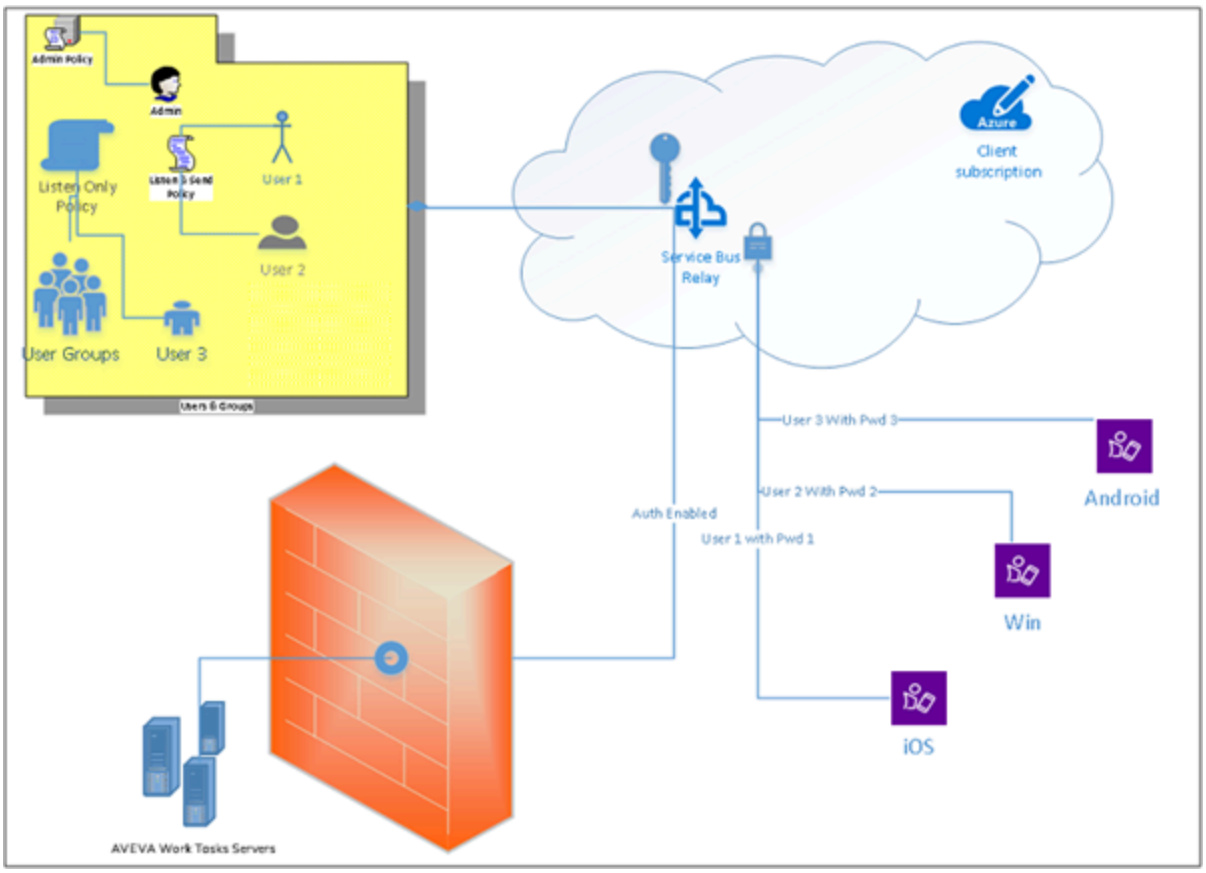
- **Platform Providers and Mobile Application**

  The platform providers deliver the notification alerts to the respective mobile devices, whenever a new task is assigned to the mobile application user. The alert is delivered to the mobile device service even when the mobile application is offline.

Enhanced Security

Any request sent from mobile to the server to fetch information might cause security threat. As a result, an enhanced authentication has been established for security.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



For more information on enhanced security, see Enabling Enhanced Security.

Install AVEVA Work Tasks Pro

AVEVA Work Tasks Pro is a premium mobile application for AVEVA Work Tasks. It is designed for business professionals who use a mobile phone or a tablet device to access work items, while on the move. It is an easy-to-use application that provides a quality visual experience.

Note:
- Sometimes the AVEVA Work Tasks Pro app on UWP does not get updated from the Microsoft Store from the AVEVA Work Tasks Pro page. We recommend you navigate to Library page in the Microsoft Store and update the app.
- AVEVA Work Tasks supports JavaScript based on the ES5 (ES2014) standard, applicable for Forms.

Capabilities

- View and act on tasks
- Receive real time notifications
- Fill form that initiates a process
- Approve or reject requests/events
- Flag tasks based on importance
- View work items based on categories

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Search for the assigned tasks
- Submit forms offline

Prerequisites

Complete the following prerequisites before you install and access AVEVA Work Tasks Pro on your mobile device.

Software Requirements

- To successfully configure and use the AVEVA Work Tasks Pro mobile applications, install **AVEVA Work Tasks** on your computer.

  For more information about installing **AVEVA Work Tasks**, see Installing Different Editions.
- Before installing AVEVA Work Tasks Pro on your device, verify if you have the right version of AVEVA Work Tasks configured.

Hardware Requirements

For more information about the supported versions of iOS, Windows, and Android, refer the *AVEVA Work Tasks 2023 SP1 Release Notes*.

Other Requirements

- Good Internet connectivity and a Service Bus Namespace if you want to access AVEVA Work Tasks Pro through Windows Azure.
- Good Internet connectivity to connect to Windows Azure if you want to receive Push Notification for AVEVA Work Tasks Pro.
- Port 80 or 443 to communicate to the Push Notification servers.
- Port 9350, 9351, 9352, or 9353 to communicate with Service Bus for the AVEVA Work Tasks Pro application. These ports are used by the AVEVA Work Tasks services, when the AVEVA Work Tasks Pro application is used outside Corporate Network.

  The Apple devices require separate ports for Push Notification. For more information, see Apple Push Notification service.

High Level View of Configuration and Installation of AVEVA Work Tasks Pro

This section provides a high-level view of the various configuration and installation steps required to use the Work Tasks Pro application. The objective of this section is to provide an overview of the various activities involved in this process. This section also enables you to visually keep track of your activities while following the sequence of tasks.

Each block in the flowchart below is associated with a step. Before continuing with the installation, it is presumed that you have read and understood the Prerequisites section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Access AVEVA Work Tasks Pro

The AVEVA Work Tasks Pro application can be accessed outside the corporate network through Windows Azure. The mobile application communicates with the AVEVA Work Tasks server which is within the firewall through Windows Azure. It is recommended to access the application through Windows Azure, as it provides high level of security as the application runs behind the firewall. Also, this setup does not require additional network settings to access the application on the go.

The speed at which the tasks are downloaded and can be acted upon, depends on the internet connectivity. Hence, is it recommended to have a good internet connectivity.

Following are the methods of accessing AVEVA Work Tasks Pro:

- Accessing AVEVA Work Tasks Pro within the Corporate Network (by making AVEVA Work Tasks host IP address static)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Accessing AVEVA Work Tasks Pro Outside the Corporate Network

To access AVEVA Work Tasks Pro through Azure, you will need to configure the Service Bus Listener. A Service Bus Listener is required to listen to the communication requests from the mobile devices to the Enterprise Console. After you configure a service bus listener, a **Service Bus Listener URL** is generated. This URL is to be provided to the mobile user to access the application in mobile devices.

Important: The Service Bus Listener URL must be unique for every Farm. If you are using the same URL for multiple Farms, ensure that the combination of the Listener Namespace and Service Bus Namespace is unique. For example, if the Service Bus Listener URL for a farm is *http://Telematics.servicebus.windows.net//MobileEC*, the combination of Listener Namespace (MobileEC) and Service Bus Namespace (Telematics) should not be used for any other farm configuration.

To configure the Service Bus Listener:

1. Use an existing Service Bus Namespace, if available or create a new Service Bus Namespace through Windows Azure Management Portal
2. Configure Service Bus Listener through the Farm Configuration Wizard (see Configure Service Bus Listener)

Note: You cannot configure AVEVA Work Tasks Pro to connect to the repositories configured with service bus and token services.

Access AVEVA Work Tasks Pro within the Corporate Network

1. Install AVEVA Work Tasks. For more information, see Install AVEVA Work Tasks.
2. Create the Enterprise Console with an anonymous authentication. For more information, see Create Enterprise Console with Anonymous Authentication.

   Note: We recommend to use HTTPS protocol while creating Enterprise Console with anonymous authentication for a secure connection.
3. Configure the AVEVA Work Tasks Mobile Notification Service settings. For more information, see Configuring Wonderware Skelta BPM Mobile Notification Service Settings.
4. Enter the Enterprise Console intranet URL in URL field of AVEVA Work Tasks Pro.

Access AVEVA Work Tasks Pro Outside the Corporate Network

1. Install AVEVA Work Tasks. For more information, see Install AVEVA Work Tasks.
2. Create the Enterprise Console with an anonymous authentication. For more information, see Create Enterprise Console with Anonymous Authentication.
3. You can make Enterprise Console accessible outside the corporate network in either of the following two ways:
   - Host the local Enterprise Console intranet URL on the internet.
   - Configure Azure URL. For more information, see Configuring Azure Setting.
4. Configure the AVEVA Work Tasks Mobile Notification Service settings. For more information, see Configuring Wonderware Skelta BPM Mobile Notification Service Settings.
5. Enter the Enterprise Console internet/Azure URL in URL field of AVEVA Work Tasks Pro.

For more information about setting up ports used by AVEVA Work Tasks services, when the AVEVA Work Tasks

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Pro applications are used outside Corporate Network, see Setting up Ports.

Install and Configure AVEVA Work Tasks Pro App

To install AVEVA Work Tasks Pro application

1. Download the **AVEVA Work Tasks Pro** application from the relevant store. The **AVEVA Work Tasks Pro** icon appears on the device screen.
2. Tap **Accept** to agree to the **End User License Agreement** and configure the application.

To configure AVEVA Work Tasks Pro application

To get started with using the AVEVA Work Tasks Pro app, you need to set the configuration.

Access Account

If you are opening the application for the first time, the **Manage Account** screen appears automatically.

- To access the **Manage Account** screen, swipe from the left edge of the screen, and then tap **Manage Account**.

Configure Server

- **Server Path:** Enter the URL of the Enterprise Console. For using the application with Azure, enter the Azure URL of the Enterprise Console.
- **Repository:** In the Repository list, tap on a repository name. When you tap on the repository control, the repositories and providers are populated.
- **Provider:** In the Provider list, tap on a provider name.

Configure Credentials

- **User Name:** Enter your user name.
- **Password:** Enter your password.

To access through Azure Active Directory

If your account is added as a federated user account to Azure Active Directory, you are automatically navigated to the Microsoft Login page (company login page).

To access through Azure Active Directory,

1. In the **Email** box, type your email ID.
2. Click **Sign In**.

Note:
- If you do not have the privileges to access Azure Active Directory, an error message appears indicating that you are not authorized to access this account.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To access through AVEVA Connect

If your account is added as a federated user account to AVEVA Connect, you are automatically navigated to the AVEVA Login page (company login page).

To access AVEVA Connect,

1. In the **Email** box, type your email ID.
2. Click **Sign In**.

   The AVEVA Connect Home page appears.

Note:
- If you do not have the privileges to access AVEVA Connect, an error message appears indicating that you are not authorized to access this account.

To access through AVEVA Identity Manager

To access the AVEVA Work Tasks Pro app using AIM authentication, enter the following details in the **AVEVA Identity Manager** login page:

- **User Name**: Enter the user name.
- **Password**: Enter the password.

To save Configuration

- **LOG ON:** Tap to save the configuration information and log in to the application.

To scan Account Settings using the QR Code

You can generate a QR code in the Farm Configuration Wizard to store settings such as Server Path, Repository, and Provider. You can then scan this QR code in the AVEVA Work Tasks Pro app to automatically fill these settings in the Manage Account screen.

1. Tap the **Scan QR Code** icon.
2. Scan this QR Code in the AVEVA Work Tasks Pro app.

   The values configured for the following settings in the Farm Configuration wizard are considered by default.
   - **Server Path**
   - **Repository**
   - **Provider**
3. Enter the user name and password.

To delete Account

You can clear all the settings configured for an account.

1. In the **Manage Account** screen, tap the Overflow Menu.
2. Tap **Delete Account**.

   All the settings configured for the account are cleared.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Verify Installation

After you install AVEVA Work Tasks Pro, verify the successful configuration and installation of AVEVA Work Tasks Pro on your mobile device. The following steps are based on the following assumptions:

- AVEVA Work Tasks is installed.
- Service Bus Listener URL has been generated.
- Users have been added (admin and John in this scenario).
- AVEVA Work Tasks Pro application has been installed and configured.

To verify the installation

1. In the **Enterprise Console,** click menu icon, and then click Workflows.
2. Create a *Leave Approval* workflow.
3. Add an **Approval** activity.

   Set its properties as explained below:

   a. **Name**: Enter a name for the activity, say *Leave Approval*.
   b. **Delivery Channel**: In the Delivery Channels window, select the **MobileWorkList** check box in the **Channels** section.

      Click **Save Channels**. This enables the activity to deliver the task to the Inbox in the mobile devices.
   c. **Assign Actor**: Set the actor to whom the task is to delivered. For example, admin.
   d. **Subject**: Set the subject for the leave approval task.
   e. **Body**: Set the Body text for the leave approval task.
4. Click **Save** to save the property changes.
5. Drag and drop an Information activity.

   Set its properties as explained below:

   a. **Name**: Enter a name for the activity, say Leave Approved.
   b. **Delivery Channel**: In the Delivery Channels window, select the **MobileWorkList** check box in the **Channels** section.

      Click **Save Channels**.
   c. **Assign Actor**: Set the actor to whom the information is to delivered. For example, John.
   d. **Subject**: Set the subject for the leave approval information.
   e. **Body**: Set the Body text for the leave approval information.
6. Click Save to save the property changes.
7. Drag and drop another Information activity.

   Set its properties as explained below:

   a. **Name**: Enter a name for the activity, say Leave Rejected
   b. **Delivery Channel**: In the Delivery Channels window, select the **MobileWorkList** check box in the **Channels** section.

      Click **Save Channels**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

c. **Assign Actor**: Set the actor to whom the information is to delivered. For example, John.
d. **Subject**: Set the subject for the leave rejection information.
e. **Body**: Set the Body text for the leave rejection information.

8. Click **Save** to save the property changes.
9. Publish and Test Run the workflow.
10. Log on to the AVEVA Work Tasks Pro mobile application as admin user. For details about configuration settings, see Install and Configure AVEVA Work Tasks Pro App.

    The user will be able to see the assigned task in the Inbox.
11. Tap on the task to view and act on it. Enter comments and click either approve or reject.
12. Log on to the AVEVA Work Tasks Pro mobile application as John. The approval or rejection information has been delivered to John's Inbox.

Note: The Backup and Restore option in iTunes will not restore the configuration settings of Work Tasks Pro. The user needs to re-configure the AVEVA Work Tasks Pro.

Configure AVEVA Work Tasks Pro

You can configure the MobileWorkList option in the Delivery Channel property of the human activities. You can also block or unblock the mobile application installed in the mobile devices.

- Delivery Channel - MobileWorkList
- Block or Unblock Mobile Application

Delivery Channel - MobileWorkList

The Delivery Channel property in Human Activities has to be set to send the tasks to the recipient through different modes of delivery like mail, web, and mobile. In order to send the tasks to the mobile device, the MobileWorkList option in the Delivery Channel property needs to be selected. This allows mobile users to view and act upon the tasks from the Inbox that is rendered on their mobile devices. This option is available in three of the Human activities - Approval, Information, and Invoke Form.

You can configure the MobileWorkList option in the Delivery Channel property of the human activities.

To configure 'MobileWorkList' option in the Delivery Channels property:

1. In the Process Designer, drag and drop the Approval/Information/Invoke Form activity on to the design area. Right-click the activity and the properties pane appears on the right side of the design area.
2. In the **Activity Properties** pane, expand the **Configuration** node and select the **Delivery Channels** property.
3. In the **Delivery Channels** window, select the **MobileWorkList** check box in the Channels section.
4. Click **Save Channels**.

   This enables the activity to deliver the task to the Inbox in the mobile devices.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Block or Unblock Mobile Application

You can block or unblock the mobile application installed in the mobile devices. This feature can be used when a user loses the mobile device and would want to block the access to the application installed in the lost device. Through this feature you can block unauthorized use of the application. After the device is retrieved, the same feature can be used to unblock the application.

The blocking or unblocking feature takes approximately 3 minutes to reflect the changes on the device. Upon receiving a request from a mobile user, the administrator should inform the mobile user about the time taken for the changes to be reflected on the device.

To Block/Unblock Mobile Application on Devices:

1. In the **Central Configuration** page, click the **Block/Unblock Mobile Application on Devices** link to navigate to the page.
2. Select the required option from Search Criteria list. Available options are **User**, **Blocked Devices**, **Unblocked Devices**, and **All**.
   - To view the connection based on the User:
     a. Select the repository to which the user belongs to, from the Repository Name list.
     b. Select the required user from the User Lookup.
   - If you select the **Blocked Devices** or **Unblocked Devices** option, the blocked or unblocked devices are listed.
   - If you select the All option, then all the devices are listed.
3. Click **Go**. This would list all the devices based on the selected search criteria. The list displays the Device ID/ UUID, Device Name, Device Type, OS Version of the application, if the device is blocked, Push Notification Token ID, Audit History of the block or unblock requests, and Connection details of the application.
4. Select the device in which you want to block or unblock the application.
5. To block or unblock all the devices listed here, select the Device ID/UUID.
6. Click **Block** or **Unblock** as per your requirement. The Block Device or Unblock Device appears.
7. Enter the comments for blocking or unblocking respectively. Specify the reason given by the user of the application to block or unblock the application and also the mode of communication used by the user to make this request.

To View Audit History of the Application

1. Follow steps 1-5 from the section To Block/Unblock Mobile Application on Devices. The list of blocked/ unblocked devices are listed.
2. Click the View link in the Audit History column.

   The Audit History appears displaying the following details:
   - Details of the administrator who has acted upon this request
   - Date of the execution of the request
   - Comments added by the administrator while executing the request

To View Device Connections

The View Device Connections allows you to view a list of all the repositories that were configured to a particular

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

device. In this screen you can also clear the active session tokens.

1. Follow steps 1-5 from the section To Block/Unblock Mobile Application on Devices. The list of blocked/ unblocked devices are listed.
2. From the list, click the View link in the Connections column.

Notifications will not be sent to blocked device, even if it has an active session token.

The View Device Connections screen appears displaying the following details:

- **Repository Name**: value specified in the Settings Configuration of the application
- **Provider Name**: value specified in the Settings Configuration of the application
- **User Name**: value specified in the Settings Configuration of the application
- **Last Sync Date**: last date and time when the user logged on the application
- **Connection Name**: displays the Connection Name given during the AVEVA Work Tasks Pro configuration.
- **Session Token**: is the unique identifier which is used to identify the connection between the repository and the device. When the session token's value is zero, it indicates that the specific repository is not connected with device.
- **Clear Session Token**: It is used to clear the session token. Clearing the session toke stops sending notifications to the device. This releases the client connection license acquired for the user.

Advanced Settings

To configure advanced settings:

1. Run the **Farm Configuration Wizard**, and click **Next**.
2. In the Edit **AVEVA Work Tasks** Farm Settings window, select **Service Setting**.
3. Click **Next**.

   The Services/Global Settings window appears.
4. Select **Custom settings**.
5. Click against a service to configure the service. For information about configuring a service, see Service Settings.
6. Click against a Website service for configuring Website settings. For information about configuring a Website service, see Change Web Site Settings.
7. Click against Global Settings. For information about configuring global settings, see Global Settings.
8. Click **Finish**.

Service Monitor

You can use the Service Monitor feature to start, stop, or restart the AVEVA Work Tasks Services. When you launch Windows or start the Farm Configuration Wizard, the Service Monitor feature is automatically started. However, you can also manually start the Service Monitor feature.

For more information about the services, see Service Settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To start or stop a service using the Service Monitor window

1. In the **Microsoft Taskbar**, double-click the icon.
   The **AVEVA Work Tasks Service Monitor** dialog box appears.
2. Perform one of the following:
   - To start a service, click the ▶ icon next to that service.
     The selected service is started and the status of the service is displayed in Blue color.
   - To stop a service, click the ■ icon next to that service.
     The selected service is stopped and the status of the service is displayed in Red color.
     The following notifications are displayed in the Microsoft Taskbar:
     - **Some services stopped working**: When a few AVEVA Work Tasks services are stopped.
     - **All services stopped working**: When all the AVEVA Work Tasks services are stopped.
   - To restart a service, click the ↻ icon next to that service.
     The selected service is restarted and the status of the service is displayed in Orange color.

   Note: The **Start**, **Stop**, and **Restart** icons will be enabled or disabled for a service, based on the status of that service.

To start or stop a service using the context menu

1. In the **Microsoft Taskbar**, right-click the icon.
   The **AVEVA Work Tasks Service Monitor** context menu appears.
2. Select the service and then perform one of the following:
   - To start a service, click ▶ **Start**.
     For example: To start **AVEVA Work Tasks - Client Service**, select **AVEVA Work Tasks - Client Service** from the context menu, and then click ▶ **Start**.
     The selected service is started and the status of the service is displayed in Blue color.
   - To stop a service, click ■ **Stop**.
     The selected service is stopped and the status of the service is displayed in Red color.
     The following notifications are displayed in the Microsoft Taskbar:
     - **Some services stopped working**: When a few AVEVA Work Tasks services are stopped.
     - **All services stopped working**: When all the AVEVA Work Tasks services are stopped.
   - To restart a service, click ↻ **Restart**.
     The selected service is restarted and the status of the service is displayed in Orange color.

   Note: The **Start**, **Stop**, and **Restart** icons will be enabled or disabled for a service, based on the status of that service.

To configure settings for Service Monitor application

1. Browse to the *<AVEVA Work Tasks Install Path>\Bin* path, where the **AVEVAWorkTasksQuickLaunch.exe.config** file is available.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. In the **AVEVAWorkTasksQuickLaunch.exe.config** file, configure the values as follows.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<startup>
<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.2" />
</startup>
<appSettings>
<add key="DisableApplication" value ="false"/>
<add key="RefreshApplicationTimerIntervalInMillisecond" value ="30000"/>
<add key="NotificationTimerIntervalInMillisecond" value="120000" />
<add key="AutoRefreshServiceDetailsScreen" value="true" />
</appSettings>
</configuration>
```

3. For the **DisableApplication** parameter, enter either of the following values:
   - **true**: If you want to disable the **Service Monitor** application.
   - **false**: If you do not want to disable the **Service Monitor** application.
4. For the **NotificationTimerIntervalInMillisecond** parameter, enter the time interval, in milliseconds, after which the notification must be displayed.

   For example: If you want the notification to be displayed after 2 minutes, enter the value as 120000.

   Note: The minimum value for this field is 10000 milliseconds (10 seconds) and the default value is 2 minutes (120000).
5. For the **RefreshApplicationTimerIntervalInMillisecond** parameter, enter the time interval, in milliseconds, after which the **AVEVA Work Tasks Service Monitor** application should be refreshed.

   For example: If you want the **Service Monitor** application to be refreshed after 30 seconds, enter the value as 30000.

   Note: The minimum value for this field is 5000 milliseconds (5 seconds) and the default value is 30 seconds (30000).
6. For the **AutoRefreshServiceDetailsScreen** parameter, enter either of the following values:
   - **true**: If you want the **Service Monitor** application screen to be automatically refreshed, at the interval specified in the **RefreshApplicationTimerIntervalInMillisecond** parameter.
   - **false**: If you do not want the **Service Monitor** application screen to be automatically refreshed, at the interval specified in the **RefreshApplicationTimerIntervalInMillisecond** parameter.
7. Save the **AVEVAWorkTasksQuickLaunch.exe.config** file.

   The settings for the **Service Monitor** application are configured successfully.

Service Settings

You can configure the following services for AVEVA Work Tasks:

- **Workflow Engine**: This service is required and must be running for processing AVEVA Work Tasks workflow requests.
- **Task Scheduler**: This service handles the execution of all tasks that have not taken place as scheduled (escalations).
- **Communication Service**: This service handles incoming e-mail and SMS communication.
- **Mobile Notification Service**: This service handles notifications sent to the AVEVA Work Tasks Pro mobile

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

application.

- **Advance Server service**: This service provides Load Balancing and Failover. This service will be enabled only if the Load Balanced Server license is acquired.

The Central Configuration Settings and the Enterprise Console Settings are available only after a Farm is created.

Note: For any of the services, if you update the port number, then manually restart the IIS server using the iisreset command.

Engine Settings

You can configure the **Workflow Engine settings** while installing the product or after installation at any point in time.

The following table provides information on the various fields and the default values for specific fields:

Field Name	Default Value	Description
Server Port	8855	This field denotes the port number used by the Workflow Engine Service. The Workflow Engine communicates with other services using this port.
Load Balanced	False	This field is used to set load balancing for AVEVA Work Tasks. This field is enabled only for the Load Balanced Server installation. To enable load balancing set the Load Balanced value to true.
Load Balanced Server Port	-	This field allows you to enter the port number for the Load Balanced Server. This field is disabled if Load Balanced value is set to False.
Load Balanced Client Port	-	This field is used to enter the port number for the Load Balanced Client. This field is disabled if Load Balanced value is set to False.

Steps to apply Engine Settings

- Click **OK** to apply these settings.

See also: Auto Resume Settings

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task Scheduler Settings

Task Scheduler Services

This service handles the execution of all tasks that have not taken place as scheduled (escalations).

Task Scheduler does not persist any information in memory. All tasks are database persisted and are performed only when due time is reached.

The Task Scheduler use a queue to process the tasks in parallel. Due to queuing up of Task, you may not get accurate timeouts with seconds accuracy.

Example of the service behavior

Consider that there are 25 threads running for processing the tasks. The polling on Database occurs at 11:00 AM which fetches 30 tasks which are due to perform by 11:45. Since all of them has to be performed at 11:45, all the 25 threads go to sleep stage for 45 Seconds for the first 25 tasks and the rest of the 5 tasks get queued up as there are no threads to process the tasks. At 11:45, tasks get performed freeing up the threads. The 5 tasks which were queued get processed now. As you can see the accuracy of 11:45 is already.

Compromised for the 5 tasks as they might perform at 11:46 or 11:47 or 11:48. To set the Task Pool limit and Thread Pool size, see *Task Scheduler Settings* section later in this topic.

Load Test Behavior of Task Scheduler

The Task Scheduler is tested along with the load test that is normally done for the engine and cannot be tested separately. The test is regarding how much load the Task Scheduler can handle. The test for the engine varies from 1000 to 20,000 instances or more. During this load the task scheduler will be loaded with more than 10,000 tasks which might perform at the same time or at a different time. If all the tasks are due for a specific time there will be a delay for the task perform time.

If there are 10,000 tasks to be executed at the same time, definitely the memory usage will be high as all the tasks will be queued and will get executed in parallel.

For example, assume that the due time is December 11th, 12:30 P.M., on heavy load, if there are many tasks which have to be executed at the same time, some of the tasks will get performed by 12:35 or 12:40. This delay time is mainly due to the threads which are available in the Task Scheduler.

Task Scheduler uses threads for executing the tasks, so on heavy load, threads get queued up and wait for one task to be performed and then the next one. By default parallel there could be 25 tasks which will get executed at the same time, still on heavy load there will be a delay.

Task Scheduler pools the database every one minute to check the tasks which are due. For example, at December 12th 10:18 AM all tasks which are new and due by 10:18 AM will be picked and performed.

Task Scheduler Settings

The Task Scheduler Settings screen is used to customize settings for the various AVEVA Work Tasks tasks.

The following table provides information on the various fields and the default values for specific fields:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Default Value	Description
Server Port	8856	This field denotes the port number used by Task Scheduler service. This value must be modified if the default port value is in use.
Load Balanced	False	This field is used to set load balancing for AVEVA Work Tasks. This field is enabled only for the Load Balanced Server installation. To enable load balancing set the Load Balanced value to true.
Load Balanced Server Port	8853	This field allows you to enter the port number for the Load Balanced Server. This field is disabled if Load Balanced is set to False.
Load Balanced Client Port	8860	This field is used to enter the port number for the Load Balanced Client.
Task Fetch Interval	60	This field is used to specify the interval in seconds at which tasks have to be retrieved.
File Watcher		This field is used to define a folder for File Watcher operation.
Thread Pool Size	25	This field allows you to define a thread pool size for the Workflow Engine. A thread pool size determines the number of threads reserved by a server for providing service requests. For a Workflow Engine deployed in a heavy load scenario, a higher thread pool size needs to be set.

Steps to apply Task Scheduler settings

- Click **OK** to apply these settings.

Communication Service Settings

AVEVA Work Tasks Communication Service makes use of communication service settings for the following activities:

- **Wait for E-mail**: The workflow processes can be designed to be initiated or processed on receipt of email

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

matching specified parameters. The Communication service monitors the specified port and IP for incoming emails that could be used in the Workflows.

- **Wait for SMS**: The communication service can be setup to monitor incoming SMS.
- **Sending SMS**: The workflows can send SMS through a Global System for Mobile communication (GSM) device. The Communication Service uses settings and configuration specified in this screen for sending SMS.

The various communication settings are:

- Communication Server
- Mail Server (The Mail Server setting feature is moved to Central Configuration. For more information, see Mail Server Setting.)
- SMS Receiver
- SMS Config
- SMS Config Custom
- SMS Send Action

Communication Server

The Communication Server settings provides the IP address and the port to be monitored by AVEVA Work Tasks's Communication Engine for incoming e-mails.

The following table provides description of the fields with their default values:

Communication Service Server

Field Name	Default Value	Description
Listener IP Address	-	This field is used to enter the IP address to be monitored for incoming emails. This field defaults to the name of the logged in machine.
Port	8770	This field is used to enter the port number for the Listener IP Address.
Wait Task Handler Path	-	This field is used to provide the assembly navigation path for handling timeouts and e-mail received.
Wait Task Handler Class	-	This field is used to enter the Class name for handling timeout tasks and incoming emails.
Block Mail Addresses	-	This field is used to enter e-mail addresses that should be blocked

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Default Value	Description
		from processing.

Enterprise Server Settings

Field Name	Default Value	Description
Load Balanced	False	This field is used to set load balancing for AVEVA Work Tasks. This field is enabled only for the Load Balanced Server installation. To enable load balancing set the Load Balanced value to true.
Load Balanced Server Port	8853	This field denotes the port number used by Advance Server. This value must be modified if the default port value is in use.
Load Balanced Client Port	8864	This field is used to enter the port number for the Client.

Mail Server

For configuring Mail Server, see Mail Server Setting.

SMS Receiver

The WaitForSMS Action works with the help of the Communication Engine Service. The Communication Engine checks the Phone device periodically for SMS.

Field Name	Default Value	Description
Enable SMS Receiver	False	This field enables the Communication Engine to check for SMS.
SMS Time Interval	0.00:01:00	This field is used to specify the time interval at which the AVEVA Work Tasks Engine must check for SMS in the phone device. Example: 0.00:01:30 = 1 minute 30 secs
Wait Task Handler Path	-	This field is used to provide the Assembly path for handling timeout actions and warnings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Default Value	Description
Wait Task Handler Class	-	This field is used to provide the class name for handling timeout actions and warnings.
Provider Configuration	-	This field is used to enter the details specific to your service provider. AVEVA Work Tasks currently supports only GSM protocols for receiving SMS. • **Name**: Enter the name of the service provider. • **Assembly**: Enter the path • **Class**: Enter the class name for receiving SMS • **Device**: Enter device details

SMS Config

The SMS Send Action settings are used by the Notification Service to send SMS.

ProviderConfig: This is for Connection Type specific configuration settings. If you are using a provider other than the default provider specified here, you will have to add the details here. Currently five Connection

Types are supported:

- SMSPatternMatch
- GSM
- SMSProvider
- Hayes
- EmailProvider.

Under each Connection Type, apart from their configuration details, two compulsory nodes, SMSMessengerAssembly and SMSMessengerClass, ave to be specified.

This is the class which sends SMS for each provider. This class should implement ISMSSender interface.

The following are some examples for setting ProviderConfig:

```
<GSM>
<SMSMessengerAssembly>C:\ Workflow.NET.SMS.Messenger.dll</SMSMessengerAssembly>
<SMSMessengerClass>Workflow.NET.SMS.Messenger.SMSSender</SMSMessengerClass>
<Device> Nokia 6310i Modem </Device>
</GSM>
```

Note: As an alternative, "Nokia 6310i Modem" can be replaced by "COM 1", provided the Nokia phone is connected on COM1.

```
<SMSProvider>
<SMSMessengerAssembly>C:\ Workflow.NET.SMS.Messenger.dll</SMSMessengerAssembly>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<SMSMessengerClass>Workflow.NET.SMS.Messenger.SMSSender</SMSMessengerClass>
<Device> Standard 19200 bps Modem </Device>
<DeviceSpeed> 1200 </DeviceSpeed>
<DeviceSettings> "8, n, 1" </DeviceSettings>
<ProviderDialString> "02012345678" </ProviderDialString>
<ProviderType>1</ProviderType>
</SMSProvider>
```

Here are some details about values in the above example:

- DeviceSettings - 8 databits, no parity, 1 stopbit
- Provider type - Standard TAP provider
- You can specify 0 for UCP Provider, 2?for TAP provider, no request for LineFeed, and 3 for TAP provider, no request for EOT

```
<Hayes>
<SMSMessengerAssembly>C:\Workflow.NET.SMS.Messenger.dll</SMSMessengerAssembly>
<SMSMessengerClass>Workflow.NET.SMS.Messenger.SMSSender</SMSMessengerClass>
<Device> Standard 19200 bps Modem </Device>
</Hayes>
```

Field Name	Description
Connection Sequences	The Connection Sequences is used to provide the sequence in which AVEVA Work Tasks Engine must choose the connection type for sending SMS. **Connection Type**: This column is used to list the various connection types such as GSM, SMSPatternMatch, EmailProvider, SMSServiceProvider, and Hayes. **Sequence Number**: This column is used to list the sequence of each connection type.
Email	This field is mapped to the connection sequence table for sending SMS. The fields are mapped in the user provider for sending SMS. The three fields - Email, Mobile Phone Provider, and SMS Number are required to be mapped to the table. Some providers allow SMS to be sent through email. Enter your email address in the email field to send SMS to these providers.
Mobile Phone Provider	This field is mapped to the table for sending SMS. Enter the name of the Mobile Phone Provider in this field. Example:Hutch
SMS Number	This field is mapped to the table for sending SMS. Enter the phone number to which the SMS has to be sent.

The above settings can be directly used for AVEVA Work Tasks list provider to send SMS. No manual settings are

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

required. However, if different providers are used, the appropriate values need to be set.

SMS Config Custom

To use a connection type other than the default supported, provide SMS Pattern Match information for SMS Messenger Assembly and Class.

Group Name	Field Name	Description
SMS Pattern Match	SMS Messenger Assembly	This field is used to define an assembly which handles sending and receiving SMS.
	SMS Messenger Class	This field is used to enter the class name which handles sending and receiving SMS.
SMS Patterns	-	• **SMS Provider**: Provider name such as, Hutch or Airtel. • **Connection Type**: Connection type like EmailProvider, GSM, SMSServiceProvider or Hayes. • **SMS Pattern**: Pattern like 9845*, 98*.

SMS Send Action

SMS Config Custom Settings enable use of a custom SMS provider.

Field Name	Description
Custom SMS Provider	• **Name**: Enter the name of the SMS Provider. • **Assembly**: Enter the path to the assembly. • **Class**: Enter the class information. • **Device**: Enter the device for the custom SMS provider.
SMS Messenger Assembly	This field is used to define an assembly which handles sending and receiving SMS.
SMS Messenger Class	This field is used to define a Class which handles sending and receiving SMS.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description
Email Server	Name of the e-mail server.
Email Pattern	• **Provider Email Pattern Name**: Name of the e-mail provider. • **Email Pattern**:Pattern used by e-mail provider.

Client Service Settings

Service Monitoring

You can monitor the AVEVA Work Tasks Windows services such as, the Client Monitor, Advance Server, Task Scheduler (TS), Workflow Engine (WF Engine), and Communication Service (CS).

AVEVA Work Tasks architecture uses a plug-in model wherein, any Custom activity, COM-object, or SOA activity can be plugged in as part of the workflow. Sometimes, these activities show a high memory usage. Due to the increased memory usage, the WF Engine service at times gives “OutOfMemoryException” and cannot continue processing the workflow tasks which are in the queue. This in turn delays the execution of the workflows which are in the queue. In addition, the administrator might not know about this until they are alerted by the users about the delays in execution of work flows. To resolve this problem, the administrator will have to restart the Workflow Engine service.

The Service monitoring feature automates this restart process to avoid workflows pending in queues and the administrator's interference in the restart process.

Terminologies

Terms	Description
WF Engine	Workflow engine
TS	Task Scheduler service
Service/Windows Service	AVEVA Work Tasks Windows service
Process	Instance of AVEVA Work Tasks Windows service
Service monitor	AVEVA Work Tasks Client service
Admin	Administrator
CS	Communication service
AS	Advance Server service

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Types of Restarts

There are three types of restarting services:

Restart during Critical Memory Limit:

Whenever the memory limit of a process (instance of Windows service) reaches a Critical Value, the service is immediately restarted. During the Critical memory value of the process, there are high chances of the process throwing *OutOfMemoryException*. By restarting the process, we the *OutOfMemoryException* exception, and thereby, the workflows pending in the queue are avoided.

For more information about Critical Memory Limit, see Process Memory Calculation.

Restart during Warning Memory Limit:

When the memory limit of a process reaches a critical limit, the service is restarted immediately. Even when the service is in high load usage, the restart is unavoidable. To avoid restarting during high load usage, we need to restrict the process from reaching the critical memory limit.

The Client service will monitor this process. Whenever the service finds that the process has reached the warning memory limit for a particular duration of time, it will make a note of it.

If the warning memory limit is exceeding at a low usage time (night times), the service is restarted immediately at that time. The low usage time is configurable in the Configuration Settings.

This attribute is not enabled by default. The Administrator will have to enable in the Configuration Settings.

Restart during Maintenance Phase:

Load of the Workflow tasks will be maximum during a particular phase (duration) in a business cycle. For example, payroll clearance will happen at the end of every month.

In such situations, it is better to restart the service in advance to clean all unknown objects which may still be consuming the memory, so that there is sufficient memory to load the workflows for execution. Restart can happen at a low usage time such as late nights. The low usage time is configurable in the Configuration Settings.

This attribute is not enabled by default. The Administrator will have to enable in the Configuration Settings.

Features of Service Monitoring

Service monitoring has the following additional features. Some of these features are explained in detail in the later sections:

Note: Ensure that the Client service account has the permission to start and stop remaining services.

- Service monitoring can be disabled or enabled depending on the configuration. If the configuration file is not present, then the monitoring is disabled. If the configuration file has some invalid syntax and if there are any errors in the configuration file, then too the service monitoring will be disabled.
- The Client service has the ability to disable or enable monitoring types. In other words, the maintenance monitoring can be disabled and the remaining monitoring can be enabled. By default, the **Critical Memory Monitoring** option is enabled and the **Warning Memory** and **Maintenance Monitoring** options are disabled.
- The Client service has the ability to enable and disable monitoring feature for a specific AVEVA Work Tasks

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

service. For example, Monitoring only the Communication service can be enabled or disabled.

- Whenever the Client service is started, all AVEVA Work Tasks Windows services are also started. Similarly, whenever the Client service is stopped, all AVEVA Work Tasks Windows services are also stopped.
- If any AVEVA Work Tasks Windows service is stopped, it is restarted by the Client service.
- The Client service can also monitor service specific conditions, such as the Workflow engine thread hanging behavior. This is always enabled. To know more about Workflow Engine thread hanging behavior, see Workflow Engine.
- The Client service automatically restarts during its failure. Suppose a memory leakage in the client service stops the client service, then the client service is restarted after 5 minutes. However, it will not be restarted if it is stopped manually or from other applications.
- By default, the Client service sends alert emails when the services are restarted. However, you can configure whether the alert emails must be sent or not.
- If a service is not installed, it will not be monitored even if its configuration section values are present in the configuration file.
- If the **Workflow Engine** service is unable to acquire the required license, then the service does not get started. In this scenario, the Client service monitoring is disabled.
- If the load balancing feature is not enabled on a server, then the **Advance Server Service** will not be monitored by the Client service.
- In a load balanced environment, if the **Workflow Engine** service or the **Advance Server Service** is unable to acquire the required license, then the service does not get started. In this scenario, the client service monitoring is disabled.

Process Memory Calculation

Restart of a Windows service is based on the memory of its processes (instances of Windows service). The Client service will calculate the memory of the process.

We use System.Diagonstic.Process to calculate the memory associated with a specific process.

System.Diagonstic.Process has the following properties to get the memory of a process:

- **WorkingSet64**: This is a set of memory pages currently visible to the process in physical RAM memory. The working set includes both shared and private data. Shared memory is the read-only memory which is used by the multiple processes. Private memory is the read-write memory used by a specific process.
- **PrivateMemorySize64**: Private memory held by the process. It is the read-writable memory which cannot be shared with other processes.
- **VirtualMemorySize64**: Total virtual memory allocated by the OS to a process. The value will be more than WorkingSet64 + PrivateMemorySize64.
- **PagedMemorySize64**: Amount of paged memory allocated for the associated process.

The Client service can be configured to use any one of the or a combination of the above properties. Combination can be in terms of addition (+), subtraction (-), multiplication(*), and/or division(/).

Syntax for the configuration is as given below.

```
<MemoryCalcRule Rule="PrivateMemorySize64"/>
<MemoryCalcRule Rule="WorkingSet64 + PrivateMemorySize64"/>
```

Default is configured as

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<MemoryCalcRule Rule="PrivateMemorySize64"/>
```

- For a 64-bit machine, 1500MB is the default configured value. This value can be changed in Configuration Settings.

Types of Monitoring

The different types of monitoring available are:

- Maintenance
- Critical Memory Limit
- Warning Memory Limit

Maintenance

AVEVA Work Tasks services are monitored for the maintenance. As part of the maintenance, these services are restarted. Services can be restarted on a recurring bases. Recurring frequency can be weekly or monthly. Weekly can be every Saturday or Sunday or a combination of them. Monthly can be every first day of the month or second day of the month or combination of them.

Syntax for the configuration is as given below:

```
<MaintainanceRestart RestartAt="01:00" WeeklyRestartDay="Sat,Sun"
MonthlyRestartMonths="Jan,Feb" MonthlyRestartDate="1" />
```

- **RestartAt** is the time at which the services need to be restarted. Specify the value in the 24 hours format. E.g.: 23:00. This is a mandatory attribute.
- **WeeklyRestartDay** is the week day(s) on which the services need to be restarted.

  Specify the value as first 3 letters of the day, separated by comma. E.g.: Sun,Mon,Tue,Wed,Thu,Fri,Sat

  Specify the value as “All” if the services need to be restarted on all days. E.g.: All
- **MonthlyRestartMonths** is the month on which the services need to be restarted.

  Specify the value for “MonthlyRestartMonths” as the first 3 letters of a month. E.g.: Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec

  Specify “All” if you want to monitor the services for all the months. E.g.: All
- **MonthlyRestartDate** is the date on which the services need to be restarted. Ensure that you configure both the attributes together.

  Specify the value for “MonthlyRestartDates” attribute as comma separated dates. E.g.: 29,30,1,2,3 .

  If the values for both weekly and monthly attributes are mentioned, then the services will be restarted on both the days.

Either **Weekly Restart** configuration, **Monthly Restart** configuration, or a combination of both can be present.

In the example given below, restart happens on every Saturday and Sunday at 1AM.

```
<MaintainanceRestart RestartAt="01:00" WeeklyRestartDay="Sat,Sun"
MonthlyRestartMonths="" MonthlyRestartDate="" />
```

In the example given below, restart happens on the first of every month at 1AM. If a date mentioned in the configuration file is not present (for example February does not have 30 days), then that value is skipped.

```
<MaintainanceRestart RestartAt="01:00" WeeklyRestartDay="" MonthlyRestartMonths="All"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
   MonthlyRestartDate="1" />
```

In the example given below, restart happens every Saturday, Sunday, and on the first of every month.

```
   <MaintainanceRestart RestartAt="01:00" WeeklyRestartDay="Sat,Sun"
   MonthlyRestartMonths="All" MonthlyRestartDate="1" />
```

To disable MaintainanceRestart, this node has to be deleted or commented.

Critical Memory Limit

Critical Memory Limit is the peak memory size, in MB, of a process. At this size, the process starts throwing OutOfMemoryException, and hence cannot perform any task.

During such a condition, the service needs to be restarted.

After the configured critical memory limit is reached, the service is restarted immediately. It will be configured as mentioned below.

```
   <CriticalMemory MemoryLimitInMB="700"/>
```

For a 64-Bit machine, 1500MB is the default configured value.

To disable Critical memory limit, the above node needs to be commented or deleted.

Warning Memory Limit

Warning Memory Limit of a service is the memory size of the process. At this size, the process can be on the verge of failure (OutOfMemoryException). If the memory size increases to a higher extent, the process can enter the phase of Critical Memory Limit. To avoid this, the process can be restarted when its load is less, probably during night time.

Process should be in warning memory phase for a specified duration (WarningTimeLimitInMinutes). Only then the service will be restarted. In addition, the warning memory limit is reached at the specified RestartAt time.

Configuration:

```
   <WarningMemory MemoryLimitInMB="500" TimeLimitInMinutes="10"
   RestartAt="01:00"/>
```

The above configuration means that if the process has reached the memory of 500MB at 1AM and it was 500MB for more than 10 minutes, only then the service is restarted.

All the three attributes are mandatory fields.

To disable Warning Memory monitoring, the above node should be deleted or commented.

Service Specific Monitoring

Service specific monitoring is used for monitoring the thread pool and checking the engine hanging behavior.

Workflow Engine

AVEVA Work Tasks services can have their own monitoring logic. The monitoring logic is executed by the Client Service.

Monitoring the thread pool and checking the engine hanging behavior:

At times, the Workflow Engine is unable to execute the Workflows though sufficient memory is available. In addition, it may not even throw up a MemoryOutOfException. The reason could be that some workflow threads

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

may be hanging and may not be released from the thread queue. Eventually, the pending workflows do not get the thread queue and hence do not get processed.

The Workflow Engine has a default thread pool configuration of 30 threads. However, repositories can have their own configuration in the Farm's SKConcurrentThreadPools.

For more information, see *Thread Pools at Repository Level* in the *Build Business Process* section.

For example, if there are three repositories where "Repository A" has a thread configuration of 20 threads in SKConcurrentThreadPools and "Repository B" and " Repository C" have no configurations in SKConcurrentThreadPools.

Then the total configured threads will be 20 + 30 = 50 threads.

The Client service will check the **InternalStatus** and **LastUpdatedDateTime** columns of the SWExecute table. After a new workflow execution starts, its status will be “ex” (executing) and **LastUpdatedDateTime** will be the current datetime. If a workflow status is "ex" for a duration of about 10 minutes (Configuration Settings in WFEngineThreadScanInterval node), then the engine is hanging as the workflow is not yet processed completely.

Out of 50 threads, if 30 threads are hanging, that is if their status is not changed from executing to complete, then there are chances that a lot of workflows are pending in queue. Hence, the Workflow Engine is hanging. The Client service has a threshold value of about 70% (Configuration Settings in the WFEngineThreadThresholdValue node). After the number of hanging threads reaches the threshold value, the service is restarted and hence all the threads are freed.

After the Client service is started followed by a change in the value of SKConcurrentThreadPools, then the Client service needs to be restarted to monitor with the updated values. Values from SKConcurrentThreadPools are cached in the Client service. This cache can be broken and recreated after the Client service is restarted.

Service Dependencies

The Client Service, Advanced Service, Workflow Engine, Task Scheduler, and Communication Service are inter-dependent. These dependencies will be considered for monitoring.

Dependencies

- If the Client service is stopped on a machine, all other services in that machine are also stopped. Similarly, if the Client service is started on a machine, all other services in that machine are also started.
- If a service crashes or is stopped manually, then its dependent services also must be stopped and restarted in a sequential manner.
- **Order of starting ad/or stopping the services**: Communication Service, Task Scheduler, and WorkFlow Engine.
- Advance service is present only in the load balanced environment. If the Advance service has to be stopped by the Client service, then stop the services in the following order: Advance Service, Communication Service, Task Scheduler, and Workflow Engine. Follow the same order while starting the services.
- For the remaining services (CS, TS, and WF Engine), if any service is stopped, then all the services are restarted in an order. For example, if Task scheduler needs to be stopped by service monitoring, then Communication Service, Task Scheduler, and WorkFlow Engine are stopped. The service monitoring will start Communication Service, Task Scheduler, and Workflow Engine in an order.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Load Balanced Environment

In a load balanced environment, ensure that the AVEVA Work Tasks Windows services are not restarted at the same time in all machines. Hence, if MaintenanceRestart is configured, ensure that the timing is not the same in all the boxes. If MaintenanceRestart timing is same, then the client service will restart the remaining Services at the same time in all the machines.

In a load balanced environment, the Communication Service is restarted only in one box. During Farm Configuration, the Administrator has to configure the Communication Server too. The configuration value "Listener IP Address" has to be the IP address or computer name of the machine where the Communication Service will be running.

The Client service will monitor the Communication Service only if the Computer name or IP address of the machine matches with the Listener IP Address. Else, the Client Service will not monitor the Communication Service in that box.

The Advanced Service is monitored only in a load balanced environment.

If it is a client machine which is installed with only the web server, then that machine will not be monitored.

Configuration Settings

The entire Service monitoring configuration is mentioned in the file **ServiceMonitoring.xml**. See the comment sections of **ServiceMonitoring.xml** to know the specification of each attribute.

By default, the **AppConstants** section is commented.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--“” The services are not monitored if the config file is not present, or if the value
of the “ServiceMonitoringEnabled” attribute is set to “false”, or if the config file
throws an invalid format(syntax) error. Alert emails are sent if the value of the
“SendAlertEmailsEnabled” attribute is set to “true", or if the “SendAlertEmailsEnabled”
attribute is not present or has an invalid value.-->
-<SMConfig SendAlertEmailsEnabled="true" ServiceMonitoringEnabled="true">
<!--DO NOT CHANGE AppConstants. The AppConstants are used for internal processing and
are commented out by default. -->
<!-- <AppConstants> <!- The frequency (in seconds) at which the client service monitors
the memory of the services. By default, the client service will check the other BPM
services every ten seconds.-> <TimerIntToMonitor>10</TimerIntToMonitor> <!-The frequency
(in seconds) at which the client service checks if all the services are installed.->
<TimerIntToCheckServicesInstalled>60</TimerIntToCheckServicesInstalled> <!-Time (in
seconds) to wait for the status of a service to be changed from “stop pending” to
“stopped”.-> <WaitTimeForStopPendingStatus>300</WaitTimeForStopPendingStatus> <!-Time
(in seconds) to wait for the status of a service to be changed from “start pending” to
“running” . ->
<WaitTimeForStartPendingStatus>15</WaitTimeForStartPendingStatus> <!-Time (in seconds)
to wait between the stop and start of services. >
<SleepTimeBeforeStartStop>30</SleepTimeBeforeStartStop> <!-For the Workflow Engine, the
frequency (in minutes) to monitor the thread lock behavior. ->
<WFEngineThreadScanInterval>10</WFEngineThreadScanInterval> <!-The number of threads
that can be in the “locked” state, defined as a percentage value. ->
<WFEngineThreadThresholdValue>70</WFEngineThreadThresholdValue> <!-This flag enables
logging of all the debug messages related to Client Service Monitoring behavior in the
AVEVA Work Tasks logger. The default value is False.->
<EnableDebugLogMessages>false</EnableDebugLogMessages> </AppConstants> -->
<!-- DO NOT CHANGE MemoryCalcRule. The memory calculation rule can be based on
Performance Monitor attributes such as PrivateMemorySize64, WorkingSet64,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
VirtualMemorySize64, and PagedMemorySize64 of the AVEVA Work Tasks processes. The rule
can have any one of or a combination of these attributes. E.g.: PrivateMemorySize64 +
WorkingSet64. “PrivateMemorySize64” can be used to measure process memory. The
"PrivateMemorySize64" attribute is considered by default, for monitoring, if the
MemoryCalcRule attribute is not present. -->
<MemoryCalcRule Rule="PrivateMemorySize64"/>
<!-- "MaintainanceRestart" is the time at which all the services can be restarted.
"RestartAt" is the time at which the services need to be restarted. Specify the value in
the 24 hours format. E.g.: 23:00
"WeeklyRestartDays" is the week day(s) on which the services need to be restarted.
Specify the value as first 3 letters of the day, separated by comma. E.g.:
Sun,Mon,Tue,Wed,Thu,Fri,Sat Specify the value as “All” if the services need to be
restarted on all days. E.g.: All
"MonthlyRestartMonths" is the month on which the services need to be restarted. Specify
the value for “MonthlyRestartMonths” as the first 3 letters of a month. E.g.:
Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec Specify “All” if you want to monitor the
services for all the months. E.g.: All
"MonthlyRestartDates" is the date on which the services need to be restarted. Ensure
that you configure both the attributes together. Specify the value for
“MonthlyRestartDates” attribute as comma separated dates. E.g.: 29,30,1,2,3
If the values for both weekly and monthly attributes are mentioned, then the services
will be restarted on both the days. If no attributes are mentioned or
“MaintainanceRestart” attribute is not present, then the maintenance monitoring does not
happen. Example of Maintenance restart: <MaintainanceRestart RestartAt="00:00"
WeeklyRestartDays="Sat,Sun" MonthlyRestartMonths="All" MonthlyRestartDates="1"/> -->
<!--<MaintainanceRestart RestartAt="" WeeklyRestartDays="" MonthlyRestartMonths=""
MonthlyRestartDates=""/>-->
<!-- If the “DefaultConfig” and “ServiceConfigs” nodes are not present, then the system
will monitor for the CriticalMemory of all AVEVA Work Tasks Services, with default
values. For the Workflow engine, the thread hanging behavior is also monitored. The
default CriticalMemory limit for a 64-Bit machine is 1500MB. -->
-<ServiceConfigs>
<!--For DefaultConfig, if the values for MonitorService, CriticalMemory, and
WarningMemory attributes are not mentioned for a service, then the default values for
these attributes are considered.
E.g.: To change the values for the Workflow Engine service, specify the attribute values
as follows:
<ServiceConfig Name="WFEngine" MonitorService="true"> <CriticalMemory
MemoryLimitInMB="1000"/>
<WarningMemory MemoryLimitInMB="800" TimeRangeInMinutes="10" RestartAt="00:00"/>
</ServiceConfig>
If the value of the “MemoryLimitInMB” attribute is zero, then the services will not be
monitored. If you do not want a service to be monitored, then set the value of the
“MontiorService” attribute for that specific service to “false”. -->
-<DefaultConfig MonitorService="true">
<!--<CriticalMemory MemoryLimitInMB="1000"/>-->
<!--<WarningMemory MemoryLimitInMB="0" TimeRangeInMinutes="10" RestartAt="00:00"/>-->
</DefaultConfig>
<ServiceConfig Name="AdvancedService"/>
<ServiceConfig Name="WFEngine"/>
<ServiceConfig Name="TaskScheduler"/>
<ServiceConfig Name="CommunicationService"/>
<ServiceConfig Name="ArchestrAMobileNotificationService"/>
</ServiceConfigs>
</SMConfig>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Application Constants

In the configuration file, there is a section for the **AppConstants**. Application constants are used by the service monitoring for the internal business logic. These values can only be changed after the careful analysis of service monitoring. By default it is commented.

If the **TimerIntToMonitor** is changed to five seconds from 10 seconds, after the client service is restarted, it will start scanning (monitoring) for the remaining AVEVA Work Tasks services for every five seconds.

Default Settings

Configuration has the default setting section "DefaultConfig". Default setting is for MonitorService property, CriticalMemory and WarningMemory sections. If these settings are not mentioned for a service, then the values are taken from the default settings. If a service has its own settings, then those settings will override the default settings.

If the "**MonitorService**" attribute is not present or its value is empty for a specific service, then the default section value is taken.

If the "**MonitorService**" attribute is not present even in the default section, then the MonitorService="true" will be considered.

If "**MonitorService**"="false" for a specific service, then that service will not be monitored. But, the same service can be restarted as it may be a dependent service of other AVEVA Work Tasks Window service. If service monitoring has restarted a AVEVA Work Tasks Window service, then its dependent services are restarted.

If the CriticalMemory node is not present or if the MemoryLimitInMB attribute is empty for a specific service, then the default setting is considered. If the MemoryLimitInMB value is zero, then the critical memory limit will not be monitored.

CriticalMemory monitoring is enabled by default. The critical memory limit is monitored even if "CriticalMemory" is not present in the default section. The monitoring is done based on the "PrivateMemorySize64" property of a process.

By default, the value for "MemoryLimitInMB" is considered as 1500MB for a 64 bit machine.

Similarly, for the WarningMemory section too, if the node value of "WarningMemory" is not found or if any of its attribute value is empty for a service, then the default configuration settings will be considered for a service.

If the value of MemoryLimitInMB attribute is zero, then the WarningMemory will not be monitored. If the WarningMemory section is not found in the defaultConfig section, then the WarningMemory monitoring will be disabled for all services which depend on the default configuration.

The Advanced Service, WorkFlow Engine, Task Scheduler, and Communication Service sections do not have any settings by default. These sections will take the values from the default settings. If you want to change any specific value, then you can mention the values in the respective sections.

User Scenarios

The Administrator needs to configure some of the setting of the client service, especially in case of a load balanced environment. In a load balanced environment, the client service should not restart the AVEVA Work Tasks services at the same time in all boxes.

The Administrator also needs to configure the scheduling time as it is recommended to schedule the restart at a low usage time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Administrator wants to change the restart schedule time settings for the WarningMemory limit

If the Administrator wants to set night 2 AM as the restart time for the warning memory, then the Administrator can change the value to the desired timing in the WarningMemory configuration section of the default section. This will affect settings of all services as they are using the default settings.

```
<WarningMemory MemoryLimitInMB="800" TimeLimitInMinutes="10" RestartAt="02:00"/>
```

The Client service monitors the AVEVA Work Tasks service. The Client service will start monitoring the Warning memory limit at the configured time, that is 2AM.

At night 2AM, the WF engine shows a memory above 800MB for more than 10 minutes. The Client service will restart the WF engine and its dependent services. The Client service immediately stops the Communication Service, Task Scheduler, and WF engine and then starts these services again.

Administrator wants to change the restart schedule time settings for the Warning Memory limit only for WF Engine service.

The Administrator wants to set night 12 AM as the restart time for the warning memory for WF engine. For the remaining services, the administrator wants the default values for the settings.

In order to do this, the Administrator has to create the WarningMemory configuration section for the WF Engine and change the value to a desired timing. This same scenario holds good for CriticalMemory too.

```
<DefaultConfig MonitorService="true">
      <CriticalMemory MemoryLimitInMB="700"/>
<WarningMemory MemoryLimitInMB="500" TimeRangeInMinutes="10" RestartAt="2:00"/>
    </DefaultConfig>
    <ServiceConfig Name="AdvancedService" />

    <ServiceConfig Name="WFEngine">
       <WarningMemory MemoryLimitInMB="500" TimeRangeInMinutes="10" RestartAt="00:00"/>
    </ServiceConfig>
    <ServiceConfig Name="TaskScheduler" />
    <ServiceConfig Name="CommunicationService" />
```

Administrator wants to restart Services on the night of 25th of every month.

There are a bunch of monthly scheduled workflows to execute for salary processing on the 25th of every month. For faster processing, the process needs more memory.

The Administrator wants to restart the services before executing the workflows, so that more memory is available for the process.

The Administrator can even set the timing for a monthly maintenance.

```
<MaintainanceRestart RestartAt="01:00" WeeklyRestartDays="" MonthlyRestartMonths="All"
MonthlyrestartDates="25"/>
```

On the 25th of every month, when the time is 1AM, the following services are automatically stopped and restarted in the order specified below.

1. Advance service
2. Workflow Engine

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Task Scheduler service
4. Communication services

If SendAlertEmails= "true", then an email is sent to the configured email.

Administrator has configured all services for the load balanced environment. Administrator wants to have monitoring of service in load balanced environment.

In a load balanced environment, monitoring of service will also happen. Client service will monitor all AVEVA Work Tasks services which are in the same box. During Maintenance monitoring, in the load balanced environment, all services might be restarted at the same time if the scheduled timing is same. Ensure that all the services are not stopped at the same time in all the boxes.

This can be avoided by scheduling the Maintenance monitoring at different time in all boxes.

You can provide a gap of 15 minutes in scheduling time in all boxes.

The same applies for the WarningMemoryLimit too.

Administrator wants to disable sending of Alert emails

If the Administrator does not want to receive monitoring alert emails regularly, then the Administrator can disable sending of alert emails by changing the value for SendAlertEmails to False. Alert emails will not be sent.

Administrator wants to change Memory limit for the Critical memory setting

For WF engine, memory limit for the critical memory settings is 700 MB. The Administrator wants to change it to 1000 MB as the server machine is configured for the process virtual address space more than 2 GB.

For more information about the Memory Size of the process, check in the section Process Memory Calculation.

The Administrator can change the value to 1000 MB in the Workflow Engine section.

```
<ServiceConfig Name="WFEngine" ServiceName="AVEVA Work TasksWorkflow2"
               ProcessName="WFEngine.NET2"        MonitorService="true">
      <CriticalMemory MemoryLimitInMB="1000"/>
      <WarningMemory MemoryLimitInMB="500" TimeLimitInMinutes="10"
               RestartAt="01:00"/>
</ServiceConfig>
```

Whenever the Workflow Engine process memory reaches 1000 MB, the Workflow Engine service is stopped. Therefore, its dependent services, Task Scheduler, and Communication Service are also stopped.

Task Scheduler, Workflow Engine and Communication service are started.

Administrator wants to change Memory limit for the Warning memory setting

If the administrator wants to set the memory limit for the warning memory, the Administrator has to access the specific process section like Workflow Engine section and change the values as mentioned below:

```
<WarningMemory MemoryLimitInMB="800" TimeLimitInMinutes="10" RestartAt="02:00"/>
```

The Client service will start monitoring the Warning memory limit at night 2 AM. If the process memory is above 800 MB at that time, it will note it down and continue monitoring for the next 10 minutes (TimeLimitInMinutes

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

configuration). If the process memory is above 800 MB even after 10 minutes, then the Client service will restart the process along with its dependent services.

For the Warning memory limit, Administrator wants to change TimeLimitInMinutes section.

TimeLimitInMinutes is the configuration setting which defines the time duration (time range) for which the process needs to be in warning memory limit phase to restart the service. If the Administrator wants this configuration then the process needs to be running for at least 15 minutes with a warning memory limit of 800 MB at night 2:00 AM, to restart. Then the configuration setting must be as given below:

```
<WarningMemory MemoryLimitInMB="800" TimeLimitInMinutes="15" RestartAt="02:00"/>
```

Administrator wants to restart services on every Saturday and Sunday night as part of maintenance.

The Administrator can provide the following configuration:

```
<MaintainanceRestart RestartAt="01:00" RestartDay="Sat,Sun" MonitorService="true"/>
```

The services are restarted every Sunday night at 1 AM.

Administrator wants to restart the services on every Sunday as well as on the 25th of every month.

The Administrator can provide the combination of monthly as well as daily configuration. If the Administrator wants to start the service on every weekend as well as on the 25th of every month, then the configuration can be provided as follows:

```
<MaintainanceRestart RestartAt="01:00" WeeklyRestartDays="Sun"
MonthlyRestartMonths="All" MonthlyrestartDates="25"/>
```

Administrator wants to disable Maintenance monitoring for all service.

The Administrator can disable Maintenance monitoring of services on specific days where there might not be high load workflow execution.

To disable Maintenance monitoring, the administrator has to comment or remove the "MaintenanceRestart" attribute.

Administrator does not want monitoring of specific service.

If the Administrator does not want to monitor a service, then the Administrator can configure the value of the "MonitorService" attribute of that service to "false".

By setting the value of "MonitorService" to "false", the following features are disabled:

- MaintenanceRestart monitoring of service is stopped.
- CriticalMemory limit monitoring of service is stopped.
- WarningMemory limit monitoring of service is stopped.

If the AVEVA Work Tasks service is stopped, it will not be automatically started by the Client service.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Client service start or stop events will not have any affect on start or stop of the remaining AVEVA Work Tasks service.

Administrator wants to change the memory calculation rule

By default, the memory calculation of the AVEVA Work Tasks process is done based on the Performance Monitor attribute "PrivateMemorySize64".

The "PrivateMemorySize64" attribute provides the value of the total memory held in the process. For more information about memory of process, check the Process Memory Calculation section.

If the Administrator includes the "WorkingSet64", then the value will be:

```
<MemoryCalcRule>WorkingSet64 + PrivateMemorySize64 </MemoryCalcRule>
```

If the Administrator wants to use VirtualMemorySize64 instead of PrivateMemorySize64, then the configuration will be:

```
<MemoryCalcRule>VitualMemorySize64 </MemoryCalcRule>
```

Administrator wants to start all services a with single click.

The AVEVA Work Tasks Windows service can now be started and/or stopped automatically.

If the Client service is started, all its dependent services such as the Advanced service, Communication services, Task scheduler, and Workflow Engine are started in the specified sequence.

Similarly, if it is stopped, all its dependent services such as the Advanced service, Communication services, Task Scheduler and Workflow Engine are stopped in the specified sequence.

Administrator wants to stop all services.

The Administrator may want to stop all the AVEVA Work Tasks Windows services for maintenance or for other reasons. The Client service will have the feature of automatically stopping all the services. Hence, the administrator can stop the client service which will in turn stop all other services such as the Advanced service, Workflow Engine, Task Scheduler and Communication services are stopped in the specified sequence.

Administrator wants Communication service to be stopped.

The Administrator wants to stop Communication service as part of maintenance or for other reasons, but wants the remaining services to continue execution. The administrator does not want the Communication service to be automatically started.

In other words, the Communication service should not be monitored.

For any service to be excluded from monitoring, set the value of the "MonitorService" attribute to "false", under the specific "serviceconfig" section.

Administrator has changed the startup type of client service from automatic to manual.

On machine restart, the Client service will not be automatically started. However, the remaining AVEVA Work Tasks services are automatically started. The Administrator needs to manually start the Client service to monitor the remaining Services.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Technical Scenarios and Features

Due to "OutOfMemoryException" Workflow Engine service has stopped. Administrator wants it to be automatically started.

AVEVA Work Tasks architecture uses plug-in model. Any Custom activity, Com-object, or SOA activity can be plugged in as part of the workflow. Due to high memory usage by these activities, Workflow Engine is throwing OutOfMemoryException. Workflow Engine stops processing the workflows and hence workflows are stocked in piles in the queue. During such a condition, the Administrator has to restart the services. This can be automated by service monitoring.

In the production environment, service monitoring is enabled. One of the features of service monitoring is, restarting the service if it is stopped.

If the Workflow Engine service is stopped, Communication services, and Task Scheduler are also stopped. Then Communication services, Task Scheduler, and Workflow Engine are started in the given order. The SendAlertEmail is enabled and an email is sent.

The Maintenance restart and WarningMemory restart time are configured with an identical time for the Workflow Engine service.

At the specified time, the memory reaches a Warning memory limit. The order of execution of logic is Service specific monitoring logic followed by Maintenance monitoring, CriticalMemory monitoring, and Warning memory monitoring.

For the Workflow Engine, the Service specific monitoring is executed first. Then the Maintenance restart is executed and hence the service is restarted. But, the WarningMemory restart will not happen as the memory has decreased below the specified limit.

Client service had failed due to OutOfMemoryException

Suppose the client service has failed due to *OutOfMemoryException*. This should not stop monitoring the remaining services. It has to be restarted automatically.

Client service installer has the recovery option set, so that it recovers automatically when there are failures.

Workflows are pending in queue for long time. Not getting processed by Workflow Engine even though there is no OutOfMemoryException and memory is available for the process.

Client service monitors the Workflow queues SWExecute. All workflows which need to be executed by Workflow Engine will be stacked in this table.

Client service will check the LastUpdatedDateTime for all “executing” workflow tasks. If the status has not changed for a long time, say 10 minutes, then it means that the Workflow Engine is not able to release the existing threads and hence not able to continue with the new workflow tasks. During such a condition, the client service will restart the Workflow Engine.

If all workflows in thread queue are taking more than 10 minutes, the client service will restart Workflow Engine. If all the worklfows continue to hang, then the client service will restart again. This behavior will continue until the workflows complete execution within 10 minutes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Service monitoring in Development environment

In the development environment, the user will be verifying the workflows, services, adding the custom forms, and actions. The user may not want to restart the services over a high memory usage in development environment.

To avoid monitoring of the services in development environment, the application will have the configuration setting ServiceMonitoringEnabled="false".

Only if the setting ServiceMonitoringEnabled="true", the monitoring feature is enabled. For any other values, it is disabled.

Some of the configuration settings are empty for the Workflow Engine

If the Monitor service attribute is not present or the value is not present, then the default section value is taken.

If CriticalMemory node is not present or MemoryLimitInMB attribute is not present, then the default setting is considered. If the MemoryLimitInMB has an empty value, then the default setting is considered. If the MemoryLimitInMB=0, then the CriticalMemory monitoring will not process.

The same logic is applicable to the WarningMemory section. If the attribute (or node) is not found or its value is empty, then the default settings will be considered. If its MemoryLimitInMB=0, then the WarningMemory monitoring will not process.

Mobile Notification Service

The **Mobile Notification Service** ensures reliable delivery of notification to the registered mobile application. These notifications are sent through the Azure notification hub.

The timely delivery of notification depends on the network speed. Hence, we recommend to have a good internet connectivity.

An entry is made in the *SKNotificationMessage table* when there is connectivity failure and the workflow engine is not able to send a notification. The Mobile Notification Service reads *SKNotificationMessage* table, fetches the notification, and sends it to the registered mobile device.

The Mobile Notification Service Settings window has the following tabs that you can use to subscribe and configure this service:

- Register to Mobile Notification Service
- Retry Settings
- Enterprise Server Settings (This tab appears only in case of Enterprise Server environment.)

Register to Mobile Notification Service

The **Mobile Notification Service Settings** registration enables the system to send notifications to mobile devices.

Use the **Registration Details** section to register for sending notification to mobile devices. You can register *n* number of mobile devices to receive notifications.

Registration can be done anytime, either during Farm configuration at install time, or later. On clicking the **Register** button, the system gets enabled to send notifications. Also, all the available devices in the system get registered in the Azure notification hub.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:
- It is recommended that the system acquires the license with the highest precedence available, before registering for Push Notification.
- In an Enterprise Server environment, if you register for Push Notification in one of the systems in the cluster, then the registration is completed on all the other systems in the cluster.
- The Load Balanced Server and Single Server licenses will point to Production Push Registration service.
- Restart the services when prompted by the Farm Configuration Wizard.

To register to Mobile Notification service

1. Run the **Farm Configuration Wizard**, and click **Next**.
2. In the Edit Farm Settings window, select **Service Settings**.
3. Click **Next**.

   The Services/Global Settings window appears.
4. Click the edit button near the **Mobile Notification Service**.

   The Mobile Notification Service Settings window appears.
5. Click the **Register** tab.
6. Click **Test Connection** to verify the connection.
7. Click **Register**.

   Note: The registration process might take a significant amount of time based on the Internet connectivity and the number of devices available in the system.

   A message about successful registration appears.
8. Click **OK**.
9. To send a test notification, perform the following in the **Test Notification** section.
   a. Select a device from the **Device** drop-down list.
   b. Enter a value in the **Badge Count** field. This number will be sent to the selected device as a test notification.
   c. Click **Send**.

      If a notification is successfully sent to the mobile device, a message about successfully sending notification appears.
10. Click **OK**.

    The system is successfully registered for sending notifications to mobile devices.

    Note: After registering the notification service, if you close the Farm Configuration Wizard and reopen it to send a Notification Badge Test Count, it results in an error. Hence, send the Notification Badge Test Count in the same session.

To regenerate the keys to register to Mobile Notification Service again

You can regenerate the notification hub key if the key is compromised in any circumstances.

Note:
- Ensure that you regenerate this key only if it is compromised and is used for malicious purposes.
- When you click the **Regenerate** button, you will have to wait for a minute for the notification/test notification to work.
- Regenerating the key after registering the notification service results in an error when attempting to send a Notification Badge Test Count.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. In the **Register** tab, click **Regenerate**.

   A confirmation message to regenerate the notification hub key appears.
2. Click **Yes**.

   A message about successful regeneration of the notification hub key appears.

   You must restart the AVEVA Work Tasks services for the changes to take effect.

To unregister from the Mobile Notification Service

To unregister from the Mobile Notification Service, perform the following:

1. In the **Register** tab, click **Unregister**.

   A message appears asking your confirmation to unregister the devices.
2. Click **Yes**.

   A message about successful unregistration appears and all the registered devices in the notification hub gets unregistered.

Retry Settings

Configuration in the **Retry Settings** tab enables you to configure the number of retries and the interval at which the system retries to send notifications and register devices.

The following image shows the **Retry Settings** tab:

Notification Retry Settings

- **Retry Count**: If the notification is not delivered at the first attempt, you can set the number of retry attempts to deliver the notification. After the defined limit is reached, the notification expires. If the **Retry Count** value is set as 0, the failed notifications will not be resent. The default value is 5.
- **Retry Interval**: If the notification is not delivered at the first attempt, you can set the interval between delivery attempts. The system tries to resend the notification at the interval defined in this field. The time entered should be in *days:hr:min:sec* format only. The maximum limit is 10 minutes. The default value is 30 seconds.

Device Registration Retry Settings

- **Retry Count**: A device needs to be registered in the Azure notification hub before it can receive notifications. If the device registration is not successful at the first attempt, you can set the number of retry attempts to register the device. If the **Retry Count** value is set as 0, the failed device registrations do not happen. The default value is 5.
- **Retry Interval**: If the device registration is not done at the first attempt, you can set the interval between device registration attempts. The system tries to re-register the device at the interval defined in this field. The time entered should be in *days:hr:min:sec* format only. The default value is 2 minutes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Enterprise Server Settings

The **Enterprise Server Settings** tab is enabled only in the Enterprise Server mode. Configuration in this tab manages the load balance on the Server and Client.

- **Load Balanced**: Set load balancing for AVEVA Work Tasks. This field is enabled only for the Load Balanced Server installation. To enable load balancing, set the Load Balanced value to *True*.
- **Load Balanced Server Port**: Enter the port number for the Load Balanced Server. This field is disabled if Load Balanced is set to *False*.
- **Load Balanced Client Port**: Enter the port number for the Load Balanced Client.

Migrating Notification Services

Systems that are not configured to use TLS 1.2 would need the following registry entries for the new Notification Hub to work:

- [HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319]

"SystemDefaultTlsVersions"=dword:00000001

"SchUseStrongCrypto"=dword:00000001

- [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]

"SystemDefaultTlsVersions"=dword:00000001

"SchUseStrongCrypto"=dword:00000001

Note:
- Before you run the **AVEVA Work Tasks Notification Migration Wizard**, unregister from the existing Mobile Notification services. For information on unregistering, see Unregistering from the Mobile Notification Service.
- After unregistering the Mobile Notification services, close the **Farm Configuration Wizard** and then run the **AVEVA Work Tasks Notification Migration Wizard**.

To migrate Notification Services

To migrate the mobile notification service from earlier versions of the product, follow the steps given below:

1. Run the **AVEVA Work Tasks Notification Migration Wizard**.
2. Provide the Database Server name, Farm Database name, Authentication Mode, and the Database credentials.
3. Click **Migrate** to migrate the Notification details.
4. Run the **Farm Configuration Wizard**, and click **Next**.
5. In the Edit Farm Settings window, select **Service Settings**.
6. Click **Next**.

   The Services/Global Settings window appears.
7. Click the edit button near the **Mobile Notification Service**.

   The Mobile Notification Service Settings window appears.
8. Click **Test Connection** to verify the connection. Make sure the URL is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

mobilenotification.worktasks.connect.aveva.com, and then click **OK**.

9. Click **Register**.

Enterprise Server Settings

You can configure Enterprise Server Settings using the Multicast mode or the Database mode, while installing the product or after installation at any point in time.

The following table provides information about the various fields and the default values for specific fields:

Field Name	Default Value	Description
Enterprise Server Mode	Multicast	Select the mode as **Multicast** or **Database**. • If you select the mode as **Multicast**, then it uses the configured Cluster Port, Cluster IP, and Multicast Local IP to share its server status with other servers. • If you select the mode as **Database**, the server status is stored in the database and is shared with other servers.
TCP Port	8853	Communication between the Workflow Engine and the Advance Server service is established through this port. This port is used to get the status of the least used server, and also analyze server performance.
Cluster Port	8850	Use this port to check the connectivity between all the machines while running the Load Balanced Server. This is not used if you select the **Enterprise Server Mode** as **Database**.
Heartbeat Interval (ms)	3000	Specify the heartbeat interval for the Enterprise Server.
Heartbeat Timeout (seconds)	10	Specify the heartbeat timeout value for the Enterprise Server.
Average CPU Count	10	Specify the average CPU count for the Enterprise Server

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Default Value	Description
Cluster IP	• IPv4: 225.0.0.10 • IPv6: FF02::1	Displays the multicast IP address. This field is not used if you select the **Enterprise Server Mode** as **Database**. In case of Enterprise Server in Multicast mode, the Cluster IP address for IPv4 address family is *225.0.0.10* and the Cluster IP address for IPv6 address family is *FF02::1*.
Load Balanced Server	True	Set load balancing for AVEVA Work Tasks. This field is available only for the Load Balanced Server installation. To enable load balancing, set the **Load Balanced** value to *true*.
Local IP	-	Displays the local server name or IP address. IP address supports both IPv4 and IPv6 address types.
Multicast Local IP	-	Displays the multicast server name or IP address. Local IP address supports both IPv4 and IPv6 address types. This field is not used if you select the **Enterprise Server Mode** as **Database**.

Note:

- Ensure to restart the Advance Server service and other related services in all the servers and clients configured for Enterprise Server after changing the **Enterprise Server Mode**. Otherwise, the modified **Enterprise Server Mode** might not get reflected in the other servers or client.

- In an environment where both IPv4 and IPv6 addresses are configured and if both the **Local IP** and **Multicast Local IP** is the machine name, then IPv6 multicast Local IP is the **Cluster IP**, by default.

Global Settings

The Global Settings comprises of the following:

- **Communication Settings:**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

These are default email settings for sending emails.

This setting provides email IDs from and to which exception emails must be sent.

- **HTTP Settings**: This setting is used for interfacing with web services.
- **Log/Transaction settings**: This setting is used to configure log information and database transaction levels.
- **File Watcher Settings**: This setting is used for Iteration count, first attempt count and next attempt count.
- **Machine Setting:** This setting is used to configure information related to machine address on which AVEVA Work Tasks is installed.
- **Azure Setting:** This setting is used to configure Azure settings for mobile application.
- **ArchestrA Services:** This setting is used to configure ArchestrA events settings in a distributed environment.
- **OpenID Settings**: This settings is used to configure AVEVA Connect and Azure Active Directory user providers to access AVEVA Work Tasks.

HTTP Settings

The HTTP settings are for legacy AVEVA Work Tasks web service and are provided for backward compatibility.

Group Name	Field Name	Description
URL Encryption Settings	Use HTTP context	Select True to use HTTP context. Select False if you do not want to include http context info while building the secure query string parameter used internally in AVEVA Work Tasks controls to pass information from one page to another.
	Hard-Coded Correction factor	This field is used to set a flag primarily used for load testing. The date correction factor used by AVEVA Work Tasks is retrieved through a cookie from client browser. This value cannot be derived when testing using ACT or any other load test tool. If this factor is set to True, AVEVA Work Tasks always uses the date correction factor for converting UTC date time stored in AVEVA Work Tasks tables to the local time. The default value is False.
HTTP Proxy	URL	This field is used to provide the URL for HTTP Proxy.
	User	Enter a user name.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Group Name	Field Name	Description
	Password	Enter a password.
	Domain	Enter domain name.
	Bypass Local Request	Select True to bypass local requests.
	Bypass List	Enter list of URLs in the bypass list; requests from a URL in this list will be ignored.

Log or Transaction Settings

The Log or Transaction settings define the level of log information displayed in the AVEVA Work Tasks Console.

Field Name	Description
Debug Level	Select from three debug levels. • Level 1 shows minimal information in the AVEVA Work Tasks Logger Console, giving only information on the start and end of the workflow. Error messages are also shown in this mode. This is the default mode. • Level 2 shows detailed information of what is happening in the background. It will show all the messages logged from the actions. • Level 3 shows only messages indicating whether the action is executed. All error messages logged in the logger console will be sent to receiver email specified in Exception Emails.

Persist Exception in database	Select either True or False.
	• If True is selected, you will be allowed to persist the exceptions which are logged in the Logger Console to the Repository database. This in turn allows you to view exceptions at the following levels:
	  • Workflow level
	  • Workflow Instance level
	  • Activity level
	• This is the default value. If False is selected, you will not be allowed to persist the exceptions which

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description
	are logged in the Logger Console to the Repository database. Refer example scenario below.

Persist Warning in database	Select either True or False.
	• If True is selected, you will be allowed to persist the warnings which are logged in Logger Console to the Repository database. Which in turn allows you to view warnings at the following levels:
	  • Workflow level
	  • Workflow Instance level
	  • Activity level
	• This is the default value. If False is selected, you will not be allowed to persist the warnings which are logged in Logger Console to the Repository database.

Example Scenario

Consider the scenario where numerous workflows are running and an instance has failed, then, you can view the exception details from the UI instead of searching the log files.

1. In the **Farm Configuration Wizard**, go to **Service settings**.
2. In the **Global Settings** window, select **Log/Transactions** tab.
3. Select the appropriate values from drop down to enable the persistence of the exceptions or warning.

At Workflow Level:

1. In the **Enterprise Console**, click menu con, and then click **Workflows**.
   The Workflows page appears.
2. Select and right-click any workflow.
3. Click **View Exceptions** from the context menu.
   A pop-up window opens and lists all the exceptions occurred for that workflow. Here it lists all the exceptions occurred for different workflow instance.

At Workflow Instance level:

This can be viewed from BAM reports.

1. In the **Enterprise Console,** click menu icon, and then click **Reports**.
   The **AVEVA Work Tasks** Business Activity Monitoring page appears.
2. Click **Analyze**, click **Execution**, and then click **Failed**.
   This lists all the failed workflow instances.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Double-click and open any record.

   The **Process Execution view** of the record opens.

4. Click the **View** exception tab.

   All the exceptions occurred for this particular workflow instance are listed.

At Activity level:

This can be viewed from BAM reports.

- Go to **Enterprise Console**, point to **Reports**, point to **Analyze**, point to **Execution**, and then click **Failed** link.

This will list all the failed workflow instances. Select any record by double clicking the item. This will open the window where the user can see the workflow process from Analyst View, Progress view, or Execution View. Click any of the tabs and double click on the failed activity. This will open a new pop up window where details of that particular activity can be seen. In the window screen click on the **View Exceptions** tab which is present on the bottom right of the page which will lists all the exceptions occurred for this activity.

File Watcher Settings

The File Watcher settings are used to set a wait time for the file copy operation to complete. An event is generated after the copy operation. File Watcher waits for this event to trigger a workflow. The functioning of File Watcher can be better understood by the following example:

Example

A user has set the File Watcher with the following settings:

Wait for file copy iteration Count = 200

Wait for file copy first attempt (ms) = 2000

Wait for file copy next attempt (ms) = 4000

The above settings indicate that when a file copy operation is executed File Watcher does an iterative check for 200 times if the copy operation is complete. In the first attempt, File Watcher waits for 2000ms (2 sec) to check if the copy operation is complete. If it is complete the iterative check is terminated and a File Watcher event is triggered. If the copy operation is not completed the iteration count is incremented by one and during the second and subsequent attempts the File Watcher waits for 4000ms (4 sec). This activity continues till the copy operation is complete.

Thus File Watcher will wait for approximately 13 minutes (199 x 4000ms+ 2000ms) for a file copy operation to complete. An event is triggered as soon as the file copy operation is complete and File Watcher terminates the iterative check.

The example above indicates that you have to set the File Watcher values based on your requirements. If you need to copy a file with a large file size, set the Wait for file copy iteration Count to a higher value or increase the values for the fields Wait for file copy first attempt or Wait for file copy next attempt. If the copy operation is not completed within the set time, the message Could not find the copying file is displayed. A large file will require a longer wait time.

The following table lists the default values with field description:

Field Name	Description
Wait for file copy iteration Count	This field is used to set the iteration value for a file

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description
	copy process. The default value is 200. **Example** The value 200 indicates that the process will execute 200 times if the file copy process is not complete.
Wait for file copy first attempt (ms)	The value in this field indicates the wait time after a file copy command is executed for the first time. This wait time is provided for an event to be generated after the file copy process is complete. The value specified is in milliseconds. The default value is 2000. **Example** The value 2000 indicates that the wait time for copying files in the first attempt is two seconds.
Wait for file copy next attempt (ms)	The value in this field indicates the wait time for subsequent attempts. The value specified is in milliseconds. The default value is 4000. **Example** The value 4000 indicates that the wait time for subsequent attempts is four seconds.

Machine Setting

You use the **Machine Setting** tab to know the information related to machine address on which AVEVA Work Tasks is installed and **Service Monitor Setting** tab to define the port used by AVEVA Work Tasks Service Monitor EXE.

Group Name	Field Name	Description
Machine Setting	Machine Address	Enter the machine address on which AVEVA Work Tasks is installed. If the **Database Type** is **SQL Server**, then you can specify an IPv6 address for the **Machine Address**.
Service Monitor Setting	Service Monitor Port	Enter the port number. This is the default port used by the AVEVA Work Tasks Service Monitor EXE.

Azure Setting

Azure Setting is required for the AVEVA Work Tasks Pro App to access the AVEVA Work Tasks server through Internet. The mobile application communicates with the AVEVA Work Tasks server through Windows Azure

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Service Bus. This platform provides the channel through which the AVEVA Work Tasks Pro App can communicate with the On-Premise AVEVA Work Tasks Service.

To access the AVEVA Work Tasks features in a mobile device, you must configure the Service Bus Listener. A Service Bus Listener enables a secure communication mechanism between AVEVA Work Tasks Pro App and the Enterprise Console. After you configure a service bus listener, a Service Bus Listener URL is generated. Mobile users can access the application in mobile devices using the Service Bus Listener URL.

For more information, refer the following topics:

- User details: See Provide Server Details to Mobile User.
- Accessing AVEVA Work Tasks Pro Outside Corporate Network and within the Corporate Network: See Accessing Work Tasks Pro Outside Corporate Network and within the Corporate Network

Note: The Service Bus Listener URL must be unique for every Farm. If you are using the same URL for multiple Farms, ensure that the combination of the Listener Namespace and Service Bus Namespace is unique.

Before configuring Azure setting, ensure that you complete the following:

1. Create Enterprise Console with Anonymous Authentication
2. Configure Service Bus Listener

Create Enterprise Console with Anonymous Authentication

To successfully configure and use AVEVA Work Tasks Pro, we recommend you to create a new Enterprise Console that specifically caters to the mobile services.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
2. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The AVEVA Work Tasks Farm Configuration Wizard appears.
3. Click **Next**.

   The Edit AVEVA Work Tasks Farm Settings window appears.
4. Select **Service Settings**.
5. Click **Next**.

   The Services/Global Settings window appears.
6. Click the edit icon near **Enterprise Console** option.

   The Enterprise Console Settings window appears.
7. Click **New EC**.
8. In the **New Website** text box, enter a name for the Enterprise Console.

   For example, enter name as *MobileEnterpriseConsole*.
9. Click the edit icon near the **Create at** option to change the port number, if required.

   The default port is 8000.

   The IIS Structure window appears.
10. Select the path where you want to create a Enterprise Console.

    For example, select *AVEVAWorkTasksSites*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

11. Click **OK**.

    The selected path is displayed in the **Create at** text box.

    The mapped repositories are displayed in the **Repositories**.

12. Select the repository(s) that you want to map the newly created site.

    Use **Ctrl** or **Shift** keys to select multiple repositories.

    Note: The mapped repositories are auto-populated on entering the URL in the AVEVA Work Tasks Pro application. However, in an shared host environment, the repositories and providers are not auto-populated. Thus, the administrator must provide the repository and provider details to the AVEVA Work Tasks Pro user. You can configure the shared host environment through the following steps:

    a. Go to SKService table of Farm Database.

    b. In the property collection of **Global Settings**, update the property **issharedhostenvironment** to *True*.

13. Select the required **Authentication Type** as *Anonymous*.
14. Click **Save**.

    The created Enterprise Console appears in the **Existing Enterprise Consoles** section.

15. Click **OK**.
16. Click **Finish** in the Farm Configuration Wizard window.

To set anonymous authentication for existing Enterprise Consoles

You can set the anonymous authentication for existing Enterprise Consoles through IIS.

1. In the Administrative Tools window, click **Internet Information Services (IIS) Manager**.

   The Internet Information Services (IIS) Manager window appears.

2. Click **Sites**, click **Enterprise Console**, and then select **Authentication** from the right pane.
3. Click **Open Feature** on the Actions pane.
4. On the Authentication pane that opens, set **Anonymous Authentication** as *Enabled*.

Configure Service Bus Listener

To configure Service Bus Listener using the Farm Configuration Wizard:

1. Click the **Azure Setting** tab.

   The **Service Bus Listener Configuration for Mobile** section appears.

2. Enter the **Service Bus Namespace** obtained from the Windows Azure Management Portal, in this field.
3. Select the required **Enterprise Console** from the drop-down field.

   Note: Only the Enterprise Console of **Anonymous Authentication Type** are listed in this field.

4. Enter the issuer value of the service bus in the **Issuer** field.
5. Enter the Issuer key of the service bus in the **Key** field.
6. In the **Listener Namespace** field, the name of the selected Enterprise Console will be populated.

   This field is in read-only mode.

   Ensure that the computer name does not have a hyphen (-).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. Select the **Activate** check box to activate the listener.
8. Select the **Enable Enhanced Security** check box to enforce additional security at the service bus level.
9. Click **Save** to save the configuration.

   This configuration appears in the Existing Service Bus Listener Configurations section.
10. Click **Test Listener Status** to test the Service Bus Listener Configuration.

Note: The Service Bus Listener Url generated above will be used while configuring the AVEVA Work Tasks Pro application.

To Access Azure Settings in AVEVA Work Tasks Pro App

You can configure the Azure settings such as Enterprise Console path, Repository name, Provider name, and User name in the Farm Configuration Wizard. You can generate a QR code to store these settings. This QR code can then be scanned by the AVEVA Work Tasks Pro App to automatically fill the settings in the Settings screen.

1. Click the **Azure Setting** tab.

   The **Service Bus Listener Configuration for Mobile** section appears.
2. Click **Generate QR Code**.

   The **Scan QR Code** dialog box appears.
3. Perform either of the following:
   - If you have selected the **Enable Enhanced Security** check box, enter the user name in the **User Name** box, and then click **Generate**. The **Scan QR Code** dialog box appears.
   - If you have not selected the **Enable Enhanced Security** check box, go to step 4.
4. In the **Scan QR Code** dialog box, click **Download**.

   The **Save As** dialog box appears.
5. Enter the **File name** for the QR code and select the **Save as type** as **PNG**.
6. Click **Save**.

   The QR Code is generated successfully.
7. Scan this QR Code in the AVEVA Work Tasks Pro App.

   The values configured for the following settings in the Farm Configuration wizard are considered by default.
   - Enterprise Console path
   - Repository name
   - Provider name
   - User name
8. Enter the password.

Note:
- The Shared Access Signature (SAS) token is valid for 1 year and is automatically renewed after this duration.
- The secured Service Bus URL can be accessed only using QR Code. You cannot enter this URL in the app and log in.

Generate SAS Token

You must generate the QR code with Shared Access Signatures (SAS token) for users who can scan the code to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

configure the Work Tasks Pro app.

For more information on SAS, see https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-sas.

To generate SAS token using the Farm Configuration Wizard

1. Click the **Azure Setting** tab.
   The **Service Bus Listener Configuration for Mobile** section appears.
2. Click **Generate QR Code**.
   The **Scan QR Code** dialog box appears.
3. Enter the user name for whom you want to generate the QR code, and then click **Generate**.
   The QR code contains SAS token for that user.
4. Click **Download** to save the QR code.
   Scan the QR code in the Work Tasks Pro App. For more information, see Configure Service Bus Listener.

To generate SAS token using the Enterprise Console

1. Log on to the Enterprise Console.
2. Point to the user menu, and then click **View QR Code**.
   The **View QR Code** dialog box appears.
3. Click **Download** to save the QR code.
   Scan the QR code in the Work Tasks Pro App. For more information, see Configure Service Bus Listener.

Provide Server Details to Mobile User

After the Service Bus details are configured, you can give the server details to the mobile application user, to access the application on mobile devices:

Server Details:

- **URL**: The Azure URL that is displayed in the Service Bus Listener URL column of the Existing Service Bus Listener section. This URL is obtained after configuring the Service Bus Listener for a particular Enterprise Console.
- **Provider**: The Provider name should be the same as the Instance Name of the repository. The Instance Name can be taken from the Instance Name column of the SKEntityDataSourceProvider table of the repository database.

ArchestrA Services

Consider a distributed environment where **AVEVA Work Tasks** is installed in one computer (say node 1) and both **ArchestrA Service Bus** (ASB) and **System Platform IDE** are installed in another computer (say, node 2). In this environment, use the **ArchestrA Services** tab to configure the ArchestrA Events setup so that node 1 can poll event details from node 2. Here node 1 acts as the ASB runtime node and node 2 acts as the ASB solution (or /Domain) node.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

However, if **AVEVA Work Tasks**, **ArchestrA Service Bus** (ASB), and **System Platform IDE** installed on the same computer, you do not need to perform these configurations.

In a distributed environment, ArchestrA services with System Platform 2014 and System Platform 2014 P01 is not supported.

Prerequisites

- In node 2, generate a registration passphrase for the ASB Service Repository node name. You can set the temporary passphrase on ASB Service Repository node using the **aaASBRegistrationUtility.exe** available in the following path:

  C:\Program Files (x86)\Common Files\ArchestrA\Services

Steps to configure the ArchestrA Services for a distributed environment

1. Click **Configure**.

   The **ASBInstall.exe** runs in the background to configure the ASB Install Utility.

   Note: After installing AVEVA Work Tasks, the ASB Install Utility is available in the following path:

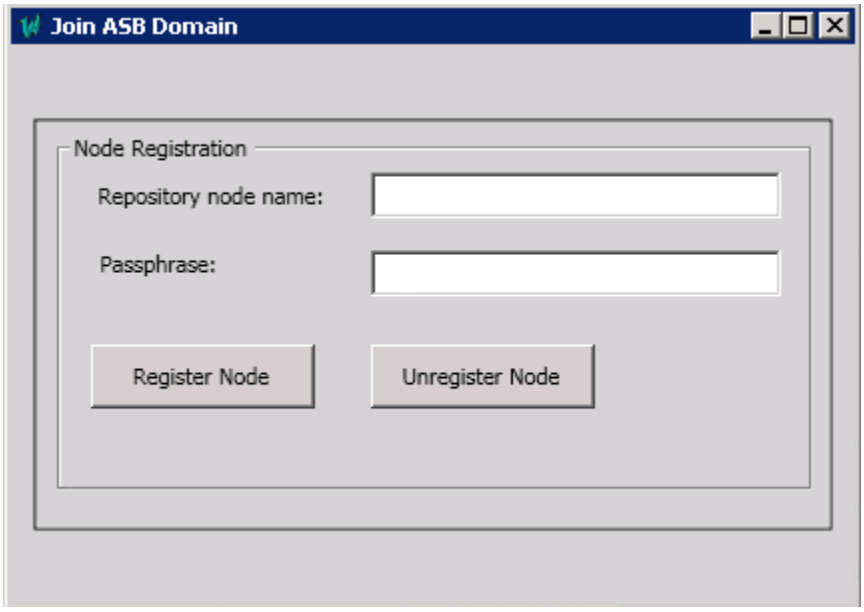
   C:\Program Files (x86)\Common Files\ArchestrA\Services\ASBInstall.exe

   You can run the **ASB Install Utility** with command line argument as follows:

   ```
   C:\Program Files (x86)\Common Files\ArchestrA\Services\ASBInstall.exe install
   ```

   This is an alternate approach to start the **ASB Install Utility**, if configuration fails.
2. Click **Launch** to connect node 1 and node 2.

   The Join ASB Domain dialog box appears.
3. Enter the **Repository node name** and **Passphrase**.
   - **Repository node name**: Indicates the ASB Service Repository node name. Here, name of node 2.
   - **Passphrase**: Indicates the registration passphrase generated for the ASB Service Repository node name.

4. Click **Register Node**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Click **Start** to start the **ArchestrA Watchdog Service**.
6. Click **Test Connection** to verify the connection.

   After upgrading to the required version of AVEVA Work Tasks, if **Test Connection** fails in a distributed environment then you must restart the computer.
7. If the connection is successful, click **OK**.

For information about associating a workflow with ArchestrA Event, see the *ArchestA Events* topic in the *User Guide*.

OpenID Settings

You can access Central Configuration and Enterprise Console through AVEVA Connect and Azure Active Directory user authentication.

Use the **OpenID Settings** tab to configure AVEVA Connect and Azure Active Directory authentication providers.

To configure AVEVA Connect provider, see AVEVA Connect Settings.

To configure Azure Active Directory provider, see Azure Active Directory Settings.

Redirect URIs for AVEVA Work Tasks Pro

In Microsoft Azure portal, where AVEVA Work Tasks Pro app is registered, you need to generate Redirect URI to get the authentication response.

To generate Redirect URI, provide the following details under **Authentication**:

- For iOS, provide **Bundle ID** as **com.schneiderelectric.worktaskspro**
- For Android, provide **Package name** as **com.schneiderelectric.worktaskspro** and **Signature hash** as **VXouuql0JzjKih7eCZqABlKW0Nw=**

For UWP, provide the Redirect URI as **https://www.worktaskspro.com**

Note:

- AVEVA Work Tasks needs to be registered with the selected Authentication Provider. Ensure that the following application URIs are added as Allowed Callback URIs.
For example:
Enterprise Console Redirect URIs:
-- https://{machinename}:{port}/EnterpriseConsole/BPMUITemplates/Default/Repository/Site/RepositoryList.aspx
-- https://{machinename}:{port}/EnterpriseConsole/BPMUITemplates/Default/Repository/Site/Login.aspx
Central Config Redirect URI: https://{machinename}:{port}/CentralConfig/CentralAdministration.aspx
Work Tasks Pro Redirect URI: https://www.worktaskspro.com

- You can switch to any of the authentication types (Windows, Anonymous, Azure Active Directory, or AVEVA Connect) to access Enterprise Console. To switch, you need to reconfigure Enterprise Console settings from Farm Configuration Wizard and set the required Authentication Type. But, in the case of Windows authentication type, Azure Active Directory and Aveva Connect repositories will not work.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

AVEVA Connect Settings

AVEVA Connect can be used as a user provider for creating a repository in AVEVA Work Tasks and can also be used to access AVEVA Work Tasks Pro app through AVEVA Connect.

In AVEVA Work Tasks, you can configure AVEVA Connect using the Farm Configuration Wizard. With this setting, the user is authenticated against AVEVA Connect provider and can access the AVEVA Work Tasks features.

To configure AVEVA Connect Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, point to **Work Tasks**, and then click **Farm Configuration Wizard**.

   The **AVEVA Work Tasks** Farm Configuration Wizard appears.
2. Click **Next**.

   The Edit **AVEVA Work Tasks** Farm Settings window appears.
3. Select **Service settings.**

   The **Enterprise Server settings** can be selected if the Farm Server Installation is using Enterprise Server Installation.
4. Click **Next**.

   The Services/Global Settings window appears.
5. Click against **Global Settings**.
6. In the **Global Settings window**, select **OpenID Settings** tab.
7. In the **Authentication Provider** drop-down list, select **AVEVA Connect**.
8. In the **Account ID** box, enter the account ID that you will use to access AVEVA Connect.
9. For **Client for User Authentication**,
   a. In the **Client ID** box, enter the client id for user authentication.
   b. In the **Client Secret** box, enter the client secret key for user authentication .
10. For **Client for Service Authentication**,
    a. In the **Client ID** box, enter the client id for service authentication.
    b. In the **Client Secret** box, enter the client secret key for service authentication.
11. For **Client for Work Tasks scope (ac_worktasks)**,
    a. In the **Client ID** box, enter the client id for Work Tasks scope.
    b. In the **Client Secret** box, enter the client secret key for Work Tasks scope.
12. For **Client for Work Tasks Pro Mobile App**,
    a. In the **Client ID** box, enter the client id for Work Tasks Pro app.
    b. In the **Client Secret** box, enter the client secret key for Work Tasks Pro app.
13. Click **Save** and then click **OK**.

    You can now update or create the repositories and configure Central Configuration and Enterprise Console with AVEVA Connect user provider.

    To create repository using AVEVA Connect as user provider, see Create a Repository.

    To configure Central Configuration with AVEVA Connect, see Define Central Configuration.

    To configure Enterprise Console with AVEVA Connect, see Add a New Enterprise Console.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:

- The user or role search is always using the starting character that you enter and does not have any search option.

- Role condition set through Filter condition of To property in the Human Activity is not supported.

- System Platform IDE with the AVEVA Connect provider is not supported.

- To call an external Web API that uses an Access Token, that token will have **ac_worktasks** as a **scope**, and the **audience** will be **https://worktasks.connect.aveva.com**.

Azure Active Directory Settings

Azure Active Directory can be used as a user provider for creating a repository in AVEVA Work Tasks and can also be used to access AVEVA Work Tasks Pro app through Azure Active Directory.

Before configuring the Azure Active Directory settings in AVEVA Work Tasks, register the app in Microsoft Azure directory.

App Registration

You need to register the app in Microsoft Azure to generate the Client Id, Tenant Id, and Object Id.

To register the App:

1. Access the Microsoft Azure portal.
2. Click **View** under **Manage Azure Active Directory** on the **Microsoft Azure Home** page.
3. Click **App registrations** in the left navigation pane.
4. Click **New registration** tab.
   a. In the Name box, enter the display name for this application.
   b. For **Supported account types**, select **Accounts in this organizational directory only (AVEVA Solutions Limited only - single tenant)**.
   c. For **Redirect URI (optional)**, specify the URI to get the authentication response.
5. Click **Register**.

   The app is registered in the Azure Directory.

Azure Active Directory Settings

In AVEVA Work Tasks, you can configure Azure Active Directory using the Farm Configuration Wizard. With this setting, the user is authenticated against Azure Active Directory provider and can access the AVEVA Work Tasks features.

To Configure Azure Active Directory Settings

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, point to **Work Tasks**, and then click **Farm Configuration Wizard**.

   The **AVEVA Work Tasks** Farm Configuration Wizard appears.
2. Click **Next**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Edit **AVEVA Work Tasks** Farm Settings window appears.

3. Select **Service settings.**

   The **Enterprise Server settings** can be selected if the Farm Server Installation is using Enterprise Server Installation.

4. Click **Next**.

   The Services/Global Settings window appears.

5. Click against **Global Settings**.
6. In the **Global Settings window**, select **OpenID Settings** tab.
7. In the **Authentication Provider** drop-down list, select **Azure Active Directory**.
8. In the **Tenant ID** box, enter the tenant ID of the Azure Active Directory instance.
9. In the **Client ID** box, enter the client ID of the Azure Active Directory instance.
10. In the **Client Secret** box, enter the client secret key of the Azure Active Directory instance.
11. Click **Save** and then click **OK**.

Note:
- If you log out from the Central Configuration (CC) site, you are also logged out from the Enterprise Console (EC) site. To access the EC site again, you need to access the EC URL, provide the access credentials, and then continue to use the EC site.

- The Azure Active Directory provider is listed in the Resource Provider drop-down list. But if you create the repository with this Azure Active Directory provider, it will work only when you configure Azure Active Directory through the Farm Configuration Wizard.

- Once you configure Azure Active Directory from Farm Configuration Wizard,
-- the Application Pool of Central Configuration changes to AVEVA Work Tasks Integrated Mode.
-- while accessing the Enterprise Console and Central Configuration, you are redirected to the Microsoft Online Page for authentication.

- When the environment is configured to Azure Active Directory, the following are not supported:
-- Synchronize Users
-- Electronic Signature
-- Disable Users from Azure Active Directory
-- Security Workflow
-- Wizard for Human Activities (In Assign Actors property of Human Activities while building the filter condition, the ‘IN’ and ‘Not In’ operators are not supported.)
-- Role Condition Set through Filter Condition of To Property in the Human Activity
-- The user’s display name with characters like # and ^
-- System Platform IDE

- The user or role search uses the starting character that you enter and does not have any search.

- You cannot have both the AVEVA Identity Manager (AIM) and Azure Active Directory user providers together.

- Scheduler Event Association installed through package does not work in Azure Active Directory when logged-in user's display name has special characters.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

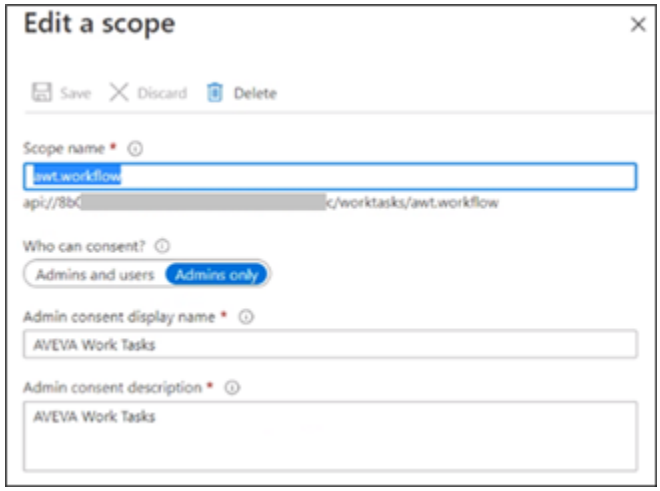
- It is recommended to use https sites for Azure Active Directory repository.

Create Custom Scope in Azure Active Directory

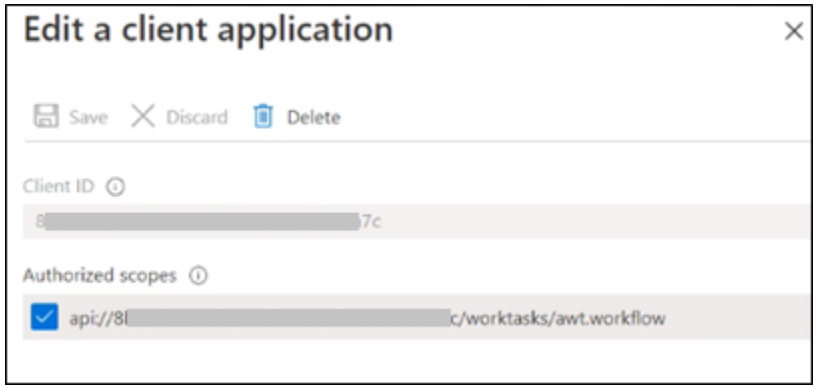
You use the custom scope in the Identity provider so that you can pass the same token to the Web API used in the workflow.

To create a custom scope in Azure Active Directory:

1. Access the **Microsoft Azure portal**.
2. Click **App registrations** in the left navigation pane.
3. Select the registered application.
4. Click **Expose an API** in the left navigation panel.
5. Click **Application ID URI** link and make sure the URI name has correct **client id** and **worktasks**. For example, *api://<client id>/worktasks*
6. To define the scope, click a link under **Scopes** for the registered application.
   a. Enter **awt.workflow** as the scope name.
   b. Enter the other details.



7. To authorize client applications, click a link under **Client Id**.
   a. Select the check box for the created custom scope under **Authorized scopes**.



8. Click **Save**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The custom scope for the Work Tasks application gets created in the Azure Active Directory.

Note: Make sure the Client ID matches the client id mentioned in the Farm Configuration Wizard.

Retrieve Additional User Properties for Azure Active Directory User Provider

You can retrieve additional user properties in Azure Active Directory (AAD) provider so that you can use that property in User Search and Assignment.

You need to add the additional property in the Definition XML as,

```
<property name="mailNickname" type="string" displayName="MailNickName"
entitymappedproperty="UserMailNickName" />
```

To Add Additional Property in the Definition Xml

1. In **Central Configuration**, go to **Repository Provider List**.
2. Select the repository configured with **Azure Active Directory Provider** and expand it.
3. Access the **Definition XML** of **User Entity** by clicking on the **Click here** option beside **User Entity**.
4. Edit the **Definition XML** and add the property for the new field.

   Example:

   ```
   <property name="UserMailNickName" displayname="MailNickName" type="string" />
   ```
5. Select the provider **Definition XML** and click the **Definition XML**
6. Edit the **Definition XML** and add,

   ```
   <property name="mailNickname" type="string" displayName="MailNickName"
   entitymappedproperty="UserMailNickName" />
   ```
7. Add the Locales text for the display name as,

   ```
   aadprovider.MailNickName:MailNickName
   ```
8. Do an IISReset to reflect the added property.

Auto Resume Settings

The Workflow Engine, a windows service, is responsible for workflow execution and routing of activities. Workflow instances and the activities within the workflow are assigned different status values based on the execution state. These status values are persisted in the database.

This section highlights the various methods available to resume specific workflow instances.

List of Workflow Status Values

Following is the list of status values which are assigned to the workflows during execution:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Status	Description
EP	Execution Pending This status is assigned to a workflow, when the workflow execution has not begun.
EX	Execution in Progress This status is assigned to a workflow, if any one of the workflow activity starts its execution.
SL	Sleep Mode The workflow is in Sleep mode when all the pending activities are in sleep mode.
FN	Finished Successfully The Workflow Execution is considered to be complete, if all the activities have completed execution.
FE	Workflow Execution Finished with Errors If any activity has an error while execution, the workflow stops with this status.
AB	Aborted This status is assigned to a workflow, when it gets aborted while execution.
PA	Paused This status is assigned to a workflow, when it is paused while execution.

Following is the list of status values assigned to the activities during runtime:

Status	Description
P	Execution Pending
E	Activity Executing
S	Sleep
C	Execution Completed
Z	Next action initiated
K	Activity killed
U	Activity paused

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

What happens when a workflow fails to execute?

When an activity within a workflow instance fails, the workflow instance is marked as Finished with Errors (FE) and the activity is marked as executing (E).

When can a workflow fail and finish with errors?

Following are some of the possible reasons for a workflow to fail:

- An activity in the workflow fails, which in turn can happen due to:
  - Invalid configuration settings for the activity properties
  - Uninitialized variable being used in the activity
  - Slow network connectivity
- Slow network connectivity to the database server

How can a workflow be modified and resumed?

If a workflow fails due to invalid configuration settings or due to improper use of variables, you can modify the workflow accordingly and then republish it.

For example, consider a workflow which has finished with errors as shown in the image given below. The activity Estimate due time has failed to execute. The user can check AVEVA Work Tasks log files or event viewer to identify the reason of failure. The workflow can then be modified accordingly, and then republished.

Different methods to resume workflow instances

- Using AVEVA Work Tasks APIs
- Using BAM reports
- Using the Auto Resume Feature

Using AVEVA Work Tasks API

AVEVA Work Tasks APIs can be used to resume workflow instances in FE state.

Given below is the sample code:

- **Namespace**: Workflow.NET.Engine
- **Assembly**: Workflow.NET.NET2.dll.

```
static void ResumeWorkflowWithErrors(int executionId, string application, string
workflow)
{
Client client = null;
try
{
client = new Client(application, workflow);
if (client.CanResumeWorkflowWithErrors(executionId))
client.ResumeWorkflowWithErrors(executionId);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
catch (Exception exp)
{
Console.Write(exp.ToString());
}
finally
{
if(client != null)
client.Close();
client = null;
}
}
```

Ensure that the client object is closed in the final block.

To use BAM Reports

Follow the steps given below to resume FE workflow instances from BAM reports:

1. In the **Enterprise Console**, navigate to **Reports.**

   The **AVEVA Work Tasks** Business Activity Monitoring page appears.

   The tree view on the left side gets loaded with reports for different workflow instances.
2. Click **Failed** node available under **Analyze** > **Execution**.

   A report of all the workflow instances that have finished with errors appears.
3. Double-click on any of the workflow instance.

   The workflow progress view pop up window appears. The left side of the pop up window has icons to pause, abort, resume paused workflows, and to resume FE workflows.
4. Click the icon to resume workflow finished with errors.

   This would resume the activity which has failed to execute and the workflow would progress to the next state.

Automatically resuming from Workflow Engine

The workflow instances that have pending events or are in EP or FE state can be automatically resumed using the Auto Resume Feature. By default all the repositories that are configured in the farm are considered for resuming the workflow instances.

Auto Resume Settings

The **Auto Resume Settings** tab of the AVEVA Work Tasks Engine Settings is used to configure properties for automatically resuming workflow instances that have pending events or are in FE or EP state.

The following properties are applicable to all three types of workflow instances (FE, EP, Pending Workflow Events):

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Select Repositories	All the repositories that are mapped to the current farm are listed here. The user can either select the Select All check-box to resume workflow instances from all the repositories or select the required repositories from the list.
Resume workflow instances that have pending events or are in FE or EP status from	Select a date which would be considered by the Workflow Engine while fetching the workflow instances. From the selected date onwards, all the workflow instances that have pending events or are in EP or FE status will be resumed automatically. This date selected here is compared with the LastUpdatedDateTime field from SWExecute table.
Enable logging of auto resume messages	It enables you to control the log messages in the logger console whenever the engine tries to resume the respective workflow instances.

FE Workflow Instances

The details of failed workflow instances that are automatically resumed by the Workflow Engine are stored in the SWFECount table. This table keeps track of the execution IDs of workflows that have finished with errors (FE) and the number of retries that engine has made to re-execute the failed workflow. After resuming the workflow, if the workflow executes successfully then the corresponding record is deleted from the SWFECount table. If the workflow instance is still in FE state even after the set number of retries, then an exception is emailed to the Admin configured in the To Email property in the farm configuration settings.

Configurations for Resuming FE Workflows

Given below is the list of properties that needs to be configured for resuming FE workflows:

Field	Description
Number of retries to resume FE workflow instances	The number of attempts that the Workflow Engine should make to resume the FE workflow instances. For example, if this value is set to 5 then the Workflow Engine would make 5 attempts to resume each of the FE workflow instance. If this value is set to 0 (zero) then, then auto resume of FE workflow instances does not occur.
Remove resume history when FE workflow instances are completed	If the value is selected as True, the history of resumed FE workflow instances which have completed successfully would be deleted from the SWFECOUNT table. If the value is selected as False, the history of resumed FE workflow instances which have completed

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	successfully would be retained in the SWFECOUNT table.
Time interval to fetch FE workflow instances	It is used to set the time interval for resuming FE workflow instances. The value entered should be greater than 5 minutes.
No. of FE workflow instances to be fetched in the specified time interval	It is used set the number of workflow instances to be fetched for auto resume in the specified time interval.

EP Workflow Instances

Given below is the list of properties that needs to be configured to resume EP workflows:

Field	Description
Enable auto resume of EP workflow instances	This property is used to enable or disable auto resume of EP workflow instances. **Note**: The **Enable auto resume** setting does not apply to the scheduled workflows.
Fetch workflow instances which have been in EP state for	This property is used to set the minimum time for which the workflow instances should have been in EP state. The value should be entered in days.hr:min:sec format. If this value is set as 5, the Workflow Engine tries to fetch the workflow instances which have been in EP state for at least 5 hours. The Workflow Engine calculates the time for which the workflow instances have been in EP state by comparing the LastUpdatedDateTime value from SWExecute table with the current time. The default time for which workflow instances have been in EP state is 5 hours. If the time set is less than 5 hours, then by default, the Engine considers 5 hours as the minimum time.
Time interval to fetch EP workflow instances	It is used to set the time interval for resuming EP workflow instances. Minimum value should be greater than 5 minutes.
No. of EP workflow instances to be fetched in the specified time interval	It is used set the number of EP workflow instances to be fetched for auto resume in the specified time interval.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Pending Workflow Events

Given below is the list of properties that needs to be configured for resuming Pending Workflow Events.

Field	Description
Enable auto resume of pending workflow events	This property is used to enable or disable auto resume of pending workflow events. **Note**: The **Enable auto resume** setting does not apply to the scheduled workflows.
Minimum time for which the workflow events are in Pending state	This property is used to set the minimum time for which the workflow events should be in Pending state. If this value is set as 0.5:0:0, the workflow engine will fetch the events which are in Pending state for at least 5 hours. The workflow engine calculates the time for which the workflow events have been pending by comparing the LastUpdatedDateTime value from SWExecute table with the current time. The default time for which workflow events should be in pending state is 5 hours. If the time set is less than 5 hours, then by default, the engine considers 5 hours as the minimum time.
Time interval to fetch workflow instances with pending events	It is used to set the time interval for resuming workflow instances with pending events. Minimum value should be greater than 5 minutes.
No. of workflow instances with pending events to be fetched in the specified time interval	It is used set the number of workflow instances with pending events to be fetched for auto resume in the specified time interval.

Cache Refresh Settings

The performance of Enterprise Console can be improved by reusing previously fetched resources. The caches reduce latency and network traffic and thus lessen the time needed to display a representation of a resource. By making use of HTTP caching, Enterprise Console become more responsive.

Enterprise Console Web Application makes use of clientCache element of the staticContent element, which update the HTTP Header `Cache-Control` with `must-revalidate` and `max-age` request directives. The value set for the `max-age` request directive is 1 hour, which can be set using the cacheControlMaxAge attribute of the `clientCache` element.

After the product is installed, you can change the cacheControlMaxAge attribute value as per your requirement. The Enterprise Console Web Application's web.config file related to the caching is usually located at <product installation path>\Web\BPMUITemplates\Default\web.config and would contain the `clientCache` settings as below:

```
<system.webServer>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<staticContent>
<clientCache cacheControlCustom="must-revalidate" cacheControlMode="UseMaxAge"
cacheControlMaxAge="0.01:00:00" />
</staticContent>
</system.webServer>
```

It is recommended to set the `cacheControlMaxAge` attribute value to 0 for a staging or non-production environment, as there would be frequent changes to the files/resources used by the site, which is used for development.

Note: If you change the .js, .css, or image files, the changes for the same would be reflected in the browser as per the `cacheControlMaxAge` attribute in the web.config file.

Administration Tasks

This section provides information on:

- Manage Databases
- Configure Services
- Cryptography Settings
- Administer Web Servers
- Work with Central Configuration
- User Providers
- Debug

About Forms Designer

The Forms is an easy to use drag-and-drop interface used for designing forms. You can use the Forms Designer to quickly design forms, maintain form templates and define business logic. The Forms are used to trigger the Workflows.

The Forms does not require a separate installation. This is a part of the core product installation.

To use the Forms Designer, the Forms Server license has to be acquired. The Forms Server license is released when the Client service is stopped.

View Installed Products

You can use the **View Installed Products** feature to view the version number of AVEVA Work Tasks and its connectors that are installed on the system. The **View Installed Products** feature is a part of the Farm Configuration Wizard.

Complete the following tasks to get a list of the installed products:

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
2. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** screen appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Click the **Installed Products** link.

   The View Installed Products window appears.
4. Click the name of the product from the **Installed Products** section.

   This will populate the product details, such as the product name, product version, and the installation type in the **Product Details** section.
5. To copy the information, click the **Copy Info.**
6. Click **OK** to exit.

Manage Databases

This topic provides information on editing Database User Settings, changing the pool size, changing the password for Windows Active Directory and for SQL user credentials. Consider the following scenarios for editing Database User Settings:

Scenario 1

The Database User password has been modified in the Active Directory. This modification has to be updated in the database. If not modified, the user will not be allowed to log in to the Farm.

Scenario 2

The user login credentials has been modified in SQL database instance. This modification has to be updated in the Farm database. If not modified, the user will not be allowed to log in to the Farm.

Change Database User Settings for Windows Authentication

The database user settings can be authenticated by Windows or SQL. In case of Windows Authentication, if the user name and password is changed in Active Directory then the same has to be updated using the Farm Configuration Wizard. You can also use this option to switch from one authentication type to the other.

You need to ensure that there are no running workflows while changing database user settings.

Changing Database User Settings for Windows Authentication

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **AVEVA Work Tasks** Configuration Wizard appears.
2. Click **Next**.

   The Edit **AVEVA Work Tasks** Farm settings screen appears.
3. Select **Database User Settings**.
4. Click Next.

   The Database User Settings screen appears.
5. Enter the updated **User Name, Password**, and **Domain**.
6. Click **OK**.

   A message to ensure that the login has required permissions appears.
7. Click **OK**.

   A message indicating Database User settings edited successfully appears.
8. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Stop the following AVEVA Work Tasks services:
    a. Client Service
    b. Communication Service
    c. Task Schedule
    d. Mobile Notification Service
    e. Workflow Engine
10. In an Enterprise Server system, ensure to stop the **Advance Server Service**.
11. In a client computer, ensure to stop the **Client service**.
12. In a client machine which is connected to an Enterprise Server, ensure to stop the **Advance Server Service** and **Client Service**.
13. Close all AVEVA Work Tasks related sites and applications.
14. To stop IIS, run **Command Prompt** as an **Administrator**, and type **iisreset /stop**.
15. Delete the *WorkflowManagementTemp* folder from *C:\Windows\Temp*.
16. Delete the AVEVA Work Tasks Cache files and folders from *C:\Windows\Microsoft.NET\Framework64\v4.0.XXXXX\Temporary ASP.NET Files*.
17. Start the AVEVA Work Tasks related services.
18. To start IIS, run **Command Prompt** as an **Administrator**, and type **iisreset /start**.
19. Click **OK**.

Change SQL User Credentials for SQL Authentication

Database User settings can be authenticated by Windows or SQL. In case of SQL authentication, if the user name/password is changed in the SQL database then the same has to be updated using Farm Configuration Wizard. You can also use this option to switch from one authentication type to the other.

To change SQL User Credentials for SQL Authentication

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **AVEVA Work Tasks Farm Configuration Wizard** appears.
2. Click **Next**.
3. Select **Database Settings**.
4. Click **Next**.

   The Connect to existing database screen appears.
5. From the **Connection String** option, modify the **User Id** and **Password** for connecting to the database.
6. Click **Test Connection**.
7. Click **OK**.
8. Click **Next**.
9. Stop the following AVEVA Work Tasks services:
    - Client Service
    - Workflow Engine
    - Task Scheduler

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Communication Service
- Mobile Notification Service

10. In an Enterprise Server system, ensure to stop the **Advance Server Service**.
11. In a client computer, ensure to stop the **Client Service**.
12. In a client machine which is connected to an Enterprise Server, ensure to stop the **Advance Server Service** and **Client Service**.
13. Close all AVEVA Work Tasks related sites and applications.
14. To stop IIS, run **Command Prompt** as an **Administrator**, and type **iisreset /stop**.
15. Delete the *WorkTasksTemp* folder from *C:\Windows\Temp*.
16. Delete the AVEVA Work Tasks Cache files and folders from *C:\Windows\Microsoft.NET\Framework64\v4.0.XXXXX\Temporary ASP.NET Files*.
17. Start the AVEVA Work Tasks related services.
18. To start IIS, run **Command Prompt** as an **Administrator**, and type **iisreset /start**.

    The user credentials are updated.

See Also: Create Farm Database

Change Pool Size of Farm Connection Settings

To change Pool Size of the Farm

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **AVEVA Work Tasks** Farm Configuration Wizard appears.
2. Click **Next**.
3. Select **Database Settings**.
4. Click **Next**.

   The Connect to existing database screen appears.
5. Select the **Connection String** option.
6. Enter the pool size in the **Advanced Settings** field.

   For example, type Connection Timeout = 30.
7. Click **Test Connection**.
8. Click **OK**.

   This will change the pool size of the Farm Database.
9. Click **Next**.

   The message, *Connection settings edited successfully* appears.
10. Stop the following AVEVA Work Tasks services:
    a. Client Service
    b. Communication Service
    c. Task Schedule
    d. Mobile Notification Service

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

e. Workflow Engine

11. In an Enterprise Server system, ensure to stop the **Advance Server Service**.
12. In a client computer, ensure to stop the **Client service**.
13. In a client machine which is connected to an Enterprise Server, ensure to stop the **Advance Server Service** and **Client Service**.
14. Close all AVEVA Work Tasks related sites and applications.
15. To stop IIS, run **Command Prompt** as an **Administrator**, and type **iisreset /stop**.
16. Delete the *WorkflowManagementTemp* folder from *C:\Windows\Temp*.
17. Delete the AVEVA Work Tasks Cache files and folders from *C:\Windows\Microsoft.NET\Framework64\v4.0.XXXXX\Temporary ASP.NET Files*.
18. Start the AVEVA Work Tasks related services.
19. To start IIS, run **Command Prompt** as an **Administrator**, and type **iisreset /start**.
20. Click **OK**.

Change Database Settings

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **AVEVA Work Tasks** Configuration Wizard appears.
2. Click **Next**.

   The Edit **AVEVA Work Tasks** Farm Settings screen appears.
3. Select **Database Settings**.
4. Click **Next**.

   The Connect to existing database screen appears.
5. Modify the **Authentication Type** or **Advanced Settings**.

   If SQL Server authentication is selected, you have to key in the user name and password.
6. Click **Test Connection** to ensure database connectivity.
7. Click **OK**.
8. Click **Next**.
9. Stop the following AVEVA Work Tasks services:
   a. Client Service
   b. Communication Service
   c. Task Scheduler
   d. Mobile Notification Service
   e. Workflow Engine
10. In an Enterprise Server system, ensure to stop the **Advance Server** service.
11. In a client computer, ensure to stop the **Client Service**.
12. In a client machine which is connected to an Enterprise Server, ensure to stop both the **Advance Server** service and **Client Service**.
13. Close all AVEVA Work Tasks related sites and applications.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

14. To stop IIS, run **Command Prompt** as an **Administrator**, and type **iisreset /stop**.
15. Delete the *WorkflowManagementTemp* folder from *C:\Windows\Temp*.
16. Delete the AVEVA Work Tasks Cache files and folders from *C:\Windows\Microsoft.NET\Framework64\v4.0.XXXXX\Temporary ASP.NET Files*.
17. Start the AVEVA Work Tasks related services.
18. To start IIS, run **Command Prompt** as an **Administrator**, and type **iisreset /start**.

The following table provides information on the various fields used in Connection String.

Field Name	Description
Authentication Type	This field is used to provide information on the process used to validate a user. AVEVA Work Tasks supports both Windows and SQL server authentication process. Example: Windows.
User ID	This field is used to enter the SQL user ID for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
Password	This field is used to enter the password for SQL Server type of authentication. For Windows Authentication, this field appears disabled.
Advanced Settings	This is an optional field. It is used to enter any additional connection information such as connection Timeout etc. Example: Connection Timeout = 20.

Configure Services

Configuring services involves the following:

- Two-way Email Notification
- Change Service Settings
- Client Service Monitoring Settings

Two-way Email Notification

Mail Server

The Mail Server settings identify the mail server to be used by the AVEVA Work Tasks Communication Service for receiving emails.

Group Name	Field Name	Description
Default	This option is selected to make Work Tasks communication server act as	

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Group Name	Field Name	Description
	a mail server and process all emails received.	
	Server IP	This field is used to provide the IP address of the Work Tasks communication server.
	Server Port	This field is used to enter the Port number used by the Work Tasks communication server. The default port is 25.
POP3	This option is selected to setup POP3 mail servers.	
	POP3 Host	This field is used to enter the IP address of the POP Mail Host.
	POP3 Time Interval	This field is used to set the time interval to check the mail server for emails. The format for this field is day.hour.minute.seconds. The default POP time is 0.00:00:15.
	POP3 Port	This field is used to enter the port number for receiving emails. The default POP Mail Port is 110.
	Is SSL Enabled?	This field is used to specify if a Secure Password Authentication is provided. Default value is false.
	POP3 Authentication - Basic	
	Add Users	The table is used to enter the **User Name** (email address) and **Password** of POP3 users. The Communication Engine makes use of this email address. To add another user, click **Add User**.
	POP3 Authentication - OAuth2 (See Configure POP3 Settings to get the values for the below fields)	
	User Account	This field is used to specify the email of the user.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Group Name	Field Name	Description
	OAuth2 Token URI	This field is used to specify the OAuth token URI.
	Authorization Code URI	This field is used to specify the authorization code URI.
	Client Id	This field is used to specify the client Id.
	Client Secret	This field is used to specify the client secret.
	Redirection URI	This field is used to specify the redirection URI.
	OAuth2 Scope	This field is used to specify the OAuth2 Scope.
	Token Refresh	This field is used to specify the time in Day.Hour:Minute.Second to refresh the token.

You can use the email server settings for sending and receiving emails.

Activities specific to sending emails comprises:

- Change mail server for sending email
- Setting "From email address" globally and at Repository level

Activities specific to receiving emails comprises:

- Changing mail channel/email receiver settings
- Changing mail server settings
- Changing POP3 password
- Adding another user to receive emails

If you choose **Default**, the Communication Service will act like a mail server. All mails received by the **Communication Service** will be forwarded to the specific Workflow for processing.

If POP3 is not enabled in mail server, then the **Default** option can be used for getting the emails.

Complete the following settings to configure the default options:

1. Select **Default** option.
2. Set **Listener IP Address** under **Communication Server** tab and **AVEVA Work Tasks Communication Service Server IP** under **Mail Server** tab with the IP address of the machine where the Communication Service is installed (that is, AVEVA Work Tasks Server).
3. Create an Email ID in exchange server.
4. Forward all mails coming to this email Id to a new email ID. The new mail ID should be in

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

user@[IPAddressOfWorkflowManagementMachine] format.

user: Name of the user.

IPAddressOfWorkflow ManagementMachine: This should be the one configured in step 2.

In an Enterprise Server environment, you must create a machine specific email account for all the engines. The mails should be forwarded to all the engines.

Change Service Settings

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Service Settings**.

   The **Enterprise Server Settings** can be selected if the Farm Server Installation is using the Enterprise Server Installation.
4. Click **Next**.

   The **Services/Global Settings** page appears.
5. Click the configuration icon against any of the services to modify a service.

Client Service Monitoring Settings

Client service monitoring settings can be changed using the file ServiceMonitorConfig.xml which is present in the installed AVEVA Work Tasks bin folder.

For more information related to the service monitoring, see Client Service Settings.

Configuration files can be changed for the following settings:

- **For enabling or disabling of service monitoring:**

  Service monitoring can be enabled or disabled using the configuration attribute ServiceMonitoringEnabled. For enabling monitoring, set ServiceMonitoringEnabled="true". For disabling monitoring, set ServiceMonitoringEnabled="false".

- **Whether to Send Alert emails or not:**

  By default, emails will be sent to the administrator whenever any service restart happens.

  If the administrator does not want to get alert emails, then set SendAlertEmailsEnabled="false".

- **Scheduled maintenance restart of all the AVEVA Work Tasks service can be done in the section MaintainanceRestart.**

  By default, this is disabled and this section is commented in the configuration file.

  <!--<MaintainanceRestart RestartAt="" WeeklyRestartDays="" MonthlyRestartMonths="" MonthlyRestartDates=""/>-->

  Uncomment the node MaintainanceRestart in the configuration file, and give the appropriate values. A more detailed explanation of the MaintainanceRestart can be found in Maintenance.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Changing the default CriticalMemory setting:**

  Default CriticalMemory setting can be changed in the DefaultConfig section which affects all AVEVA Work Tasks service.

  Uncomment the CriticalMemory node and set the MemoryLimitInMB value for it. A more detailed explanation of the Critical memory can be found in Critical Memory Limit section.

- **Changing the default WarningMemory setting:**

  Default WarningMemory setting can be changed in the DefaultConfig section which affects all services. Uncomment the WarningMemory node and set the values for it. A more detailed explanation of the Warning memory can be found in Warning Memory Limit section.

- **Changing the service settings for a specific service:**

  If the specific service does not want to use the default settings, then it can override it with its own settings. Suppose WFEngine wants to have separate settings, then replace

  `"<ServiceConfig Name="WFEngine" />"`

  with

  ```
  <ServiceConfig Name="WFEngine" MonitorService="true">
  <CriticalMemory MemoryLimitInMB=""/>
  <WarningMemory MemoryLimitInMB="" TimeRangeInMinutes="" RestartAt=""/>
  </ServiceConfig>
  ```

  and set the appropriate values.

- **Enabling or disabling the service monitoring for a specific service:**

  If a specific service should not be monitored, then make MonitorService="false" for that service. By default, it takes the value from the DefaultConfig section where it is enabled.

  After the change, when the client service is restarted, the specified service will not be monitored for maintenance restart, critical memory limit, warning memory limit, and service specific monitoring logics.

  Example: `<ServiceConfig Name="CommunicationService" MonitorService="false"/>`

Cryptography Settings

You can export and import the cryptography settings of one server to the other servers. For Load Balanced Server of Work Tasks, import the cryptographic settings with the same passphrase in all the servers. Using the same passphrase will help the easier recovery of the servers.

The following are the application pools and services that access AVEVA Work Tasks:

Applications Pools

- AVEVA Work Tasks Integrated
- Quickflow

Services

- NetworkService
- Current

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- System

If you use any other application pool or service to access Work Tasks, you need to add them to one of the following groups:

- Administrators
- aaAdministrators
- aaConfigTools

To Export/Import Cryptography Settings

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** screen appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Cryptography Settings** and click **Next**.
4. On the **Cryptography Settings** page, do the following:
   a. Enter the passphrase for the settings.

      Note: The passphrase should have characters between 8 and 20, consisting of lowercase, uppercase, numbers, and special characters. Use the same passphrase for all servers in Load Balanced Server of AVEVA Work Tasks.
   b. Click **Import** to import the cryptography settings file in the server.
   c. Click **Export** to export the settings. Keep the file safe.
5. Click **Finish**.

   The cryptography settings get imported to or exported from the server based on the action performed.

To Configure Cryptography Settings using Cryptography Configurator Tool

1. Navigate to **[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin** and run **CryptographyConfigurator.exe** to configure the cryptography settings.
2. On the **Cryptography Settings** page, do the following:
   a. Enter the passphrase for the settings.

      Note: The passphrase should have characters between 8 and 20, consisting of lowercase, uppercase, numbers, and special characters. Use the same passphrase for all servers in Load Balanced Server of AVEVA Work Tasks.
   b. Click **Remove** to clear the cryptography settings from the logged-in server.
   c. Click **Import** to import the cryptography settings file in the server.
   d. Click **Export** to export the cryptography settings. Keep the file safe.
3. Click **Finish**.

   The cryptography settings gets removed, imported to, or exported from the server based on the action performed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Migrate Passphrase using Migration Tool

You can migrate the passphrase and the configuration settings from one machine to another using Passphrase Migration Tool. Using the tool, you can migrate the following:

- Farm Database
- Datasources
- Central Configuration

To migrate passphrase using Passphrase Migration tool,

1. Navigate to **[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin\Tool** and run **AVEVAWorkTasksPassphraseMigrationWizard.exe** to migrate the existing configuration settings.
2. On the **Passphrase Migration Wizard**, do the following:
   a. In the **Existing Passphrase** box, enter the exiting passphrase configured earlier in a machine.
   b. In the **New Passphrase** box, enter the new passphrase for migrating existing configurations to a different machine.

   Note: The passphrase should have characters between 8 and 20, consisting of lowercase, uppercase, numbers, and special characters. Use the same passphrase for all servers in Enterprise Edition of AVEVA Work Tasks.
3. Click **Migrate**.

   The wizard shows the status of the configurations getting migrated. Also, if you check the database tables, you will see that the database columns has the new encrypted values.

   Note: Do not run this tool for migrating the settings when,
   - AVEVA Work Tasks is provisioned with external connectors (for example, AVEVA System Platform), as external connectors are not supported.
   - DataSource is created using Connection provider as the DataSource will not get migrated.

Administer Web Servers

You can change the settings of web sites, change or set the desired port number for Central Configuration and Enterprise Console. You can also add new Enterprise Console site and map/un-map a repository from an existing Enterprise Console.

Change Web Site Settings

The Farm Configuration Wizard allows you to install either the Central Configuration or the Enterprise Console on the AVEVA Work Tasks Client machines. This topic provides steps to configure these Web site settings:

- Changing IIS path for the Central Configuration Web site
- Changing IIS path for the Enterprise Console Web site

  Note: A Fully Qualified Domain Name will be displayed while configuring Enterprise Console (EC) and Central Config (CC) sites for the fresh installation of the product.
  When the product is upgraded, the existing EC and CC sites will remain the same.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The new EC and CC sites created after the upgrade will have the Fully Qualified Domain Name.

To change IIS path for the Central Configuration Web site

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Configuration Wizard screen appears.
2. Click **Next**.

   The Edit AVEVA Work Tasks Farm Settings window is displayed.
3. Select **Service settings**.

   The **Enterprise Server Settings** can be selected if the Farm Server Installation is using Enterprise Server Installation.
4. Click **Next**.

   The Services and Settings page appears.
5. Click against **Central Configuration** to edit AVEVA Work Tasks **Central Configuration settings**.

   The Central Configuration Settings page appears.
   - **http/https**: This option enables you to determine if you want a secured (https) or a non-secured site (http). It is recommended to use **https** for a secure connection.
   - **Create at**: This field allows you to provide the Internet Information Service (IIS) path where the new site is created.
   - **Port Number**: This field is used to provide a port number for the Central Configuration.
6. Click against **Create at** to change the path of the Central Configuration.

   A message indicating that if the path has been changed the Central Configuration will no longer be accessible appears.
7. Click **Yes**.

   The IIS structure screen is displayed. Select the desired IIS path for the Central Configuration.
8. Click **OK**.

   The selected URL appears in the Central Configuration Settings screen.
9. Click **OK**.

To change the IIS path for the Enterprise Console Web site

1. Follow the steps from 1 to 5 as indicated for the Central Configuration above.
2. Click against Enterprise Console option.

   The settings for the selected Enterprise Console appears.
3. Select the row in the Existing Enterprise Consoles table.
4. Click **Delete**.

   A prompt appears to confirm the deletion.
5. Click against **Create at**.

   The IIS structure screen appears.
6. Select the required IIS path for Enterprise Console.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. Click **OK**.
8. Click **Save**.
9. Click **OK**.

Note: Central Configuration and Enterprise Console cannot be accessed in Chrome browser when configured with unsecured access (http://) for Azure Active Directory and AVEVA Connect authentication types.

Set Desired Port Number for Central Configuration and Enterprise Console During Installation

Enterprise Console and **Central Configuration** are virtual applications [or Virtual Directories] of a Website.

Virtual Applications URL contains the port number of a Website. Virtual applications always inherit the port number of a Website.

If the Website's port number (for the http binding) is changed, then automatically, Virtual Applications inherit the new port number (changed port number).

Example:

Consider the name of the Website is "Telematics" and it has the following two Virtual Applications:

- CentralConfig
- EnterpriseConsole

URL of "Telematics" Website is *http://<computer123:8000/*

Port number of http/https binding of the Telematics Website is *8000*

Change the port number of the Telematics website to *8009*

URL of Telematics Website and its Virtual Applications- Before change

- Website URL: *http://computer123:8000/*
- Central Configuration URL: *http://computer123:8000/CentralConfig*
- Enterprise Console URL: *http://computer123:8000/EnterpriseConsole*

URL of Telematics Website and its Virtual Applications -After Change

- Website URL: *http://computer123:8009/*
- Central Configuration URL: *http://computer123:8009/CentralConfig*
- Enterprise Console URL: *http://computer123:8009/EnterpriseConsole*

There are three ways to create the Website (under which we are going to create Enterprise Console and Central Configuration) with the desired port number:

- Create the Website with the Desired Port Number Through Farm Configuration Wizard
- Use an Existing Website
- Change the Port Number of AVEVA Work Tasks Sites

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create Website with Desired Port Number Through Farm Configuration Wizard

Steps to create the Website with the desired port through Farm Configuration Wizard:

1. In Services/Global Settings page, select **Custom Settings**.

   By default, for custom settings, Website, Central Configuration, and Enterprise Console nodes will not be checked.
2. Select the above mentioned tree nodes, and then click **Next**.

   The Global Settings page appears.
3. Click **OK**.

   The Central Configuration Settings window appears.
4. Click the icon against **Create at** text box in the Central Configuration Settings page.

   The IIS structure screen appears.
5. Select the root node, and click **New Web Site**.

   The Website Settings page appears.
6. Type the **Website Name, Port number, Home directory**, and application pool value.
7. Click **OK**.

   Newly created site gets listed in the IIS Structure Page.
8. Select the site, and click **OK**.

   The selected URL with the desired port number appears in the Central Configuration Settings page.
9. Click **OK**.

   The Enterprise Console Settings page appears.
10. Click the icon against **Create at** text box.

    The IIS Structure gets displayed.
11. You can either create the new Website or select the newly created Website (for Central Config) and click **OK**.

    The selected URL with the desired port number appears in the Enterprise Console Settings page.
12. Click **Save** to add the Enterprise Console to the Existing Enterprise Consoles grid.
13. Click **OK**.

Change Port Number of AVEVA Work Tasks Sites

Steps to change the port number of AVEVA Work Tasks Sites

Note: These steps are specifically for Internet Information Services (IIS) 7.

1. In Services/Global Settings page, select **Custom Settings**.

   By default, for custom settings, Website, Central Configuration, and Enterprise Console nodes will not be checked.
2. Select the above mentioned tree nodes, and then click **Next**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Global Settings page appears.

3. Click **OK**.

   The Central Configuration Settings window appears.

4. Click **Start > Administrative Tools > Internet Information Services (IIS) Manager**.

   Alternatively, go to **Start > Run**. Type `inetmgr,` and then click **OK** to open Internet Information Services (IIS) Manager.

5. Find the site named "**AVEVAWorkTasksSites**".
6. Right-click **AVEVAWorkTasksSites** site and select **Edit Bindings**.
7. Select the http binding, and the click **Edit**.
8. Provide desirable port number and click **OK**.
9. Refresh the sites and check the updated port number is changed or not.
10. In the AVEVA Work Tasks Farm Configuration wizard, click the icon against **Create at** box in the Central Configuration Settings page.

    The IIS structure screen appears.

11. Select the desired IIS path for the Central Configuration (Select the site for which you have changed the port number- here AVEVA Work TasksSites).
12. Click **OK**.

    The selected URL with the desired port number appears in the Central Configuration Settings Page.

13. Click **OK**.

    The **Enterprise Console Settings** page appears. If you have changed the port number of the "AVEVA Work Tasks Sites" then automatically, the port number of Enterprise Console gets changed.

14. Click **Save**, so that Enterprise Console gets added to the Existing Enterprise Consoles grid.
15. Click **OK.**

Change Port Number of Central Configuration and Enterprise Console

Before you perform the settings mentioned in this section, ensure that the port number for the Central Configuration and the Enterprise Console are manually changed from the IIS manager.

For more information, see Setting Desired Port Number for Central Configuration and Enterprise Console During Installation and Change Port Number of AVEVA Work Tasks Sites.

To change the port number for Central Configuration

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Configuration Wizard appears.

2. Click **Next**.

   The Edit AVEVA Work Tasks Farm Settings screen is displayed.

3. Select **Service Settings**.
4. Click **Next**.

   The Service/Global Settings page appears.

5. Click against **Central Configuration**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Central Configuration Settings page appears.

- **http/https**: This option enables you to determine if you want a secured (https) or a non-secured site (http).
- **Create at**: This field allows you to provide the Internet Information Service (IIS) path where the new site is created.
- **Port Number**: This field is used to provide a port number for the Central Configuration.

6. Enter the new port number.

   The **Change Port number** button is enabled.
7. Click **Change Port Number.**
8. Click **OK**.

To change port number for Enterprise Console

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Configuration Wizard appears.
2. Click **Next**.

   The Edit AVEVA Work Tasks Farm Settings screen is displayed.
3. Select **Service Settings**.
4. Click **Next**.

   The Service/Global Settings page is displayed.
5. Click against Enterprise Console.

   The Enterprise Console Settings screen appears.
6. Select the required Enterprise Console entry in the Existing Enterprise Consoles table.
7. Enter a port number.

   The **Change Port** button is enabled.
8. Click **Change Port**.
9. Click **Save**.

Add a New Enterprise Console

The Farm Configuration Wizard allows you to create a new Enterprise Console.

Note: You can create as many Enterprise Consoles as you require, but you cannot create more than two Central Configuration sites.

To add a New Enterprise Console

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The Welcome to the AVEVA Work Tasks Farm Configuration Wizard appears.
2. Click **Next**.

   The Edit AVEVA Work Tasks Farm Settings window is displayed.
3. Select **Service settings**.
4. Click **Next**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Services/Global Settings page is displayed.

5. Click the icon against **Enterprise Console** option.

   The Enterprise Console Settings page appears.

6. In the **New Website** text box, enter a name for the Enterprise Console.

   For example, enter name as 'Telematics'.

7. To create the Enterprise Console in different port, click against **Create at** option, and then select the site with the required port.

   The default port is 8000.

   The IIS Structure page is displayed.

8. Select the path where you want to create an Enterprise Console.
9. Click **OK**.

   The selected path is displayed in the **Create at** text box.

   The mapped repositories are displayed in the Repositories.

10. Select the repository(s) to map to this created site.

    You can select multiple repositories by using Ctrl or Shift keys.

11. Select the required authentication type in the **Authentication Type** drop-down list.

    The available authentication types are Windows, Anonymous, AVEVA Connect, and Azure Active Directory.

    Note: The AVEVA Connect and Azure Active Directory authentication types are available only if these user providers are configured in the system.
    To configure AVEVA Connect user provider, see AVEVA Connect Settings.
    To configure Azure Active Directory user provider see Azure Active Directory Settings.

    See Authentication and Personalization topic to know more about Authentication Type.

12. Click **Save**.

    The created Enterprise Console is displayed in the Existing Enterprise Consoles section.

13. Click **OK**.
14. Click **Finish**.

Repeat the above steps to create one more Enterprise Console.

Note:
- If you want to update the authentication type to AVEVA Connect or Azure Active Directory of the already configured Enterprise Console, you need to reconfigure the AVEVA Connect and Azure Active Directory user providers for the respective Enterprise Console. For information on configuring user providers, see AVEVA Connect Settings and Azure Active Directory Settings.
- The Enterprise Console cannot be accessed in Chrome browser when configured with unsecured access (http://) for Azure Active Directory and AVEVA Connect authentication types.

Map or Un-map a Repository from an Existing Enterprise Console

To map a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Welcome to the AVEVA Work Tasks Configuration Wizard appears.

2. Click **Next**.
   The Edit AVEVA Work Tasks Farm Settings window is displayed.
3. Select **Service settings**.
4. Click **Next**.
   The Services/Global Settings page is displayed.
5. Click against Enterprise Console option.
   The Enterprise Console Settings page is displayed.
6. Select the existing Enterprise Console from Existing Enterprise Consoles section.
7. Select the required repository(s) to be mapped to the selected site.
   For example, select 'SPTest' and 'HCL-Roland-Repo' repositories using Ctrl key.
8. Click **Save**.
9. Click **OK**.
10. Click **Finish**.

To un-map a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.
   The Welcome to the AVEVA Work Tasks Configuration Wizard appears.
2. Click **Next**.
   The Edit AVEVA Work Tasks Farm Settings window is displayed.
3. Select **Service settings**.
4. Click **Next**.
   The Services/Global Settings page is displayed.
5. Click against Enterprise Console option.
   The Enterprise Console Settings page is displayed.
6. Select the existing Enterprise Console from **Existing Enterprise Consoles** section.
7. Deselect the repository(s) in the **Repositories** to un-map.
   This will un-map the selected repository from the selected Enterprise Console.
   You can select multiple repositories using the Ctrl or Shift keys.
8. Click **Save**.
9. Click **OK**.
10. Click **Finish**.

Edit Client Machines

You can configure the Client machine in the Farm Configuration Wizard.

To edit Client machines

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Welcome to the **AVEVA Work Tasks Farm** Configuration Wizard appears.

2. Click **Next**.

   The Edit **AVEVA Work Tasks** Farm Settings window appears.

3. Select **Service settings.**

   The **Enterprise Server settings** can be selected if the Farm Server Installation is using Enterprise Server Installation.

4. Click **Next**.

   The Services/Global Settings window appears.

5. Click against **Central Configuration** to edit the Central Configuration settings.

   The Central Configuration Settings page appears.

   - **http/https:** This option enables you to determine if you want a secured (https) or a non-secured site (http).
   - **Create at**: This field allows you to provide the Internet Information Service (IIS) path where the new site is created.
   - **Port Number**: This field is used to provide a port number for the Central Configuration.

6. Click against **Create at** to change the path of the Central Configuration. A message indicating that if the path is changed the Central Configuration will no longer be accessible appears.
7. Click **Yes**.

   The IIS structure screen is displayed.

8. Click **OK**.

To configure Enterprise Console on Client machines

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The Welcome to the **AVEVA Work Tasks** Configuration Wizard appears.

2. Click **Next**.

   The Edit **AVEVA Work Tasks** Farm Settings window appears.

3. Select **Service settings.**

   The **Enterprise Server settings** can be selected if the Farm Server Installation is using Enterprise Server Installation.

4. Click against Enterprise Console option.

   The settings for the selected Enterprise Console is displayed.

   To change port number of an Enterprise Console, select the required Enterprise Console entry in the Existing Enterprise Consoles table and modify the port number.

   - **New Website**: This field is used to provide the name of the Web site. Every Web site should have a unique name.
     - **http**: This option is to set the Web site as a non-secure site.
     - **https**: This option is to set the Web site as a secured site.
   - **Create at**: This field is used to provide the IIS path for installing the Web site.
   - **Internet Path**: This field is used to set the path for Internet users to access the Enterprise Console.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Intranet Path**: This field is used to set the path for Intranet users to access the Enterprise Console.
- **Repositories**: This field lists the Repositories mapped to the Enterprise Console.
- **Authentication Type**: This field is used to provide the authentication type required to access an Enterprise Console.
- **Existing Enterprise Consoles**: This field is used to list Enterprise Console already existing on the local machine.

5. Click **Save**.

Edit License Server Details

You can edit the license server details, such as license server name and port,

To edit License Server details

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click the **Farm Configuration Wizard.**

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** page appears.
3. In the **Edit AVEVA Work Tasks Farm Settings** window, select the **Configure License Server** check box.

   The **Configure License Server** dialog box appears.
4. In the **License Server Name** box, type the name of the AVEVA license server. If you have already configured the license server, then the name of configured license server is displayed.
5. In the **License Server Port** box, type the port number used by the AVEVA license server. Ensure that you enter a valid port number.
6. Click **Configure**.

   The confirmation messages appears in the **Configuration Messages** section, and the **Next** button is enabled.

   Note: AVEVA Work Tasks does not validate the license server details entered.
7. Click **Next**.

   The license server details are saved successfully.

Work with Enterprise Console List

The Enterprise Console List lists all the Enterprise Consoles mapped to repositories in the Central Configuration. This list enables easy access to the Enterprise Console. The columns in the list are Enterprise Console Name, URL and Mapped Repositories.

Note: EC plug-in is integrated with Email activity as part of the **Body** property, and the link gets embedded in the email body when you append it. When you click on the link in the email, it redirects you to the Enterprise Console List page.

To work with the Enterprise Console List

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **AVEVA Work Tasks** Central Configuration page appears.

2. Click **Enterprise Repository** from the Central Configuration home page.

   The Enterprise Console list appears.

3. Click on **Filter by repository name** to list Enterprise Consoles mapped to the selected repository.

Work with Central Configuration

The **Central Configuration** is a Web-based interface that you can use to define, create, and manage repositories for the workflows and execute them. The Central Configuration is setup on a computer with the Client installation. It can also be setup on a AVEVA Work Tasks Server.

The Farm Configuration Wizard is used to specify the location and settings for the Central Configuration. Only users who are part of the Administrators group and the AVEVA Work Tasks Admins group can logon to the Central Configuration.

Use the Central configuration to do the following:

- Create and manage data sources
- Create and manage repositories
- View the list of all the Enterprise Consoles mapped to repositories using the Enterprise Console List
- Manage Users
- View the list of User Providers mapped to a selected repository
- View the list of Enterprise Servers

Create a Repository DataSource

A **Repository DataSource** is used to store all repository related data. The **Repository DataSource** must be created on a machine where the **Central Configuration** site is installed.

Example

A TelematicsDataSource is created for TelematicsHR Repository.

A Repository contains Workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating the repository.

To create a Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**.

   The Central Configuration web page appears.

2. Click **DataSource Settings**.

   The Datasource List page appears.

3. Click **Add**.

   The Manage Datasource page appears.

4. Enter the required details.
   a. **Name**: Enter a valid data source name. Example: *TelematicsDataSource*.
   b. **Description**: Enter a description for the data source. Example: *This data source is used to store*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

connection details.

c. **Connection String**: Select this option to establish a connection with the database. The connection string contains fields that that defines the connection. The connection information is encrypted and stored.

Server Name

To use a database on the local machine, enter "localhost", ".", or the local computer's name for the **Server Name**.

Note: Server Name can be machine name/IP address/SQL Server AlwaysON/SQL Server instance name. Instead of entering "local host", it is preferable to give the local machine name.

If the **Database Type** is **SQL Server**, then specify an IPV6/IPV4 address for the **Server Name**.
To use **SQL Server AlwaysOn Availability Groups**, enter the SQL AlwaysOn Group name or the IP address, and the port number (separated by comma). For example: *localhost,1455*
To use SQLExpress instance as a database instance of the repository database, the **Server Name** must be *<server name>\SQLExpress*.

Authentication Type

Use this field to determine the process to validate a user who is trying to connect to the database. AVEVA Work Tasks uses Windows and SQL Server authentication process. SQL Server authentication requires a user ID and a password already defined in the SQL Server database. The **User ID** and **Password** field is disabled if Windows Authentication is selected. Example: *Windows*

Advanced Settings

Use this field to define advanced connection settings like providing the time (in seconds) that a connection string must wait to connect to the database server before terminating the connection.
Example: *Connection Timeout = 20*
or
For defining maximum connections that can be opened in a connection pool.
Example: *max pool size = 250*
Default connection pool is set as 250. We recommend to enter a higher value on heavily loaded environment.
While creating a data source for a repository, use the following custom settings to set properties for the connection pool.
encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;
encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.
enableltm: AVEVA Work Tasks internally implements the light weight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to the connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature, AVEVA Work Tasks controls the transaction from escalating to an expensive distributed transaction.
The above property can be used to turn off this feature. The default value is set as *True*.
connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time for retry for a new connection. The default value is 30 seconds. The value should be set in seconds.
maxconnectionretries: While opening a new connection, if the database server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is infinite.
sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

generating an error. The default value is set as 300.
These settings are case sensitive and should be mentioned only in lower case.

Database:

Specify or select a database that can be used as a repository database. This database stores all repository related data. The repository database is created if it does not exist on the database server instance.
Example: *TelematicsDB*

Enable Table Partitioning:

Select to enable database table partitioning. For more information, see Partitioning Database.

d. **Connection Provider**: Use this to provide connection related information from a Dynamic Link Library (dll). Connection related details are encrypted and stored in the library.

The selected dll must be copied to all the machines, all servers and all clients, that are connected to the current Farm.

Assembly Path

Click to browse and choose the required dll file.
To use custom properties (enableltm, connectionretrywaittime, maxconnectionretries, and sqlcommandtimeout) in the Assembly connection string, the database must be created before.
See *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Developer Guide* for creating the assembly. Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly.

5. Click **Test Connection** to ensure database connectivity.

   A message about successful test connection appears.

6. Click **OK**.

   A message about successful creation of the data source appears.The message also states that the required tables will be created in the data source after a repository is associated with the data source.

When a data source is saved, an entry is created for the data source in the **Farm Database**. When a first repository is created for the data source, AVEVA Work Tasks tables are added to the data source. While creating a data source, the archival database is created with the same database name suffixed by *_BI*. This database is used for archival in future use.

Note: After you configure the AVEVA Work Tasks, you must add the created **Farm Database** to the **SQL Server AlwaysOn Availability Groups**. Contact your Database Administrator to add the database to **SQL Server AlwaysOn Availability Groups**.

To view the data source created, click **DataSource Settings** link.

Manage Repository DataSource

You can use this option to modify an existing data source.

To modify Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The Central Configuration web page appears.

2. Click **DataSource Settings**.

   The Datasource List page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. In the **DataSource** column, click the required data source to be modified.

   The Manage DataSource page appears.
4. Modify the required information related to the following fields in the connection string:
   - Authentication Type
   - User ID and Password
   - Advanced Settings
5. Click **Test Connection** to verify the database connectivity.

   If **SQL Server** authentication is used, enter the password again after clicking **Test Connection** and before clicking **OK**.
6. Click **OK**.

Create a Repository

A **Repository** contains workflow definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the repository is also defined while creating a repository.

To create a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The Central Configuration page appears.
2. Click **Repository Settings**.

   The Repository List page appears.
3. Click **Add**.

   Note: If the number of repositories exceeds the permitted limit for the license, then the message "You have exceeded the maximum number of repositories" appears, and the repository creation is stopped. Click **Close**.

   The **Manage Repository** page appears.
4. Enter the required details.
5. Click **OK**.

   The **Config Administrator** page appears.
6. Click to configure an administrator.

   The User Lookup dialog box appears.
7. Select the provider from the provider drop-down list.
8. Enter the details to search the administrator for the repository that is created.

   Note: You can also search a user based on *ObjectSID* and *objectGUID* (applicable only for Active Directory provider).
9. Click **Search**.

   The user is listed under **Available Resources**.
10. Select and click the arrow to list the user under **Selected Resource**.
11. Click **Update**.

    The selected user is displayed in **User Lookup**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

12. Click **OK**.

    A message confirming the successful creation of an administrator appears.

13. Click **OK**.

    The **Repository List** page appears.

    For more information about configuring the system user, see Configure System User.

Manage Repository Window Fields

The following table explains the fields of the Manage Repository window:

Field Name	Description	Example
Repository DataSource	Lists all the data sources created in a Farm. The selected data source will be used to store data for the current repository.	TelematicsDatasource
Attach DataSource	Select Attach DataSource checkbox and then select a repository from the selected DataSource .Data associated with the selected repository will be copied to the new repository. The version of the installed product and destination DataSource should be the same as that of the DataSource from which the repository data is to be copied. For example, in an upgraded environment, from build Work Tasks version 1 to Work Tasks version 2, you cannot attach a DataSource created in build Work Tasks version 1 to build Work Tasks version 2. To match the build version of the installed product to the version of the DataSource: 1. If the DataSource was created in build Work Tasks version 1, then the entire build Work Tasks version 1 must be upgraded to Work Tasks version 2 by running the product installer in the upgrade mode.	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	2. All the Farm, DataSources, Repositories, and Lists of build Work Tasks version 1 must be upgraded to build Work Tasks version 2 by running the AVEVAWorkTasksUpgradeManager.exe located at the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder. 3. After upgrading all the AVEVA Work Tasks components to build Work Tasks version 2 build, that DataSource can be used in any machine that has same version by using this feature.	
Repository Name	Assign a name for the repository. Ensure that you do not use special characters other than **()-=[]{}|;,<>./!@$^`~** in the repository name.	Telematics Repository
Description	Provide a description for the repository.	This stores Telematics Employee Leave Details.
From mail	Email ID used by the email channel to send emails. We recommend to define the **From mail** and **Exception to mail** in a production environment. This will assist the process administrator to track any exception that might occur in the Workflow instance due to data mismatch or any configuration related issue. An email with exception details will be sent to the mail ID defined in **Exception to mail**. The From address of email channel mails continue to use the previously configured email ID, even after this property is edited and the repository is saved. To	Provide the valid email id that is specified in the **User Name** field while configuring **SMTP Settings** in the **Mail Server Settings** page. For more details, see Mail Server Setting.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	avoid this, restart the Workflow Engine Service and Communication Service.	
Exception to mail	Define an exception email ID. Exceptions generated by the Workflow Engine when an activity fails is sent to the email ID specified in **Exception to mail**.	Provide the valid email id.
Alert initiator of the Workflow	Determines if the status of a workflow should be sent as a notification mail to the initiator of the workflow.	A process with a rule to check the pending leave available for the employee exists. If this rule fails due to a data mismatch, the employee who submitted the leave application will receive an e-mail with details related to the exception as a notification.
Servers	Lists the servers available in the Farm. Repository will be available only to the mapped servers.	Work Tasks Server Name
Enterprise Consoles	Lists all the Enterprise Consoles for a selected AVEVA Work Tasks Server. Repository will be available only to the mapped Enterprise Console.	http://worktaskservername:8001/EnterpriseConsole
Enable Theme	Select this checkbox to enable the defined theme.	-
Theme Name	AVEVA Work Tasks supports themes that you can use to change the look and feel of the user interface. It provides **Default** and **Lite** themes.	-
Resource Provider	Lists the various default resource providers supported by AVEVA Work Tasks. Resource providers are also termed as User Provider. A User Provider is a data store containing information about the user. AVEVA Work Tasks also supports custom resource provider. For more information about	AVEVA Work Tasks Lists or Active Directory

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	Resource Provider, see User Providers. AVEVA Connect and Azure Active Directory resource providers are also available after you configure them. To configure these providers, see AVEVA Connect Settings and Azure Active Directory Settings. AVEVA Work Tasks allows secured LDAP configuration. The default port number for LDAPS is 636. You can map the repository in IDE and use the Secured LDAP. The logged-in Active Directory is displayed by default, and **Active Directory Datasource** expects **LDAP://** or **Fully Qualified Domain Name** as value.	
Custom Application Folder	Enter the names of the folders that contain the custom activities and custom resource providers. You must separate multiple folders with a semicolon.	-
Hide Repository	Select this option to make the repository unavailable to the Farm.	-
Source path	Enter the path to the Source Elements Folder.	-
Attachment Storage Path	Enter the path for storing attachments in the file system. **Note:** - The path specified should be an absolute path. - The default path is the **Attachments** folder in *C:\ProgramData\AVEVA Work Tasks*. - The default path is used when no path is specified. - Ensure appropriate rights are set to the folder, where attachments are stored, for the following identities:	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	-- The application pool identity for **Central Configuration** and **Enterprise Console(s)**. -- The account configured for the **Workflow Engine** service. - In case of mapped drives, ensure they are available to the identities. - In multiple-server environments, the path specified must be in a shared location.	
Auto Log Off Settings for Enterprise Console	Automatically logs out the logged in user after the designated period of inactivity. Enter the time in minutes. This field accepts only valid integer values between 0 to 360. 0 minute means the user will not get automatically logged off. The maximum time allowed is 360 minutes. In some cases there may be a slight time lag of few seconds which can be maximum of 10 seconds. The default value for automatic log off is 30 minutes. You use this field to define the log off time for all the providers of a particular repository. If you have administrator rights to a repository, you can also set separate log off time manually for each provider. For more information about manually setting the auto log off time at the provider level, see Configuring the Definition XML.	0 to 360 (minutes)
Auto Log Off Settings for AVEVA Work Tasks Pro	Request for user credentials for AVEVA Work Tasks Pro app after the set time is elapsed. Enter the time in minutes. This field accepts only valid integer values	0 to 2880 (in minutes)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	between 0 to 2880. 0 minute means the user will not get automatically logged off. The default value is 30 minutes. If AVEVA Work Tasks Pro app is offline, the Auto Log Off will not get active, and the user will not be automatically logged off as per the defined settings.	
Offline Support	Select this option to enable offline submission of items in AVEVA Work Tasks Pro app.	-
Refresh on Work Item submission	Select this option to enable Inbox refresh on Work Item submission in AVEVA Work Tasks Pro . Refresh option is enabled by default.	-
Timeout	Enter the time in seconds to set the timeout. If the value is set as 0, then there is no timeout. Default value is 60 seconds.	-
Label for Fill Form	Enter the label for Fill Form. The customized label will be displayed for the Fill Form in AVEVA Work Tasks Pro.	Work Tasks
Date Configuration Flag for Alternate Actors	Select the **Enable Date Settings** option to set the **From** date and **To** date as mandatory while creating alternate actors. For more information about creating alternate actors, see *Alternate Actors* topic in the *User Guide*.	-
Display Domain name with User Name	Select this checkbox to display the domain name along with the user name in the Enterprise Console and the List. The domain name will be displayed only:	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description	Example
	• If this checkbox is selected. • If the **Resource Provider** is selected as **Active Directory**.	

User Providers

AVEVA Work Tasks supports the following User Providers by default:

Active Directory

When you select **Active Directory** as the **Resource Provider**, details such as **Active Directory datasource** and the **Active Directory search base** are populated by default.

Authentication User, **Authentication Password**, **User Log-on ID**, and **User Log-on Display Name** are optional fields. Access to the active directory might not be available for all users. In such cases, provide the active directory authentication user name and the active directory authentication password with which you can connect to the Active Directory.

AVEVA Work Tasks considers *SAM-Account-Name* as **User Log-on ID** by default. If you want to use active directory attributes other than the *SAM-Account-Name* user, then enter the desired active directory attribute details in the **User Log-on ID** field and **User Log-on Display Name** field. **User Log-on Display Name** is the user friendly display name of the active directory attribute. This **User Log-on Display Name** is referred in the User Lookup dialog box against the **User Log-on ID**.

Steps

1. When you select **Active Directory** as the **Resource Provider**, then the **Use Token Service** field appears. In this field, select one of the following:
   - If you do not want to use AIM authentication, select **Without Token Service**.
   - If you want to use AIM authentication, then select a server from the list of alias server names.
2. Enter **Active Directory datasource**.
3. Enter **Active Directory search base**.
4. Enter **Domain Name**.

   The domain name is case-sensitive.
5. Enter **Authentication User**.
6. Enter **Authentication Password**.
7. Check **Is Cross Domain Role Provider** checkbox, if cross domain users and roles mapping is required.

   Example: Users *John* and *Jim* of domain *TelematicsU* and users *Mike* and *Bob* from domain *TelematicsAU* are added to role Team HR of domain *TelematicsMain*.
8. Enter **User Log-on ID** if you want to use another Active Directory attributes instead of *SAM-Account-Name*.

   We recommend to use attributes of type *string* as input for **User Log-on ID.** However, *ObjectSID* and *objectGUID* are also supported.

   After you set the **User Log-on ID**, you cannot change it later.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Enter the **User Log-on Display Name**.

   After you set the **User Log-on Display Name**, you cannot change it later.

   This is the default display name of the user account of the active directory user used. **User Log-on Display Name** is used in user lookups.

10. Click **Test LDAP**.

    A message about successful LDAP test appears.

Note:
- Some of the active directory attributes or properties are not indexed. If you use an active directory attribute that is not indexed, it might impact the log-on validation performance based on the active directory entries. We recommend to use indexed attributes as **User Log-on ID** or index the attributes.
- For an upgraded repository, you cannot change both **User Log-on ID** and **User Log-on Display Name**.
- You cannot localize **User Log-on ID** and **User log-on Display Name** fields.
- Disabled users will not be shown in user lookup search.

Using Forms

For Forms Resource Provider:

- If the selected repository data source is of type SQL Server, then Forms provider is of type SQL Server database.

1. In the **Resource Provider** drop-down list**,** select **Forms.**
2. In the **User Entity properties** field, enter XML data.

   Click **Sample Code** to get the XML Schema Template.

   Example

   ```
   <properties>
   <property name="userid" type="string" purpose="identifier" />
   <property name="username" type="string" purpose="name" />
   <property name="pwd" type="string" purpose="password" />
   </properties>
   ```

3. In the **User Provider Properties** field, enter the same XML Schema.

   There must be at least one identifier in the XML Schema. For example, UserID

4. In the **Datasource** field, enter a valid data source or specify a dll.

   The value for the data source can be specified as **Physical path of the dll** or **Physical path of the dll::param. Param** is not mandatory.

   See topic *Implementation of IDBConnectionString Interface* under *Data source Related Code Samples* sub-chapter in the *Build Business Process* section for creating the assembly.

   Ensure that there is only one class implementing *IDBConnectionString* interface within this assembly, whereas one class can return multiple connection strings.

   Examples of data source parameter:

   - c:\DLLStore\GetConnectionString.dll
   - c:\GetConnectionString.dll::Leave

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If there are two data sources, *Leave* and *Licensing*, and only one DLL is used to return a connection string based on the data source selected. The parameter can be specified as **c:\GetConnectionString.dll::Leave** or **c:\ GetConnectionString.dll::Licensing**. The connection string can be returned from the connection dll based on this parameter.

5. Enter a proper table name or view name which holds the User Name and Password for the Authentication purpose.

   Example: Passmast (the table which is available in IMS_NEW Database)

6. Click **Test SQL Connection**.

   A message about successful test connection appears.

Using ASP.NET Membership

This resource provider lists users defined in the asp.net membership database.

Lists

This resource provider lists users created using AVEVA Work Tasks Users Creation. If you are unable to determine a User Provider to create a repository, use AVEVA Work Tasks Lists.

To verify successful creation of the Repository:

1. Click **Enterprise Console** URL from the site path column.

   The repository created is listed.

   Add the created Enterprise Console link into "Add to Favorites" to easily navigate in future.

2. Click **Repository User Provider** link to logon to the repository.

   The Enterprise Console log-on page appears.

3. Enter the logon details, and then click **Login**.

   The Enterprise Console home page appears. This indicates successful creation of a Repository.

   All features can only be available for a user with administrative privileges.

Note: Consider you created the Repository and then deleted it. If you create the same Repository again immediately and try to log into it through Enterprise Console, the Provider does not list the Repository. You need to access IIS and recycle AVEVA Work Tasks from the Application Pool to get the Repository listed in the Provider list..

Configure Administrator User

You can configure a user account as an administrator. It is mandatory to configure a user account with administrative privileges after creating a AVEVA Work Tasks repository. The configured user account for a repository has full rights on the repository whereas the other users have limited rights. The configured user can control the security of the repository.

After a repository is created, the Central Configuration page appears:

By default, the administrator account for the selected User Provider will be populated in the drop down box.

Example

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If a Repository is mapped to the AVEVA Work Tasks List user provider then user [AVEVA Work Tasks List] will be populated, where user is an administrator in the List. Similarly, if the Repository is mapped to Active Directory user provider then Administrator [activedirectory] will be populated.

To configure a user as an Administrator

1. Click the user lookup icon.

   The User Lookup screen appears.
2. Enter the **User Name** in the **Search For** box.
3. Select **All Provider** from the **User-based Advanced Search-in Provider** field.
4. Click **Search**.

   The user is listed in the **Available Resources** box.

   If a resource is not found in the user provider, the message *No resources found* appears.
5. Select the user name from the **Available Resources** box and click the arrow mark to add the resource to the **Selected Resource** box.

   To add multiple users, select the multiple users from the **Available Resources** box, and then click the appropriate arrow mark.

   To remove a resource from the **Selected Resources** box, select the resource, and then click the appropriate arrow mark.

   To remove multiple resources from the **Selected Resources** box, select the multiple resources, and then click the appropriate arrow mark.
6. Select the user from the **Selected Resources** box and click **Update**.

   The user account appears in the **User Lookup** box.

   This user account will be configured as an administrator.
7. Click **OK** to configure the selected user as an admin user.

Configure System User

A system user is used by a File Watcher event for alerting workflows. Events for File Watcher are received without user context and uses the system user context while alerting workflows. The system user profile requires administrator level rights and can be re-configured at any time.

The File Watcher workflows fails if the system user configurations are invalid.

To configure a user as a System user

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click Central Configuration.

   The Central Configuration page appears.
2. In the **Central Configuration** page, click **Repository Settings**.

   The Repository List page appears.
3. On the **System User** column, click **Configure Now** against the required repository name.

   The Configure System User page appears.
4. Click the user lookup icon.

   The User Lookup dialog box appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Enter the user name in the **Search For** box.
6. From the drop down box, select a proper classification for the user name.
7. Select the required provider from the drop-down field, and enter the user name in the **Search For** box.
8. From the drop down box, select a proper classification for the user name.
9. Click **Search For**.

   The user is listed in the **Available Resources** box.

   A message about unable to find resources appears if the resource is not found in the user provider.
10. Select the user name from the **Available Resources** box and add the resource to the **Selected Resource** box.

    You can add or delete one or more users at the same time.
11. Select the user from the **Selected Resource** box, and then click **Update**.

    The user account appears in the **User Lookup** box. This user account is configured as an administrator.
12. Click **OK** to configure the selected user as a system user.

Configure Exception Mail Templates

You can configure the exception mail templates at the repository level.

There are two options to configure exception emails.

- You can configure exception mail templates for two groups, Group A and Group B, both at the workflow level. Do one of the following:
  - Select an existing default or sample template
  - Use an editor

    Based on the exception mail configuration, the members of each group will receive mails. There must be a notification to the users about the exception occurred in an activity. These users and the template for the exception mails can be configured in the **Configuration for Exception Mail Template** settings in the **Central Configuration**.
- You can configure exception mail templates for two groups, Group A and Group B, both at the repository level.

  It is not mandatory to set both user groups A and B.

  For example, you can configure *administrator* users in Group A and *normal* users can be configured in Group B.

  Do one of the following:
  - Select an existing default or sample template
  - Use an editor

    Based on the exception mail configuration, the members of each group will receive mails.

    Place the template file in the **..\WorkflowElements\ExceptionTemplate** folder. In the template file, place the placeholders in the html file.

To configure Exception Mail Template

1. Select the repository from the **Repository Name** drop-down list.
   a. You can configure users in the User Lookup dialog box or the Role Lookup dialog box.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

By default, the User is selected.

i. Click the **User Lookup** icon.
ii. The User Lookup dialog box appears.
iii. Select the required provider from the drop-down field.
iv. Enter the name of the user in the **Search For** field.
v. Select the **Display Name** from the drop-down list.
vi. Click **Search**.
vii. All the users are listed in the **Available Resources** section.
viii. Select the required user and click the right arrow to display the users in the **Selected Resource** section.
ix. Click **Update**.

b. To configure a user through **Role Lookup**, select the Role radio button.
   i. Enter the name of the required user, and click **Search**.
      All the users are listed in the **Available Resources** section.
   ii. Select the required user and click the right arrow to display the users in the **Selected Resource** section.
   iii. Click **Update**.

The user's name (either User or Role) appears in the **Exception Mail Group A** field.

2. Select the template from the **Template File** drop down list.
   You can select either a **Default** template or a **Sample** template.
3. Follow the same steps to select a User or Role and a template for the **Exception Mail Group B**.
4. Click **OK**.

Change Pool Size of Repository DataSource Connection Settings

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The Central Configuration page appears.
2. Click **DataSource Settings**.
   The DataSource List page appears.
3. Click the required data source to be modified.
   The Manage DataSource page appears.
4. Enter the pool size settings in the **Advanced Settings** field of the **Connection String** option.
5. Click **Test Connection**.
6. Click **OK**.
   The pool size of the **Repository DataSource** connection is changed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Change Password of SQL Server User Credentials Used to Connect to Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The Central Configuration page appears.
2. Click **DataSource Settings**.
   The DataSource List page appears.
3. Click the required data source to be modified.
   The Manage DataSource page appears.
4. Modify the **User ID** and **Password** field in the **Connection String** option.
5. Click **OK**.
   The password of SQL Server user credentials are changed.
6. Stop the following AVEVA Work Tasks services:
   a. Client Service
   b. Workflow Engine
   c. Task Scheduler
   d. Communication Service
   e. Mobile Notification Service
7. In an Enterprise Server system, ensure to stop the **Advance Server Service**.
8. In a client computer, ensure to stop the **Client service**.
9. In a client machine which is connected to an Enterprise Server, ensure to stop the **Advance Server Service** and **Client Service**.
10. Close all the AVEVA Work Tasks related sites and applications.
11. To stop IIS, run **Command Prompt** as an **Administrator**, and type **iisreset /stop**.
12. Delete the *WorkflowManagementTemp* folder from *C:\Windows\Temp*.
13. Delete the AVEVA Work Tasks Cache files and folders from *C:\Windows\Microsoft.NET\Framework64\v4.0.XXXXX\Temporary ASP.NET Files*.
14. Start the AVEVA Work Tasks related services.
15. To start IIS, run **Command Prompt** as an **Administrator**, and type **iisreset /start**.
16. Click **OK**.

Map or Un-map a Repository from Servers and Enterprise Consoles

To map a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The Central Configuration page appears.
2. Click **Repository Settings**.
   The Repository List page appears.
3. Click the repository that you want to map.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. In the **Servers Mapped** section, do the following:
   a. Select the server to be mapped from the **Servers** list.
   b. Select an Enterprise Console to be mapped from the **Enterprise Consoles** list.
5. Click **OK**.
   The selected repository is mapped to the select server.

To un-map a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The Central Configuration page appears.
2. Click **Repository Settings**.
   The Repository List page appears.
3. Click the repository for which you want to remove the mapping.
4. In the **Servers Mapped** section, do the following:
   a. Press **Ctrl** and click the required server in the **Servers** list.
   b. Press **Ctrl** and click the required Enterprise Console to be unmapped from the **Enterprise Consoles** list.

Hide or Unhide Repository

Use the **Hide Repository** to make the Repository unavailable to the Farm. You can make the repository available again by selecting the **Unhide** option.

To hide a Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The Central Configuration page appears.
2. Click **Repository Settings**.
   The Repository List page appears.
3. Select the check box against the repository that you want to hide.
4. On the **Select Actions** drop down list, select **Hide**.
5. Click **Go** to update the changes.
   The selected repository is hidden.

To unhide a hidden repository:

1. Go to the Repository List page.
2. Select the check box against the repository that you want to show.
3. Select **Unhide** from the **Select Actions** drop down list.
4. Click **Go**.
   The hidden repository appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Delete Repository

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The Central Configuration page appears.
2. Click **Repository Settings.**
   The Repository List page appears.
3. Select the box against the repository that you want to delete.
4. Click **Delete Repository**.
   A message confirming the deletion appears.
5. Click **OK**.
   The selected repository is deleted.

Delete Repository DataSource

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The Central Configuration page appears.
2. Click **DataSource Settings**.
   The DataSource List page appears.
3. Select the box against the data source that you want to delete.
4. Click **Delete**.
   A confirmation message about deleting the data source appears.
5. Click **OK**.

If the selected data sources are mapped to one or more repositories, you must delete all the repositories associated with the selected data sources before deleting the data sources.

Diagnostics

The **Diagnostics** feature allows you to run the diagnostics tests and determine if AVEVA Work Tasks is functioning properly.

You can select the services and run the diagnostics test to know the status of the selected services. The status of the selected services is displayed in the Logs section of the Diagnostics page. You can also filter the logs based on Pass or Fail status of the services, export the Logs, and maximize the Logs view.

You can select the Log levels to display the relevant messages. The following Log levels are available:

Level 1: This level shows minimal information in the Logs section. The Logs section displays information related to the start and end of a workflow. Error messages are also shown in this mode.

Level 2: This level provides detailed information of background activities. It will show all the messages logged from the actions.

Level 3: This level displays messages indicating whether an action is executed.

The following services can be diagnosed to check their availability in AVEVA Work Tasks:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Mobile Notification Service**: Tests whether Mobile Notification Service is running.
- **Azure Service Bus Relay**: Tests whether Azure Service Bus Relay is configured and the response is received from Http Web Request.
- **Message Broker (amqp/amqps)**: Tests whether the repository is provisioned and the Message Broker (amqp/amqps) exist in the repository.
- **Quickflow**: Tests whether Quickflow is deployed, AMQP protocol is validated on the Message Broker, and Quickflow Host is connected.
- **Service Communication**: Tests whether Enterprise Server service and all the AVEVA Work Tasks services are running.

To run the diagnostics test

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, point to **Work Tasks**, and then **Central Configuration**.

   The AVEVA Work Tasks **Central Configuration** page appears.
2. Click **Diagnostics Settings**.

   The **Diagnostics** page appears.
3. Select the check box against the items for which you want to run the diagnostics tests and then click **Test** button.

   After the diagnostic test is complete, the status of the selected items appears in the **Logs** section of the **Diagnostics** page.

To reset IIS

After you change the log level, click the **Reset IIS** button in the **Diagnostics** page to reset IIS and update the log level. On IIS reset all the AVEVA Work Tasks sites will get restarted.

To restart all services

After you change the log level, click **Restart Service** button in the **Diagnostics** page to restart all the AVEVA Work Tasks services and update the log level.

Note: If you modify the log level, ensure to restart services and reset IIS to update the log level.

Mail Server Setting

The **Mail Server Settings** web page allows you to configure SMTP server and POP3 server to send and receive emails from the workflow.

Note: Configuring the mail host with a valid username and password is mandatory. Sending emails anonymously is not supported, so provide the valid username and password for Basic authentication while configuring SMTP server and POP3 server.

SMTP Settings

The SMTP settings specified here are the default settings used by AVEVA Work Tasks. These settings are used to send notifications in the form of emails. The email settings can also be specified at the Repository level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Exchange Server configured On Premises	Office 365 - Exchange Online
Host	Enter the name or IP address of the mail host.	Enter the name for the Exchange Online server, as **smtp.office365.com**.
Port	Enter the port number for sending e-mails. The default port is 25.	Enter the port number for sending e-mails. Ensure that you specify the value for port number as **587**.
Email Timeout	Enter timeout for e-mails in seconds. The default timeout is 10000.	Enter timeout for e-mails in seconds. The default timeout is 10000.
Is SSL Enabled?	This field is used to define if the e-mail has to be secured by SSL before it is sent. The default value is **False**.	This field is used to define if the e-mail has to be secured by SSL before it is sent. Ensure that the value is always set to **True**.
Authentication	This field is used to select the authentication type - **Basic** or **OAuth2**.	This field is used to select the authentication type - **Basic** or **OAuth2**.

For **Basic Authentication**, provide details in the below fields:

Note: Sending emails anonymously is not supported, so provide the valid username (valid email address) and password for **Basic** authentication.

Field Name	Description
User Name	Enter the user name for logging in to the mail server. Ensure that the user account is an Office 365 account with "administrator" privileges.
Password	Enter the password.
Exception From Email	Enter the e-mail Id that will be used to send exceptions which occurred during workflow execution. **Note**: Provide the valid email id that is specified for the **User Name** field.
Exception To Email	Enter the e-mail Id to which the emails detailing exceptions in the workflow execution must be sent.

For **OAuth2 Authentication** (See Configure OAuth2 to Access Google Account to get the values for the below fields)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description
OAuth2 Token URI	This field is used to specify the OAuth token URI.
Authorization Code URI	This field is used to specify the authorization code URI.
Client Id	This field is used to specify the client Id.
Client Secret	This field is used to specify the client secret.
Redirection URI	This field is used to specify the redirection URI.
OAuth2 Scope	This field is used to specify the OAuth2 Scope.
Token Refresh	This field is used to specify the time in Day.Hour:Minute.Second to refresh the token.

POP3 Settings

The Mail Server settings identify the mail server to be used by the Work Tasks communication service for receiving e-mails.

Note: Sending emails anonymously is not supported, so provide the valid username (valid email address) and password for **Basic** authentication.

Server	**Field Name**	**Description**
Default	This option is selected to make Work Tasks communication server act as a mail server and process all emails received.	
	Server IP	This field is used to provide the IP address of the Work Tasks communication server.
	Server Port	This field is used to enter the Port number used by the Work Tasks communication server. The default port is 25.
POP3	This option is selected to setup POP3 mail servers.	
	POP3 Host	This field is used to enter the IP address of the POP Mail Host.
	POP3 Time Interval	This field is used to set the time interval to check the mail server for emails. The format for this field is day.hour.minute.seconds. The default POP time is 0.00:00:15.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Server	Field Name	Description
	POP3 Port	This field is used to enter the port number for receiving emails. The default POP Mail Port is 110.
	Is SSL Enabled?	This field is used to specify if a Secure Password Authentication is provided. Default value is false.
	For Basic Authentication	
	Add Users	The table is used to enter the **User Name** (valid email address) and **Password** of POP3 users. The Communication Engine makes use of this email address. To add another user, click **Add User**.
	For OAuth2 Authentication (See Configure OAuth2 to Access Google Account to get the values for the below fields)	
	User Account	This field is used to specify the email of the user.
	OAuth2 Token URI	This field is used to specify the OAuth token URI.
	Authorization Code URI	This field is used to specify the authorization code URI.
	Client Id	This field is used to specify the client Id.
	Client Secret	This field is used to specify the client secret.
	Redirection URI	This field is used to specify the redirection URI.
	OAuth2 Scope	This field is used to specify the OAuth2 Scope.
	Token Refresh	This field is used to specify the time in Day.Hour:Minute.Second to refresh the token.

In case of **Default**, the Work Tasks communication server will act as a mail server and process all emails received. The user should configure the **Server IP** and **Server Port** of the mail server.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If POP3 is not enabled in a mail server, then **Default** option can be used for getting the emails.

Note: In an Enterprise Server environment, you must create a machine specific email account for all the engines. The mails should be forwarded to all the engines.

Configuring SMTP Settings

The SMTP Server settings identify the mail server to be used by the Work Tasks communication service for sending e-mails. You can configure the SMTP server using Central Configuration.

To configure SMTP Server with OAuth2 authentication, you need to provide the following OAuth2 authorization details:

- User Account
- OAuth2 Token URI
- Authorization Code URI
- Client Id
- Client Secret
- Redirection URI
- OAuth2 Scope
- Token Refresh

An example of Google account is considered to explain how to get the OAuth2 authorization details. For more information on configuring OAuth2 details, see Configure OAuth2 to Access Google Account.

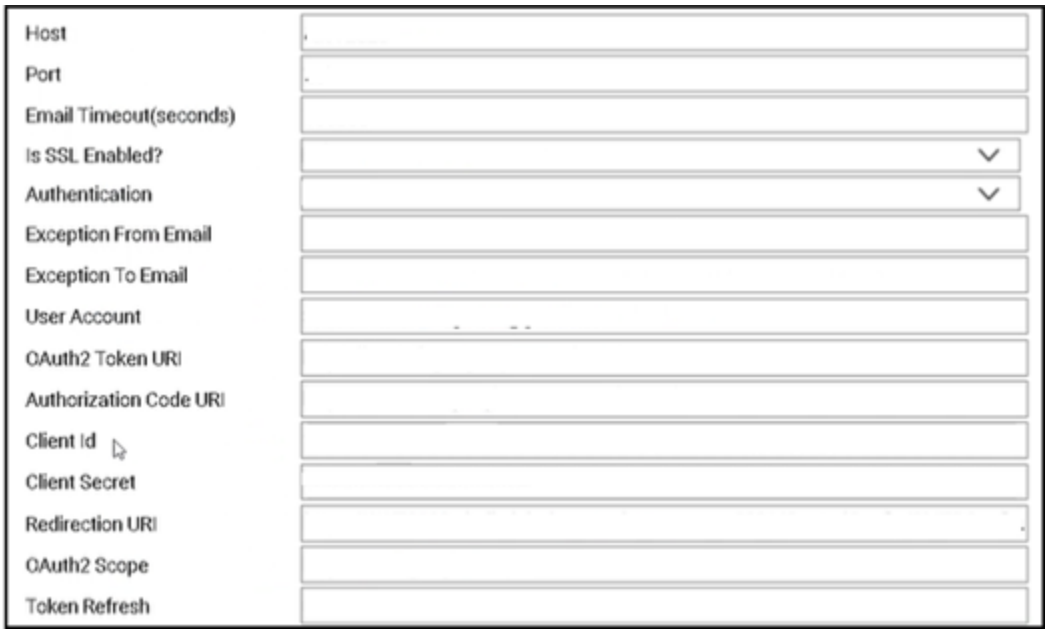
To configure SMTP Server

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The Central Configuration web page appears.
2. Click **Mail Server Settings**.
3. Select **SMTP Settings** tab.
   The **Manage SMTP Settings** page appears.
4. In the **Host** box, enter the SMTP server name.
5. In the **Port** box, enter the SMTP server port.
6. In the **Email Timeout** box, enter the timeout for emails in seconds.
7. For **Is SSL Enabled?** field, select **True** for secured password authentication.
8. For **Authentication** field, select either **Basic** or **Oauth2**.
   a. For **Basic** authentication, provide the **Username** and **Password** for accessing the SMTP mail server. Also, provide e-mail address in **Exception From Email** and **Exception To Email**. Click **Save** button to save the configuration.

      Note: Configuring the mail host with a valid username and password is mandatory. Sending emails anonymously is not supported, so provide the valid username and password.

   b. For **Oauth2** authentication, provide details in the fields shown in the screen below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



See **step 8** mentioned in Configure OAuth2 to Access Google Account to get the values for **OAuth2 Token URI**, **Authorization Code URI**, **Client Id**, **Client Secret**, and **Redirect URI** fields.

Provide the URL *http://<MachineName>:8000/CentralConfig/SMTPConfig.aspx* in **Redirect URI** field to configure SMTP server.

c. Click **Get Access Token** to get the access token. You are navigated to your mail account. For example, Google account is considered here.

d. Click **Continue** for verification and to provide access to your Google account to get the access token.

Configure POP3 Settings

You can access the Central Configuration to configure Mail server with POP3 and fetch emails from the configured POP3 server.

To configure Mail Server with POP3 and OAuth2 authentication, you need to provide the following OAuth2 authorization details:

- User Account
- OAuth2 Token URI
- Authorization Code URI
- Client Id
- Client Secret
- Redirection URI
- OAuth2 Scope
- Token Refresh

An example of Google account is considered to explain how to get the OAuth2 authorization details. For more information on configuring OAuth2 details, see Configure OAuth2 to Access Google Account.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To configure POP3 Mail Server

1. On the **Windows Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The Central Configuration web page appears.
2. Click **Mail Server Settings**.
3. Select **POP3 Settings** tab.

   The **Manage Mail Server Settings** page appears.
4. From the **Mail Server Configuration** drop-down, select either
   - **Default**: Provide **Server IP**, **Server Port** and then click **Save** button to save the changes.
   - **POP3**: Provide **POP3 Host**, **POP3 Port**, and **POP3 Time Interval**.

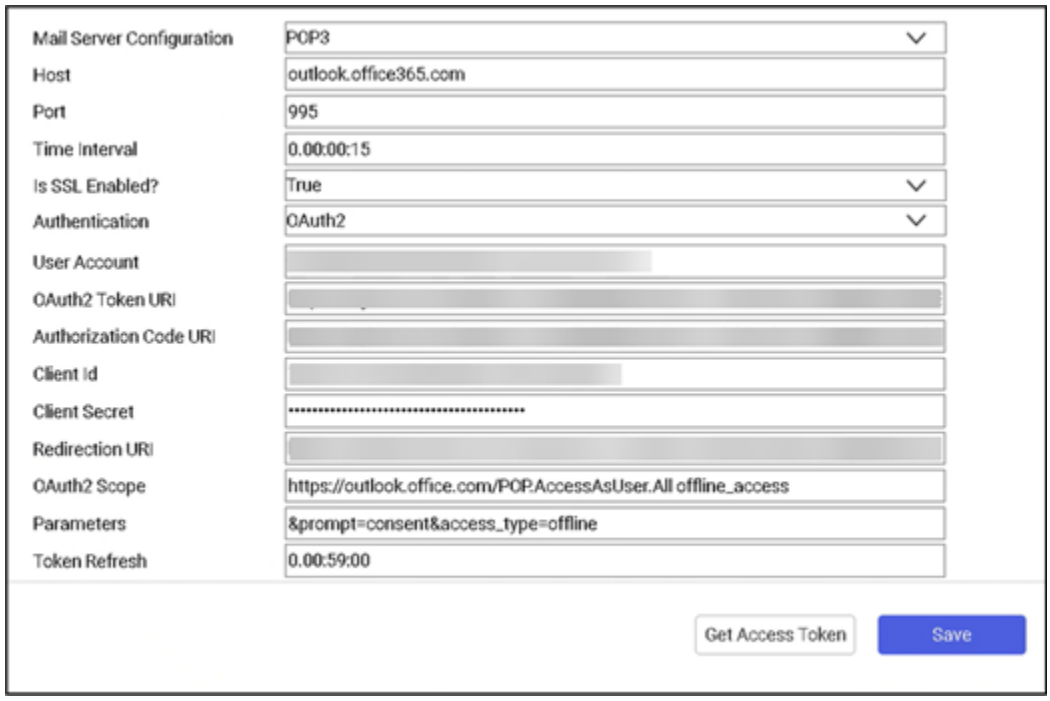
     There are two types of authentication available for POP3:

     Note: It is recommended to use **OAuth2** authentication. Support for **Basic** authentication varies by provider, so please review the provider's documentation before proceeding with its use.

   a. For **Basic** authentication, click **Add Users** to add **Username** and **Password**. Click **Save** button to save the configuration.

      Note: Configuring the mail host with a valid username and password is mandatory. Sending emails anonymously is not supported, so provide the valid username and password..

   b. For **Oauth2** authentication, provide details in the fields shown in the screen below:



   See **step 8** mentioned in Configure OAuth2 to Access Google Account to get the values for **OAuth2 Token URI**, **Authorization Code URI**, **Client Id**, **Client Secret**, and **Redirect URI** fields.

   Provide the URL *http://<MachineName>:8000/CentralConfig/ConfigMailServer.aspx* in **Redirect URI** field to configure POP3 server.

   c. Click **Get Access Token** to get the access token. You are navigated to your mail account. For example, Google account is considered here.

   d. Click **Continue** for verification and to provide access to your Google account to get the access token.

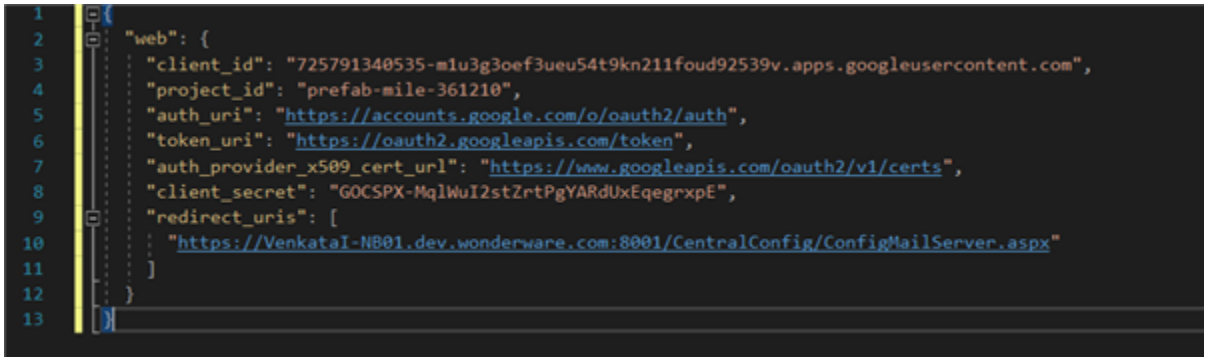
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure OAuth2 to Access Google Account

You can use OAuth2 authentication to access mail servers (SMTP and POP3) to send and receive the emails from the configured SMTP and POP3 servers using Google account.

To configure OAuth2 details to access Google account

1. Browse the link https://console.cloud.google.com/apis/credentials and create a New Project.
2. Select **OAuth consent screen** option from the left pane and provide **App information**, **Domain**, and **Developer contact information**.
3. Provide **Authorised domains** and **Developer contact information**.
4. To manually add or remove scopes, access link https://mail.google.com/. Click **ADD TO TABLE** and then click **UPDATE**.
5. To add users, click **ADD USERS**, provide user email id, and then click **SAVE AND CONTINUE**.
6. Select **Credentials** option from the left pane and update **Authorized redirect URIs**.
7. Download OAuth Client JSON file.
8. Open downloaded JSON file.



9. Update **OAuth Token URI**, **Authorization Code URI**, **Client Id**, **Client Secret**, **Redirection URI** from the downloaded JSON file.

   OAuth Scope is as configured in https://mail.google.com/
10. Refresh Interval as **0.01:00:00**.
11. Click **Save** button to save the configuration.

Configure OAuth2 to Access O365 Account

You can use OAuth2 authentication to access mail servers (SMTP and POP3) to send and receive the emails from the configured SMTP and POP3 servers using O365 account.

Following are the steps to configure Oauth2 to access O365 account:

1. Create an Azure AD Tenant Account and purchase services from Microsoft 365

1. Access Microsoft Azure portal to create Azure AD Tenant account.
2. Purchase services from Microsoft 365.
3. Navigate to **Users > Active users** to check if you are listed under **Active users**.
4. Make sure the active user has the Mailbox permission. Also, the licences and apps permissions are activated

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

for the user.

5. Select **Mail** tab for the selected active user, select **Manage Email apps**, and then select the **Authenticated SMTP** option.

2. Register Exchange Online in the Azure Portal

For more information, refer to https://documentation.commvault.com/v11/essential/93448_registering_exchange_online_with_azure.html

1. Log on to the Azure portal (https://portal.azure.com/) using your global administrator account.
2. Go to Azure Active Directory.
3. In the navigation pane, click **App registrations**.

   The **App registrations** page appears.
4. Click **New registration**.

   The **Register an application** screen appears.
5. In the **Name** box, type a name for the app.
6. Under **Supported account** types, select **Accounts in this organizational directory only** (tenant_prefix - Single tenant).
7. To verify the status of the app and to authorize the app from the Command Center, under **Redirect URI**, enter the Command Center URL.
8. For example, enter https://host_name.domainname.com/commandcenter.
9. Click **Register**.
10. Copy and paste the following values in a document that you can access later:
    - Application ID
    - Directory ID

    Note: You will enter these values in the Commvault software when you complete the Office 365 guided setup.

3. Request and Grant Permissions for Azure APIs for Azure Apps

1. In the navigation pane, click **API permissions**.
2. Click **Add a permission**.
3. Click **Microsoft Graph** and complete the following steps:
   a. Click **Application permissions**.
   b. Select the following permissions:
      a. Group.Read.All
      b. SMTP.Send
      c. User.Read
   c. Click **Add permissions**.
4. On the app **API permissions** page, click **Add a permission**.
5. Click **APIs my organization uses** and complete the following steps:
   a. On the search bar, type **Office 365 Exchange Online**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

b. Select **Office 365 Exchange Online**, and then click **Application permissions**.
c. Select the following permissions:
    a. Group.Read.All
    b. SMTP.Send
    c. User.Read
d. Click **Add permissions**.

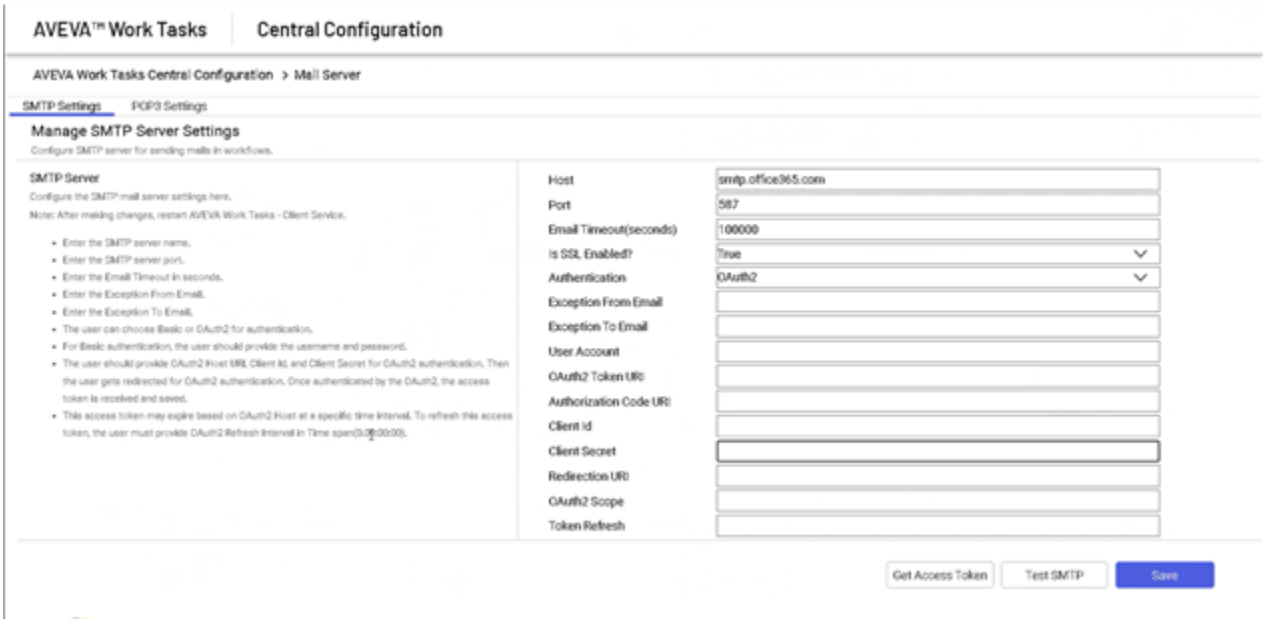
6. On the app **API permissions** page, click **Grant admin consent** for tenant_name.

4. Create a Client Secret

1. In the navigation pane, click **Certificates & secrets**.
   The **Certificates & secrets** page appears.
2. Click **New client secret**.
   The **Add a client secret** dialog box appears.
3. Enter a description, and then select the maximum value.
4. Click **Add**.
5. Copy and paste the client secret value in a document that you can access later.

5. Register Oauth2 in the CentralConfig Site

Access the Central Configuration site and navigate to **Mail Server** > **SMTP Settings** to register Oauth2..



View the Enterprise Servers List

You can view the list of servers that are running in Enterprise Server mode and the CPU performance for these servers.

The Enterprise Servers List node is available only when the Enterprise Server license is activated for the product.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The Central Configuration page appears.
2. Click **Enterprise Servers List**.

   The **View Enterprise Servers List** page appears.

   The list displays the following details:
   - **Host Name**: Displays the server name which is configured to work in the Enterprise Server mode.
   - **CPU Utilization (%)**: Displays the CPU utilization of the Enterprise Server in percentage.
   - **CPU Level**: Displays the CPU Level of the Enterprise Server. The level of CPU depends on the CPU usage. It can show the following values:
     - **Idle**: When the CPU utilization ranges from 0 to10.
     - **Normal**: When the CPU utilization ranges from 10 to 50.
     - **Overloaded**: When the CPU utilization is above 50.
     - **Down**: When the Enterprise Server is unable to communicate within the Heartbeat timeout.
   - **Registered Date Time**: Displays the date and time when the server was registered to work in the Enterprise Server mode.
   - **Last Updated Date Time**: Displays the last date and time when the request was received from the server.
   - **Is Server**: Displays whether the machine is a server.
   - **Service List**: Displays the list of services running on the server in the Enterprise Server mode. It can show the following list of services:
     - Task Scheduler Service as TS
     - Communication Service as CS
     - Workflow Engine Service as WS
     - Mobile Notification Service as MN

- **Engine Authority Server**: Displays which server acts as an Authority server for Workflow Engine.
- **Engine Least Used Server**: Displays which server acts as the Least Used server for Workflow Engine.

Configuring and Registering Token Service

You can manage AIM token configuration using:

- Configurator tool
- AVEVA Work Tasks Central Configuration

For Enterprise Console logout to function,

1. Configure SMS on a Web Server.

   To configure SMS, select the option '**This machine is the System Management Server**' in the **Configurator** window.
2. Token service should point to the machine where SMS is configured.

   To do this, navigate to **Central Configuration** > **Token Service Settings** page and for the *servername* in the **Token Service URL** field, provide the machine name where SMS is configured.

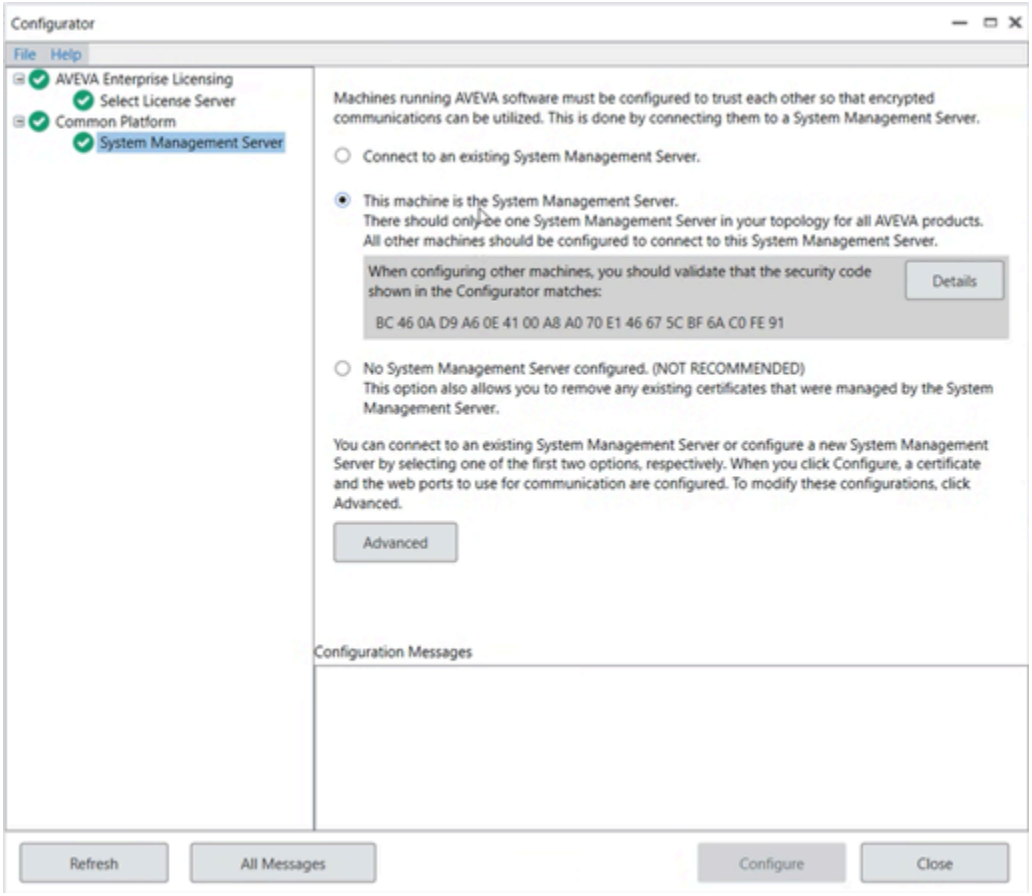
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Servers mapped should only point to the Enterprise Console where SMS is configured.

   To do this, navigate to **Central Configuration** > **Repository Settings** > **Servers Mapped** > **Enterprise Consoles** and select the Enterprise Console where SMS is configured.

Token Service Configuration using Configurator Tool

The **Configurator** window lists the System Management Servers as illustrated below:



- Click the **Configure** button to configure certificate manager.

Token Service Configuration and Registration using Central Configuration

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then right-click **Central Configuration**, and select **Run as Administrator**.

   The **Central Configuration** web page appears.

2. Click **Token Service**.

   The **Token Service** page appears.

3. Select the **Configure** check box to specify the server name and attach the certificate.

   The confirmation message appears whether you want to configure the token service.

   Note:
   - You need to select **Configure** check box to configure the token service. If not selected, the token service will

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

only get registered.
- If you reconfigure the token service, the existing configuration and the certificate will be deleted. Also, the applications which use the same token service will stop working.
-If the token service is reconfigured, the token service registered from Central Configuration should be re-registered.

4. Click **OK** to configure the token service.
5. In the **Server Name** box, select the server name where AIM is installed.
6. Click the attachment icon against the **Certificate** box to select the certificate. The selected certificate should be in pfx format.
7. In the **Password** box, enter the password for the selected certificate.
8. In the **Token Service URL** box, enter the URL of the server where AIM is installed and configured. The URL format should be *https://servername/identitymanager*. The *servername* should be fully qualified host name.
9. In the **Alias Name** box, enter the alias name for the token service.
10. In the **Registrar User** box, enter the user name used for registering to AIM.
11. In the **Password** box, enter the password for the Registrar user.
12. Click **Register**.

    The token service is registered successfully.
13. Click **Reset** and then repeat steps 3 - 11 to configure and register additional token services for authentication.

Note:
- While registering Token Service, all existing Enterprise Console URIs will be added to redirect URI. If you are creating a new Enterprise Console, you must re-register the Token Service.
- To access the Enterprise Console Web Site, ensure that Anonymous authentication is enabled and Windows authentication is disabled.

Enable or Disable Security for a Repository

To enable or disable the security

We recommend to enable Security for the repositories while working in the Production environment.

1. Connect to Farm database.
2. Query the properties column for the specified repository, as shown in following code snippet:

```
SELECT Properties, * FROM SKRepository
WHERE Name = 'RepositoryName'
```

It will display the following record for the specified repository. Select the properties column value and copy it:

```
<properties>
<property key="spexplorerDisplayLimit"><![CDATA[0]]></property>
<property ey="sourceelements"><![CDATA[WORKFLOWELEMENTS\Default]]></property>
<property key="fromemail"><![CDATA[admin@domain.com]]></property>
<property key="toemail"><![CDATA[admin@domain.com]]></property>
<property key="alertuser"><![CDATA[False]]></property>
<property key="filewatcher"><![CDATA[]]></property>
<property key="enablesecurity"><![CDATA[True]]></property>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<property key="enablewebparts"><![CDATA[True]]></property>
<property key="isadminconfigured"><![CDATA[True]]></property>
<property key="isThemeEnabled"><![CDATA[True]]></property>
<property key="theme"><![CDATA[Default]]></property>
<property key="isDMSEnabled"><![CDATA[False]]></property>
<property key="AutoMapActiveDirectoryProviders"><![CDATA[True]]></property>
</properties>
```

3. Modify the property value for enablesecurity property to ‘True/False’.
4. Update the column value for the specified repository.

```
UPDATE SKRepository
SET Properties = 'Modified properties'
WHERE Name = 'RepositoryName'
```

5. Do an IISRESET.

Enable or Disable Web Parts for a Repository

To enable or disable the Web Parts

We recommend to enable Security for the repositories while working in the Production environment.

1. Connect Farm database.
2. Query the properties column for the specified repository, as shown in below code snippet:

```
SELECT Properties, * FROM SKRepository
WHERE Name = 'RepositoryName'
```

It will display the record for the specified repository, select the properties column value and copy it. It will look something like this:

```
<properties>
<property key="spexplorerDisplayLimit"><![CDATA[0]]></property>
<property ey="sourceelements"><![CDATA[WORKFLOWELEMENTS\Default]]></property>
<property key="fromemail"><![CDATA[admin@domain.com]]></property>
<property key="toemail"><![CDATA[admin@domain.com]]></property>
<property key="alertuser"><![CDATA[False]]></property>
<property key="filewatcher"><![CDATA[]]></property>
<property key="enablesecurity"><![CDATA[True]]></property>
<property key="enablewebparts"><![CDATA[True]]></property>
<property key="isadminconfigured"><![CDATA[True]]></property>
<property key="isThemeEnabled"><![CDATA[True]]></property>
<property key="theme"><![CDATA[Default]]></property>
<property key="isDMSEnabled"><![CDATA[False]]></property>
<property key="AutoMapActiveDirectoryProviders"><![CDATA[True]]></property>
</properties>
```

3. Modify the property value for enablewebparts property to ‘True/False’.
4. Update the column value for the specified repository.

```
UPDATE SKRepository
SET Properties = 'Modified properties'
WHERE Name = 'RepositoryName'
```

5. Do an IISRESET.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

User Providers

Each Repository could potentially have its own user database, for example, one application may connect to Active Directory, the other may connect to AVEVA Work Tasks Databases, and others to Lightweight Directory Access Protocol (LDAP) or Active Directory Application Mode (ADAM).

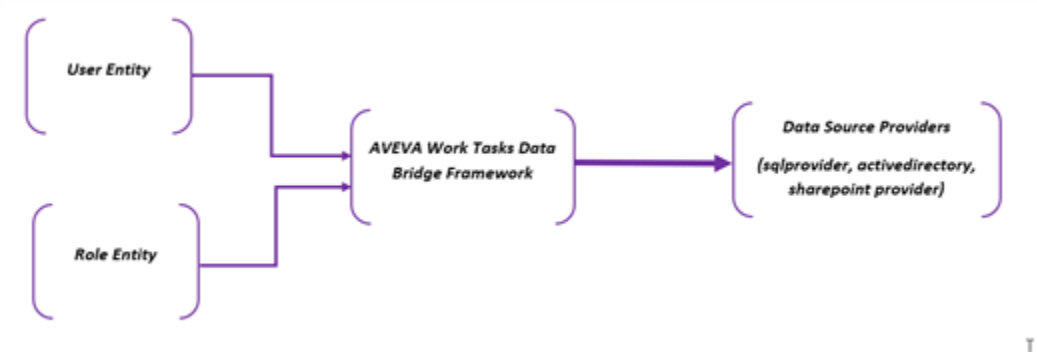
When AVEVA Work Tasks uses each of these Repositories, it needs to ensure that users of these Repositories can work with AVEVA Work Tasks without any complicated configurations. AVEVA Work Tasks also ensures that connection to these repositories are secure. The secure connection indicates that only authorized functionality is available to each and every resource.

Different repositories may be using the same authentication source, but may have different metadata and grouping information with them. For Example, the authentication source could be active directory, but there could be five other repositories which are using the same user and maintain Repository specific user metadata.

The user integration needs to be outside the Repository boundary since users can span across multiple Repositories. In AVEVA Work Tasks, we use a concept known as User Entity and Role Entity for managing users across different platforms.

User Entity manages all the information for the users across different data sources. Data Source may be Active Directory, SQL providers, and so on. The Role Entity manages users in roles.

Data bridge framework is the common framework that bridges the entity item and the corresponding data source providers. Each entity item will internally communicate with the Data bridge framework to know from which Data Source provider the entity details should be retrieved.



Change User Provider Properties

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**. The Central Configuration page appears.
2. Click **Repository Settings**. The **Repository List** page appears.
3. Click the desired repository from the **Repository Name** column.
4. Modify the properties from the **User Provider** properties box.

Add Additional User Providers

AVEVA Work Tasks allows you to have an additional provider for an existing repository. Because of this feature, one repository can make use of users from different providers. The example below illustrates the need to add additional User Providers.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example

There is a need for a AVEVA Work Tasks Repository to communicate to Active Directories from multiple domains. The two Active Directory domains are "Workflow Management" and "NG" and the existing repository is "Main Repo" connected to both the Active Directories.

To add additional User Providers

1. Create a Repository with an Active Directory (AD) as User Provider.

   While creating a Repository select the Active Directory as User Provider from the drop down box. The currently connected Active Directory [AVEVA Work Tasks AD in our case] details need to be filled. If the Active Directory details are not displayed, provide the details manually and check the connection, by clicking Test LDAP button. You can also use AVEVA Work Tasks Active Directory Tool, to retrieve the details.
2. Configure the repository created in step 1 with the other Active Directory [NG] available in the domain.
3. Navigate to the AVEVA Work Tasks Repository Database associated with the Repository.
4. View the User Provider/Authentication provider details maintained in the "SKEntity" and "SKEntityDataSourceProvider" Tables.
5. Locate two entries for the repository from the SKEntity table.

   The DefinitionXML column will have the XML schema, of the Authentication Provider of the repository.

   Sample Entry:

   UserEntity:

   ```
   <entity type="user"><properties><property name="UserName" displayname="Name"
   type="string" /><property name="UserId" displayname="Id" purpose="identifier"
   type="string" /><property name="UserDisplayName" displayname="Display Name"
   purpose="name" type="string"/><property name="UserEmail" displayname="Email"
   purpose="email" display="true" valuecollection="true" type="string" /><property
   name="UserDesignation" displayname="Designation" purpose="designation"
   type="string"/><property name="UserDepartment" displayname="Department"
   purpose="department" type="string"/><property name="UserManager"
   displayname="Manager" purpose="manager" type="string"/><property name="UserRole"
   displayname="Role" purpose="groupname"
   type="string"/></properties></entity>%%$#@@!!!*&
   ```

   RoleEntity:

   ```
   <entity type="role"></entity>%%$#@@!!!*&
   ```
6. Open the SKEntityDataSourceProvider table and make the necessary changes.
   a. Add Two more rows for the new authentication provider, Active Directory details, which is required to associate with the same repository.
   b. Provide the details of the new Active Directory in XML format.
   c. Enter the details in the SKEntityDataSourceProvider, before making the new AD entry. The columns, EntityId are mapped with SKEntity ID column.

   The DisplayName column has the display value. The Definition XML will have the XML schema of attributes used to identify users from the AD.

   Sample Entry:

   ```
   <dataproviders><provider name="activedirectory"
   type="activedirectory"><properties><property name="name" type="string"
   displayname="Resource ID" entitymappedproperty="UserName"/><property
   name="sAMAccountName" persistcolumn="UserIdString" type="string" purpose="identifier"
   displayname="Account Name" entitymappedproperty="UserId"/><property
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
name="displayName" type="string" purpose="name" persistcolumn="Name"
displayname="Resource Name" entitymappedproperty="UserDisplayName"/><property
name="mail" type="string" purpose="email" displayname="Email"
entitymappedproperty="UserEmail"/><property name="title" type="string"
purpose="designation" displayname="Designation"
entitymappedproperty="UserDesignation"/><property name="department" type="string"
purpose="department" displayname="Department"
entitymappedproperty="UserDepartment"/><property name="manager" type="string"
displayname="Manager" suppressdn="true" purpose="manager"
entitymappedproperty="UserManager"/><property name="memberof" type="string"
purpose="groupname" displayname="Group" suppressdn="true"
entitymappedproperty="UserRole"/></properties><Preferences><SearchScope>Sub
Tree</Search Scope><Referral Chasing>External</Referral
Chasing><Timeout>60</Timeout><Size Limit>0</Size Limit><Authentication User>WORKFLOW
MANAGEMENT\anilk</AuthenticationUser><AuthenticationPassword
IsEncrypted="true">iB9rbif4XqNm4+8cO3nnTw==</AuthenticationPassword></Preferences><LD
APServer>LDAP://</LDAPServer><LDAPSearchBase>DC=Workflow
Management,DC=dom</LDAPSearchBase><LDAPBaseFilter>(objectCategory=Person)</LDAPBaseFi
lter><DomainName>WORKFLOW MANAGEMENT\</DomainName></provider></dataproviders>
```

This entry will be created when we create the repository, by providing a user provider. In our case its Active Directory -AVEVA Work Tasks .

7. Add another entry for the other domain that we need to configure with the repository.
   - The new entry will have the similar schema in the DefinitionXML column, as the one above, but the LDAP SearchBase will be pointed to the AD that we want to associate with the repository, as shown in the below sample:
   - The DisplayName column will have the value that we want to display in the EC Site.
   - The InstanceName will have to be different. This is mandatory.
   - The Provider column will have the same as other row entry value, activedirectory in our case.
   - After making the changes, the SKEntityDataSourceProvider will have a new row added as shown.
8. Navigate to the Central Configuration and click on the Repository Provider List link.
9. View the repositories and there respective Authentication Provider details.
10. Select the repository that was created with the changes. Observe that there are two entries under User Entity.

    Notice that the Repository has UserEntity mapped with two active directories-Workflow Management and NG.
11. Click on the Click Here link to view the DefinitionXML schema.

    This confirms that the necessary changes have been applied, and a repository associated with two user providers is created.
12. Navigate to the Enterprise Console page.

    Observe that the repository can be accessed by users of both the domains configured with the Repository.

    Now users from both the Active Directories, AVEVA Work Tasks and NG, can access the repository.

If there are users belonging to both the configured Active Directories, then you need to use <mappingfields> element to map the similar columns of the user properties from both the providers. This element will be part of the DefinitionXML column which has the XML schema.

A sample schema with <mappingfields> element is as below:

```
<mappingfields><mappingfield mappedfromprovider="activedirectory"><fields
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
mappedtoprovider="ngactivedirectory"><field name="sAMAccountName" mappedto="sAMAccountName"
/></fields></mappingfield><mappingfield
mappedfromprovider="ngactivedirectory"><fieldsmappedtoprovider="activedirectory"><field
name="sAMAccountName" mappedto="sAMAccountName" /></fields></mappingfield></mappingfields>
```

Configure User Providers

AVEVA Work Tasks Repository can be configured to use any of the following user providers for user authentication:

- Active Directory Server
- ASP.NET Membership
- Repository User Provider
- Forms
- List
- AVEVA Connect
- Azure Active Directory

User authentication is required to enable users to participate in human activities that form a part of the Workflow.

User authorization is handled in AVEVA Work Tasks through AVEVA Work Tasks security.

Windows Active Directory Server

Select Windows to use Windows Active Directory as the user provider. The logged in Active Directory is displayed by default in the Active Directory Data Source and Search Base fields. Enter another LDAP and search base connection, if required. Use the Test LDAP button from the UI to test the LDAP connection. If Test LDAP is unsuccessful, enter the authentication user name and password and check Test LDAP again.

The user information from Active Directory, like the SamAccountName, Display Name and Email ID, gets stored in a table. This is done to avoid sending frequent requests to the Active Directory. In addition, the user credentials are validated by using the Microsoft Win32 API LogonUser.

ASP.NET Membership

Select ASP.NET Membership. Enter the connection string to the data source and application name. Use the Test ASP.NET Membership button on the UI to test connection. The ASP.NET membership table already contains a column for password. Ensure that the value for the column **PasswordFormat** is set to 0 as this field currently supports only clear text format.

Forms

To retrieve user related information from another application database, select Forms. The following information must be provided for Forms user Data:

- Data source: Enter the data source for the user table.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Data Table: Enter the name of the table that stores user information. If user data is present in multiple tables, create a view and enter the name of the view in this field.
- User Entity Properties: Each column in the table or view can be provided as an attribute to the User Entity. Properties defined here are available to Workflow during design and execution. The User Entity properties must be provided as XML.

Sample XML Schema

```
<properties>
                    <property name="name" type="string" displayname="Full Name"
purpose="name"/>
                    <property name="empname" type="string" displayname="Name"
purpose="identifier"/>
                    <property name="Address" type="string" displayname="Address"/>
                     <property name="Sex" type="string" displayname="Sex"/>
                     <property name="Email" type="string" displayname="Email"
purpose="email"/>
               </properties>
```

Tags used in XML schema are:

- <properties>: The collection of fields that requires to be retrieved from the source database object.
- <property>: A single field reference in the source table or view.

Attributes of the <property> element are:

- name: Refers to the name of the column in the Database.
- type: Refers to the type of the column (int, string or guid).
- displayname: Refers to the name for display within .AVEVA Work Tasks
- purpose: Indicates how AVEVA Work Tasks should treat this column (name, email, and identifier).
- display: Indicates whether the column requires to be displayed or not (true or false).

User Provider Properties

Map the user entity properties to the fields in the user provider. To associate the password information in the Enterprise Console for forms authentication:

In the entity provider, add a column as purpose="password" as in the example below:

```
<property name="Password" type="string” display="false" purpose="password" />.
```

Currently the default provider supports only clear text password.

List

When a Repository is created, AVEVA Work Tasks user tables are automatically added to the Repository database. Select List to use this table. Lists within the AVEVA Work Tasks List Provider can be customized, see User List for more information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

AVEVA Connect

Select AVEVA Connect to use AVEVA Connect as the user provider and authenticate the user. To configure AVEVA Connect, see AVEVA Connect Settings.

Azure Active Directory

Select Azure AD to use Azure Active Directory as the user provider. To configure Azure AD, see Azure Active Directory Settings.

For more information on Azure Active Directory, see the links:

https://azure.microsoft.com/en-in/services/active-directory/

https://docs.microsoft.com/en-us/windows-server/identity/active-directory-federation-services

User List

The User list stores user details forAVEVA Work Tasks List provider, which can be customized to add new fields and edit existing fields. Customization can be done to the User List and the Form associated to the list present within the system folders of Manage List and Manage Forms respectively.

- The User list is present in the **System** folder of **Manage List** and contains customizable pre-defined lists.
- The form associated to the User List is present in **Manage Forms > Dynamic List Form > System** folder and contains customizable pre-defined forms.

Scenario

A workflow using Active Directory as user source needs additional user information for decision making. It may not always be possible to modify the user source to capture additional metadata. Customized User List provides an easy way to store additional information about a user and map the same to Active Directory.

Customizing User List

If the user wants to add any new fields in the User List, the user can go to manage List and edit the user List, add new controls, and regenerate the Form.

If a User wants to see Age filed in the Enterprise Console Settings Users List:

1. Add new column **Age** in SKEUsers Table in repository database.
2. Navigate to **Manage List** > **System** folder, select **Users list** and click **Design** from the ribbon bar.
3. The **List Designer** dialog is displayed
4. Drag and drop the Decimal Input control to the orange section in the Design Area
5. The **Control Property** dialog is displayed.
6. In the **Tag Name** field enter Age and in the **Description** field enter the Age of the User.
7. In the Bound to DBColumn type in Age.

   This field will be bound to the column created in database, hence the value entered in this field should be same as the name given to the column in SKEUserTable.
8. Click **Save**. The Age field will now appear in the List Designer.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Click **Submit** to submit the list and regenerate the Form.
10. Access Users from Enterprise Console Settings and create a new user.
11. The new field Age will be present in the User dialog.

In Manage List if user is adding any new control and if the form is regenerated the password field property will get changed. You have to follow the steps below to reset the properties for the password field:

1. Navigate to **Enterprise Console** > **Forms**.
2. Double click on the **Dynamic List Forms** folder and access the **System** folder.
3. Select the SKAWS_Users list, click **Design** in the ribbon bar.
4. Edit the **Password** field, and make the following changes.
   - In the **Basic** tab, set **Mode of text** as *Password*.
   - In the Appearance tab set:
     - **Persist password on postback** as *True*
     - Encryption mode as **AVEVA Work Tasks List type.**

Manage Users in List Provider

Use the **Users** option in the **Enterprise Console** to enable user maintenance tasks, such as adding new users to the Enterprise Console, editing and deleting existing users, and managing security.

To add additional users

1. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Users**.
   The **Users** page appears.
2. On the action bar, click **New**.
   The New User dialog box appears.
3. In the **Name** field, enter a unique name for the new user.
4. In the **Password** field, enter a password.
5. In the **Description** field, provide an appropriate description for the user.
6. In the Email field, enter a valid email ID.
   The following email IDs are invalid:
   - "Abc\@def"@example.com
   - "Fred Bloggs"@example.com
   - "Joe\\Blow"@example.com
   - "Abc@def"@example.com
   - customer/department=shipping@example.com
   - $A12345@example.com
   - !def!xyz%abc@example.com
   - firstname+lastname@domain.com
   - email@123.123.123.123

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- email@[123.123.123.123]
- "email"@domain.com

7. In the **Department** field, enter the user's department.
8. In the **Manager** field, choose the user's manager from the drop-down list.
9. In the **Role** field, specify the user's role.
10. In the **Mobile Phone Provider**, provide a valid name of the provider.
11. In the **SMS Number** field, provide a valid mobile phone number to send or receive SMS messages.
12. Click **Submit**.

    The new user is added to the User's list.

To edit users

1. Right-click the required user from the User's list and select the **Edit** option from the context menu.

   The **Edit User** dialog box appears.
2. Edit the required fields and click **Submit**.

   The edited user appears in the User's list.

To delete users:

- Right-click the required user from the User's list and select the **Delete** option from the context menu.

  Alternatively, you can select the user and select the **Delete** option from the menu bar.

  The selected user is deleted from the list.

To create users in Repository User Provider using API

To create a new user in Repository User Provider, use the API given below:

```
Skelta.Core.User user = new Skelta.Core.User("Repository Name");
user.Name = "UserName";
user.Email = "UserEmail@domain.com";
Hasher hasher = new Hasher();
byte[] salt = hasher.GenerateSalt();
user.Salt = Convert.ToBase64String(salt);
user.IterationCount = hasher.GenerateIterationCount();
user.Password = hasher.GenerateHash("password", salt, user.IterationCount);
user.Save();
```

Working with Repository-Provider List

The Repository Provider List displays the list of user providers mapped to a selected repository.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**. The Central Configuration page appears.
2. Click Repository-Provider List.
3. Select the Repository name from the drop down box to view the resource provider for a repository.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure Provider Definition XML

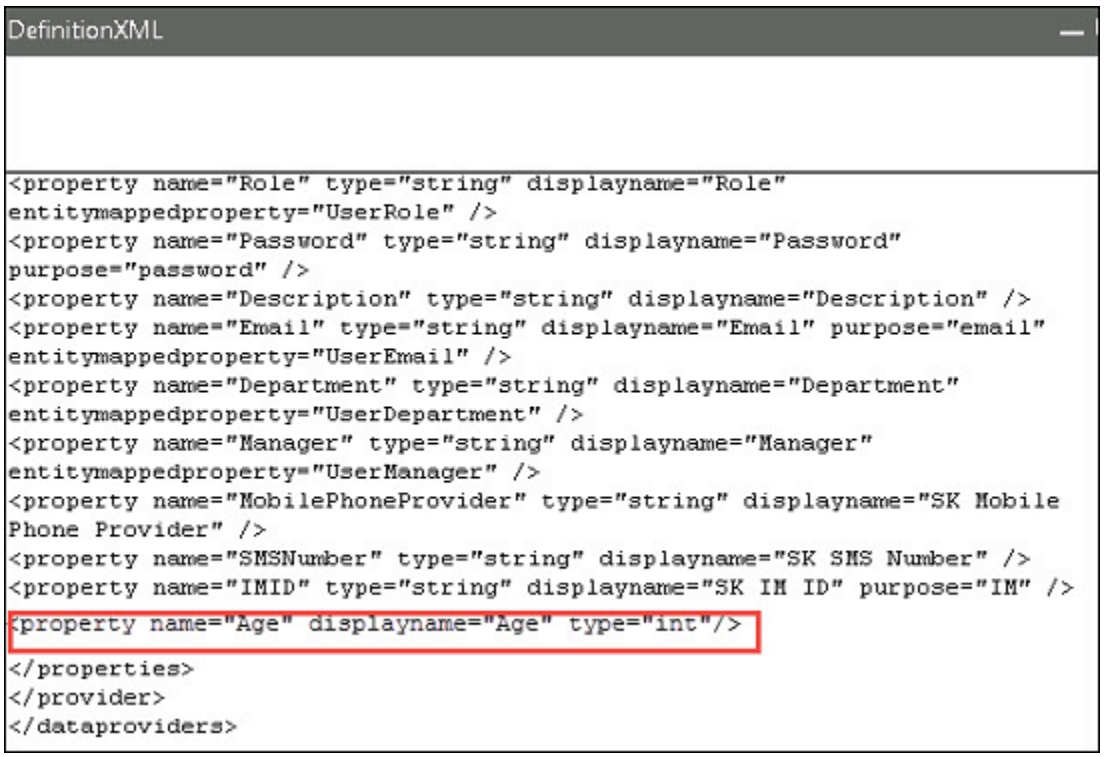
To add new properties

To use new provider properties or fields in a workflow, that property needs to be added in the Provider List Definition. To add a new property or to edit an existing property, follow the steps below:

1. In **Central Configuration**, go to **Repository Provider List**.
2. Select the **AVEVA Work Tasks List** repository and expand it.
3. Access the **Definition XML** of User Entity by clicking on the **Click here** option beside **User Entity**.
4. Add the property for the new field.

   Example:

   <property name="Age" displayname="Age" type="int"/>



5. Select **Edit the definition XM**L and click **Update**.
6. Close the Definition XML.
7. Expand **User Entity** and access the **Definition XML** of authenticationuserprovider.
8. Add the property for the new field.
9. Select the option **Edit the definition XML** and click **Update**.
10. Expand **User Entity** and access the **Definition XML** of authenticationuserprovider.
11. Close the Definition XML.

Performing these steps will ensure all operations can be performed based on the new field added to the User List. If you regenerate the form all the existing properties of the controls of SKAWS_Users form will be overwritten.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Auto Log Off Settings

To enable the **Auto Log Off** property at the provider level, use the `<AutoLogOff>` tag in the **DefinitionXML** and set the auto log off time. For example, `<AutoLogOffTime>5</AutoLogOffTime>`. Place this tag after the `<provider>` tag in the **DefinitionXML** file.

When you define an auto log off time for the repository, ensure the following:

- Enter valid time limit between 1 to 360 minutes. The system does not validate the time entered here, hence if you provide a time beyond the defined limit, you will not be logged out automatically.
- You can map multiple providers to a single repository. When you define the auto log off time both at provider level and repository level, the provider level time set take precedence over the repository level time set.

Note: If the Enterprise Console is opened in the same browser session and the **Auto Log Off** time is set for the repository, the following error message appears:
The user context is not valid. Please log on to the Enterprise Console and try again.

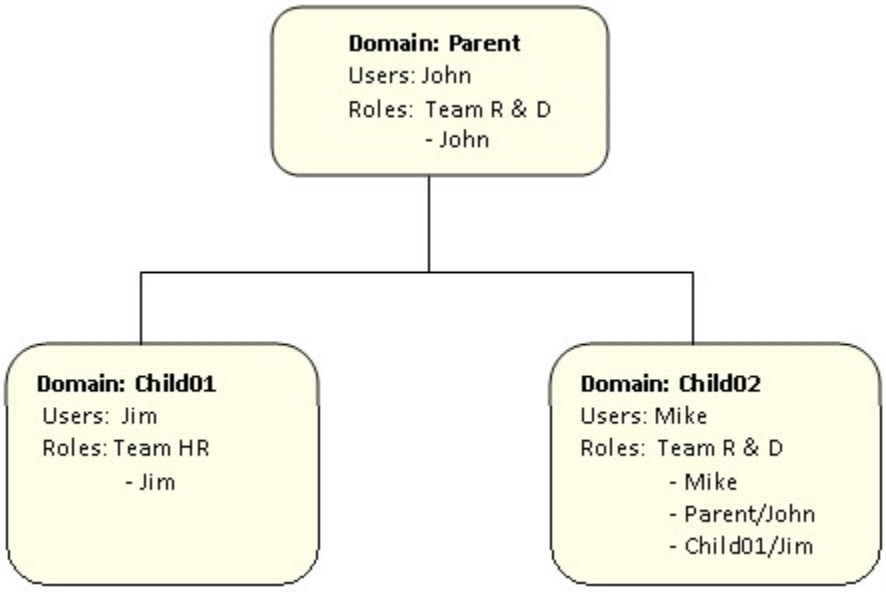
Enable an Existing Repository with Cross-domain Role Provider

Introduction to Cross Domain

An Active Directory role with users from different domains (cross domains) can be mapped to a AVEVA Work Tasks Role using AVEVA Work Tasks Role Provider.

For example, Users 'John' and 'Jim' of domain 'TelematicsUS' and users 'Mike' and 'Bob' from domain 'TelematicsAU' are added to role "Team HR" of domain 'TelematicsMain'. Now the "Team HR" role of domain 'TelematicsMain' contains four users 'John', 'Jim', 'Mike' and 'Bob'.

Consider the following scenario



- **Child01**: Domain Child01 has a user 'Jim' and a role 'Team HR'. User 'Jim' belongs to the role 'Team HR'.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Child02**: Domain Child02 has a user 'Mike' and a role 'Team R & D'. The 'Team R &D' role has the following users:
  - 'Mike' of domain Child02
  - 'John' of domain Parent
  - 'Jim' of domain Child01

To enable an existing repository(without cross-domain role provider) to a Cross-Domain Role Provider.

Prerequisite:

- All available active directory must be added as a provider for the repository.
- The domain controllers must be in the same Forest with parent-child relationship.
- Separate Forests are not supported.
- The **IsCrossDomain** property must be set as *True* for all the active directory role provider in **SKEntityDataSourceProvider** table in the **DefinitionXml** column.

After configuring the prerequisites, see Synchronize Roles.

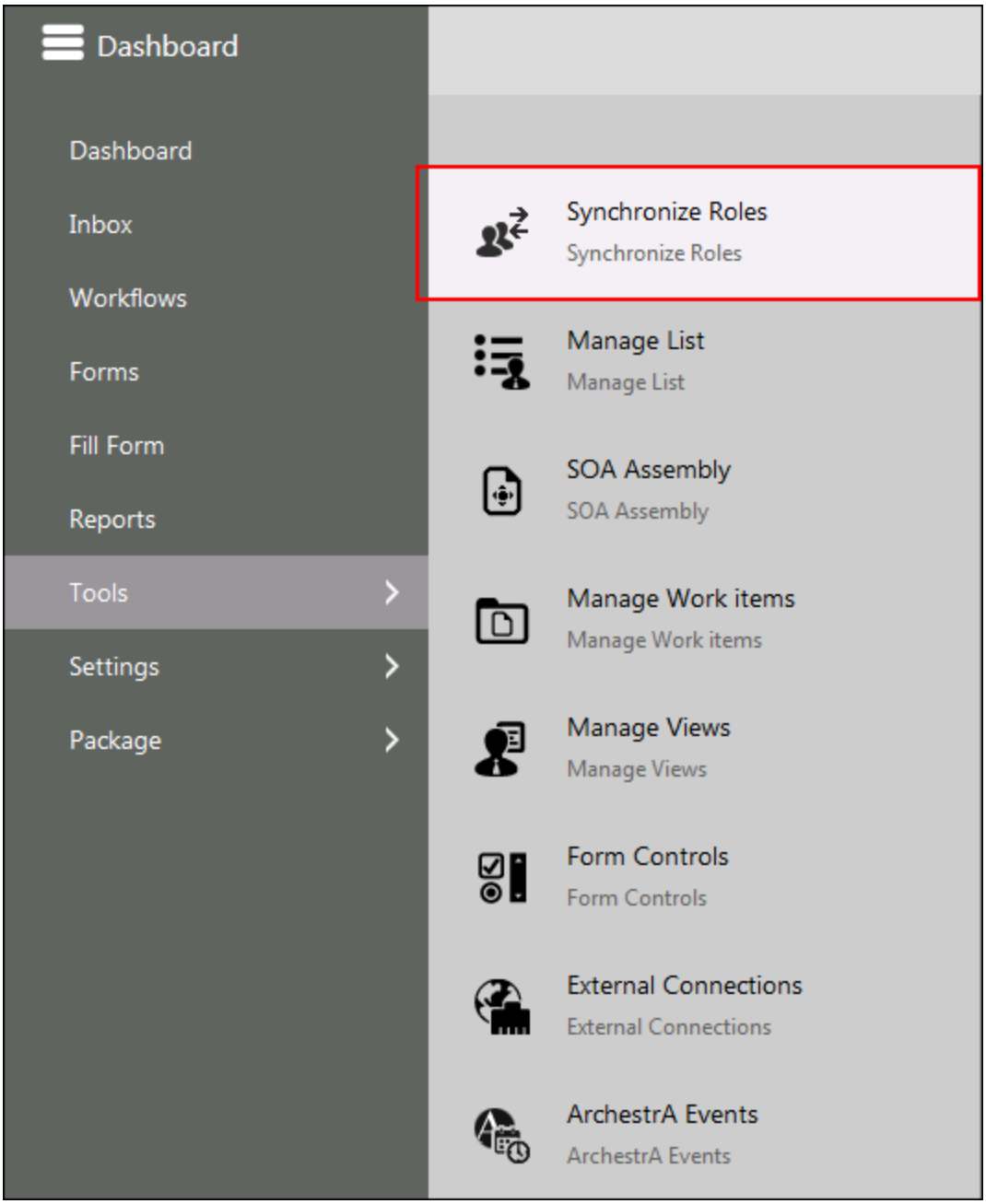
Synchronize Roles

If you want cross domain users and roles mapping across active directory provider, then use the **Synchronize Roles** option to map the roles.

Example: Users 'John' and 'Jim' of domain 'TelematicsUS' and users 'Mike' and 'Bob' from the domain 'TelematicsAU' are added to the role "Team HR" of the domain 'TelematicsMain'.

You can see the Synchronize Roles list item in the Enterprise Console menu that you can use to map the cross domain provider roles. This list item is not available by default in the Enterprise Console menu. While creating a repository, if you choose **Resource Provider** as **Active Directory** and enable the **'Is Cross Domain Role Provider'** option for cross domain users and roles mapping, then the **Synchronize Roles** sub-menu appears in **Enterprise Console** > **Tools**. This menu is available only for users with administrative rights.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



To synchronize roles

1. In the Enterprise console, click the **menu** icon, click **Tools,** and then click **Synchronize Roles**.
   The Synchronize Roles window appears.
2. Enter the role name in the **Search Role** text box.
3. Click **Search**.
   The available roles will be listed in **Choose Roles**.
4. Choose the required role to be synchronized.
   For example, search for 'team' role and it will list the available roles.
5. Click **Synchronize**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

After the synchronization of the selected role(s) is complete, a confirmation message about successful synchronization appears.

Delete Synchronization Cross Domain Tasks

To delete synchronization cross domain tasks

1. Go to Repository Data Source Database.
2. Open **SWTaskList** table.
3. Delete the required synchronization cross domain tasks for repository.

Note: Delete only those records whose Description column is 'Cross domain Role Refresh Task' for that repository.

Synchronize User Properties with Data Source Providers

You can synchronize the user properties with the data source provider. For example, User properties like Name, E-mail, and any persisted columns can be synchronized.

Synchronization can be done for Active and Inactive users but it cannot be done for the users who are not part of the data source provider.

1. In the **Central Configuration**, click **Manage Users**. The **Manage Users** page appears.
2. Click **Synchronize user properties with DataSource Providers**. The **Synchronize Users List** page appears.

Synchronize Active or Inactive User Properties with Data Source Providers

To synchronize Active or Inactive User Properties with Data Source Providers

1. In the **Manage Users** page, click **Synchronize** user properties with DataSource providers. The **Synchronize Users List** page appears.
2. In the User status drop down list, select the user status as Active User or Inactive User.
3. Click **Go** to list the active users.
4. Select the user(s) name from Synchronize Users List, and then click Synchronize Properties.

Example for Synchronizing Active User Properties

1. Log on to **Enterprise Console**.
2. Click **Manage** Users.
3. Click **Synchronize user properties with Data Source Providers**.
4. In the Repository name drop down list, select the repository name as 'SkeltaListRepository'.
5. In the User status drop down list, select the user status as Active User.
6. Click **Go** to list the active users.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To change the user properties of a user, for example, John:

1. In the Enterprise Console, click menu icon, point to Settings, and then click Users. The Users page appears.
2. Click Settings > Users from Enterprise Console pull down menu. The following screen is displayed.
3. Select the user, and then click Edit. The Edit User page appears.
4. Change email ID and password. For example, change 'john@WMDemo.com' to 'john@domain.com'.
5. Click Save. A dialog box confirming the updates appears.
6. Click OK.
7. Go to Synchronize Users List window and select the user. For example, select john.
8. Click Synchronize Properties to reflect the changes.

Schedule Synchronization of User Properties

Schedule Synchronization of User Properties can be used to schedule user properties.

To schedule Synchronization of User Properties

1. In the **Synchronize Users List** page, click Schedule synchronization of user properties. The Schedule User Properties page appears.
2. Click **Add New Task** to add repository. The Repository list appears.
3. Select the repository name.
4. Click **Add**. The Schedule dialog box appears.
5. Select **Start Date** from Date Picker.
6. Select **End Date** from Date Picker.
7. In the **Schedule Task** drop down list, select the task. Task can be on Daily basis, Weekly basis, or Monthly basis.
   - If Schedule Task selected is Daily, mention number of days.
   - If Schedule Task selected is Weekly, mention number of weeks and also select day(s) to run the task.
   - If Schedule Task selected is Monthly, mention date and also select the month(s) to run the task.
8. Select **Start Time** from Time Picker.
9. Check the Repeat Task check box to repeat the task. Select Minutes or Hour(s) from drop down list to repeat the task for specified hour(s) or minute(s).
10. Click **Save** to schedule the task.

Map Active Directory to SQL Provider

You can map Active Directory to SQL Provider.

Advantages of Mapping

- Mapping enables user accounts existing in multiple user providers to login or work on a given task from any of the user providers.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Duplicate entries for a user account from different user providers can be avoided.

Scenario:

Log on to the Enterprise Console using domain login and use the resource properties defined in the SQL provider (form) in workflow design.

To set mapping

Identify the column which is unique for all the providers, that is, user has to identify the common column which shares the same information for the given user across user providers. In this scenario email is considered as the common column.

Active Directory provider properties

```
<dataproviders>
<provider name="activedirectory" type="activedirectory">
<properties><property name="sAMAccountName" persistcolumn="UserIdString" type="string"
purpose="identifier" displayname="Account Name" />
<property name="displayName" type="string" purpose="name" displayname="Resource Name" />
<property name="mail" type="string" purpose="email" displayname="Email" />
<property name="title" type="string" purpose="designation" displayname="Designation" />
</properties>
<Preferences><SearchScope>SubTree</SearchScope>
<ReferralChasing>External</ReferralChasing><TimeOut>60</TimeOut><SizeLimit>0</SizeLimit>
<AuthenticationUser>skelta\testuser</AuthenticationUser><AuthenticationPassword
IsEncrypted="true">tj49KeDsRuxVbVZlZDFLFw==</AuthenticationPassword>
</Preferences>
<LDAPServer>LDAP://</LDAPServer>
<LDAPSearchBase>DC=Testskelta,DC=dom</LDAPSearchBase>
<LDAPBaseFilter>(objectCategory=Person)</LDAPBaseFilter>
<DomainName>TestSKELTA\</DomainName>
</provider>
</dataproviders>
```

SQL Provider Properties

```
<dataproviders>
<provider name="sqlprovider" type="sql server" authenticationsource="false">
<properties><property name="userid" type="int" purpose="identifier" />
<property name="name" type="string" displayname="Name" purpose="name" display="true"/>
<property name="email" type="string" displayname="Email ID" purpose="email"
persistcolumn="Email"/>
</properties>
<datasource>server=skeltadtp75a;database=TestUser;Integrated security=true</datasource>
<SourceTable>users</SourceTable>
</provider>
</dataproviders>
```

Instance name for ActiveDirectory is *activedirectory*.

Instance name for SQL is *sqlprovider*.

To map between Active Directory and SQL provider

The mail property from active directory and email property from database has the common field to map, that is, they both have the same information for the given user. Mail is an active directory property whereas email is a column in SQL provider.

```
<mappingfields>
<mappingfield mappedfromprovider="ActivedirectoryInstance name">
<fields mappedtoprovider="sql provider instance name">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<field name="mail" mappedto="email" />
</fields>
</mappingfield>
<mappingfield mappedfromprovider=" sql provider instance name">
<fields mappedtoprovider=" ActivedirectoryInstance name ">
<field name="email" mappedto="mail" />
</fields>
</mappingfield>
</mappingfields>
```

A user can map between any number of providers and on any common columns. Provider instance name are taken from SKEntityDatasourceprovider table.

To map Active Directory provider with SQL provider

1. Create a AVEVA Work Tasks Repository with Active Directory as resource provider.
2. Insert a record to SKEntityDataSourceprovider with Application set to Repository name, EntityID taken from the SKEntity table (Userentity for mapping users), provider to sqlprovider and Instancename with any UserdefinedName.
3. Update the Defnitionxml column for the added record with appropriate resource XML data.

   Example

```
<dataproviders>
<provider name="sqlprovider" type="sql server" authenticationsource="false">
<properties><property name="userid" type="int" purpose="identifier" />
<property name="name" type="string" displayname="Name" purpose="name"
display="true"/>
<property name="email" type="string" displayname="Email ID" purpose="email"
persistcolumn="Email"/>
</properties>
<datasource>server=skeltadtp75a;database=TestUser;Integrated
security=true</datasource>
Authenticationsource="false" at the provider level disables the login to the
Repository from the Enterprise Console for this provider. Using the
"Authenticationsource" attribute, you can decide which provider to use for user
authentication while log in to the Enterprise Console. Add persistcolumn attribute
for the common properties in both the providers.
Update Definitionxml column of the active directory by adding persistcolumn attribute
for the common property.
Example
<dataproviders>
<provider name="activedirectory" type="activedirectory">
<properties><property name="sAMAccountName" type="string" purpose="identifier"
displayname="Account Name" entitymappedproperty="userid" />
<property name="displayName" type="string" purpose="name" displayname="Resource Name"
entitymappedproperty="name" />
<property name="mail" type="string" purpose="email" displayname="Email"
persistcolumn="Email" entitymappedproperty="email" />
</properties>
<Preferences><SearchScope>SubTree</SearchScope>
<ReferralChasing>External</ReferralChasing><TimeOut>60</TimeOut><SizeLimit>0</SizeLim
it>
<AuthenticationUser>skelta\testuser</AuthenticationUser>
<AuthenticationPassword
IsEncrypted="true">tj49KeDsRuxVbVZlZDFLFw==</AuthenticationPassword>
</Preferences>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<LDAPServer>LDAP://</LDAPServer>
<LDAPSearchBase>DC=Testskelta,DC=dom</LDAPSearchBase>
<LDAPBaseFilter>(objectCategory=Person)</LDAPBaseFilter>
<DomainName>TestSKELTA\</DomainName>
</provider>
</dataproviders>
```

For each of the property there can be entitymappedproperty attribute. This attribute value can be used to map to other property names from different providers if both the purpose are same.

4. Update Definitionxml of SKEntity table for the user entity.

   Example:

```
<entity type="user"><properties><property name="userid" type="string"
purpose="identifier" />
<property name="name" type="string" displayname="Name" purpose="name"
display="true"/>
<property name="email" type="string" displayname="Email ID" purpose="email"/>
</properties></entity>%%$#@@!!!*&<mappingfields>
<mappingfield mappedfromprovider="activedirectory"><fields mappedtoprovider="AD_SQL
provider"><field name="mail" mappedto="email" /></fields></mappingfield>
<mappingfield mappedfromprovider="AD_SQL provider"><fields
mappedtoprovider="activedirectory"><field name="email" mappedto="mail"
/></fields></mappingfield>
</mappingfields>
```

   The properties configured in definition xml of SKEntity table are available to create the query for assigning the activities to the resources. As per the above scenario, we are retaining only SQL provider properties. If we require active directory properties in the workflow design same can be updated in this column, however we need to make sure that the properties defined in the xml are unique. Also it is mandatory to append the mapping column.

   Important: Even if the identifier is of type int, you need to mention it as string globally (in SKEntity table). This is because when we are accessing the resource properties, it is in the string form that is, "providername::userid".

5. When the Activities are assigned to resources, it will look for the resource in both the providers and if common field found for the same, it will assign the activity to the user based on the mapped ID.

User Table

When the user with common field is found and if that users details are already created in SKVirtualActor table, AVEVA Work Tasks automatically maps that user using MappedToId.

Log on to Repository using Active Directory credentials.

Debug

You can debug AVEVA Work Tasks. The various methods implemented to debug include:

- Logger Console
- Setting the Debug Level
- Configuring Exception Email Settings
- Logger Message Categories
- Implementing Work Item Information interface

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Debugging Custom Actions or Properties using Debugger

Logger Console

The Logger Console is used to debug AVEVA Work Tasks. This is a utility that shows a real-time log of events during the execution of AVEVA Work Tasks. It is automatically installed at *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder when AVEVA Work Tasks is installed.

<AVEVA Work Tasks_installed path> is the path where AVEVA Work Tasks is installed on your system.

Steps to Debug

- Run **AVEVAWorkTasksLogConsole.NET2** from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin*. Or

  On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click WTLogger Console.

  The **WTLogger Console** window appears.

  The events are displayed in the WTLogger Console when AVEVA Work Tasks is running. The WTLogger Console displays events using three types of log categories. These log categories are used to identify the type of information.

To set the Debug Level

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click Farm Configuration Wizard.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** screen is displayed.
3. Select **Service Settings**.
4. Click **Next**. The **Service/Global Settings** page is displayed.
5. Click against **Global Settings**. The **Global Settings** screen appears.
6. Click **Log/Transaction**.
7. Select the **Debug Level** from the drop down box.
8. Click **OK**.

The following debug levels are available:

- **Level 1**: This level shows minimal information in the Logger Console. The Logger console displays information related to the start and end of a workflow. Error messages are also shown in this mode.
- **Level 2**: This level provides detailed information of background activities. It will show all the messages logged from the actions.
- **Level 3**: This level displays messages indicating whether an action is executed.

All error messages logged in the Logger Console will be sent to the user e-mail ID. This e-mail ID is picked up from the SMTP node in the configuration file.

To configure Exception Email settings

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Configuration Wizard** appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** screen is displayed.
3. Select **Service Settings**.
4. Click **Next**. The **Service/Global Settings** page is displayed.
5. Click against **Global Settings**. The **Global Settings** screen appears.
6. In the **Exception Email Settings** section, enter From Email and To email.

Logger Message Categories

There are three categories of messages logged in the log files.

- **Information (i)**: This type of message is preceded by a lower case 'i'. It indicates messages containing general information about the events being logged. Each message of this type has four parts:
  - the event information
  - the module generating the event
  - the machine on which the event took place and
  - the date and time of the event
- **Warning (?)**: This type of message is preceded by a question mark '?'. These messages are warning notes to the users about the events being logged. The messages of this type have a structure similar to the 'Information' messages with the same four parts as mentioned earlier.
- **Error (!)**: This type of message is preceded by an exclamation mark '!'. It indicates error messages with details specific to the error. Messages of this type can be further classified into four parts similar to the other two messages. In addition, it has the 'Exception Stack Trace', which occurs before the machine name and date/time. It can be identified by the keyword 'Source' and it can be used to trace the source of the error.

Interpret Log Files

Two types of Log files are created by AVEVA Work Tasks:

- The first type of Log file is created for generic log messages. This Log file is created in the path - <AVEVA Work Tasks_installed path>\log. The Log files are saved in this folder using the date of logging as the filename with the format "MMDDYYYY" (there will be one Log file for each day).
- The second type of Log file is application-specific; it will log messages related to applications. These Log files are created in separate folders corresponding to each application. The folders are located in the path <AVEVA Work Tasks_installed path>\log and are saved with names in the format "<application name>+n" where "n" is a serial number of the log folder created.

To open and view a Log file

1. Navigate to the log file location using the relevant path,that is, "\Logs" or "\Logs\<application name>" depending on the type of Log file you need to view.
2. Locate the required Log file by looking at the date of the log file.
3. Open the log file in a text editor to view the details. It will display all the events for the day using three types of log categories to identify the type of information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Logger Activity

The Logger Activity can be used to log custom messages like information, warnings, and errors which should be logged to the Log Console from the workflow.

Use BAM Report

This report presents details related to failed workflows.

Implement Work Item Information interface

The Workflow Engine alerts the IWorkflowInformation interface on changes in the status of Actions or Workflows. The class implementing the Workflow.NET.Interfaces.IWorkflowInformation.NET2.dll interface is used to get the information from the Workflow Engine on change in the status of Action/Workflow. This interface is used only for getting information. It does not involve any transaction.

Debug Custom Actions/Properties using Debugger

Debugger can be added in the custom code to debug custom development done in AVEVA Work Tasks.

Log Files

These are files that log information or errors related to the product. These files are located in the <AVEVA Work Tasks_installed path>\Logs folder.

The log messages are represented by the following symbols:

- Information (i): This type of message is preceded by a lower case 'i'. It indicates messages containing general information about the events being logged. Each message of this type has four parts:
  - the event information
  - the module generating the event
  - the machine on which the event took place and
  - the date and time of the event
- Warning (?): This type of message is preceded by a question mark '?'. These messages are warning notes to the users about the events being logged. The messages of this type have a structure similar to the 'Information' messages with the same four parts as mentioned earlier.
- Error (!): This type of message is preceded by an exclamation mark '!'. It indicates error messages with details specific to the error. Messages of this type can be further classified into four parts similar to the other two messages. In addition, it has the 'Exception Stack Trace', which occurs before the machine name and date/time. It can be identified by the keyword 'Source' and it can be used to trace the source of the error.

Exception Messages

This topic describes some of the exception messages that appear in AVEVA Work Tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If the **password in Active Directory is modified**, you can see the following error message when you access the Central Configuration:

Cannot proceed further. A valid connection to the AVEVA Work Tasks Farm could not be established. Check the Database User setting specified in AVEVA Work Tasks Farm Configuration Wizard and MSDTC Settings.

You can also see the error message in the logger console as follows:

If the password in Active Directory is modified, you see the following error message when you access the Farm Configuration Wizard:

A valid connection to AVEVA Work Tasks database could not be established. Would you like to change the Database User Settings?

The Logger Console (related to the Farm Configuration Wizard) displays the following message:

To resolve the above issue, follow the steps given below:

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.
   The **Welcome to the AVEVA Work Tasks Configuration Wizard** appears.
2. Click **Next**.
3. The **Edit AVEVA Work Tasks Farm settings** screen appears.
4. Select **Database User Settings**.
5. Click **Next**. The **Database User Settings** screen appears.
6. Update the User Name, Password, and Domain.
7. Click **OK**. A message to ensure that the login has required permissions appears.
8. Click **OK**. A message indicating Database User settings edited successfully appears.
9. Click **OK**.

AVEVA Identity Manager

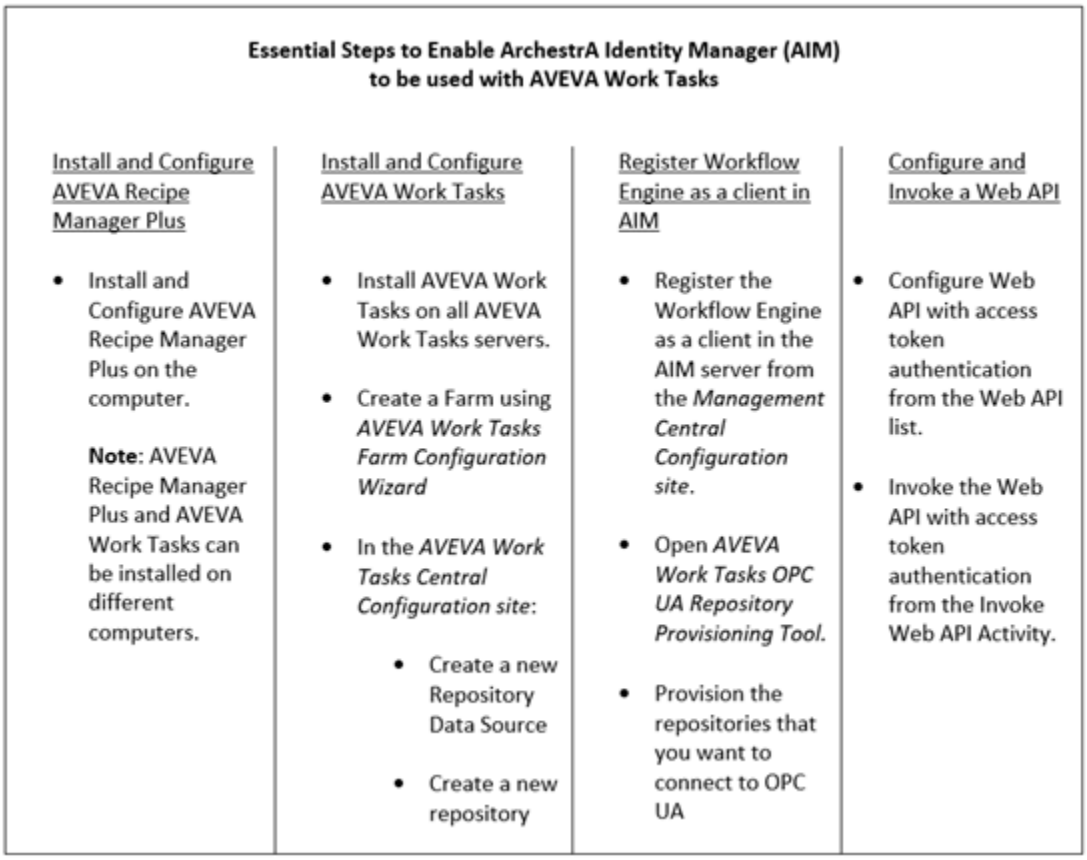
The AVEVA Identity Manager (AIM) provides a mechanism that handles user authentication requirements. It

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

issues a access token to a workflow engine server registered as a client , requesting access to a secure software application.

Enabling AVEVA Identity Manager to be used with AVEVA Work Tasks

The following image shows the essential steps required to enable AIM with AVEVA Work Tasks:



Step 1: Install and Configure AVEVA Work Tasks

- Install and configure **AVEVA Work Tasks**. For more information about installing and configuring workflow, see Installing Wonderware Skelta BPM.

Step 2: Register AVEVA Work Tasks as a client in AIM

- Register **AVEVA Work Tasks** as a client in AIM. For more information about registering as a client, see Configuring and Registering Token Service.

Step 3: Configure and Invoke a Web API

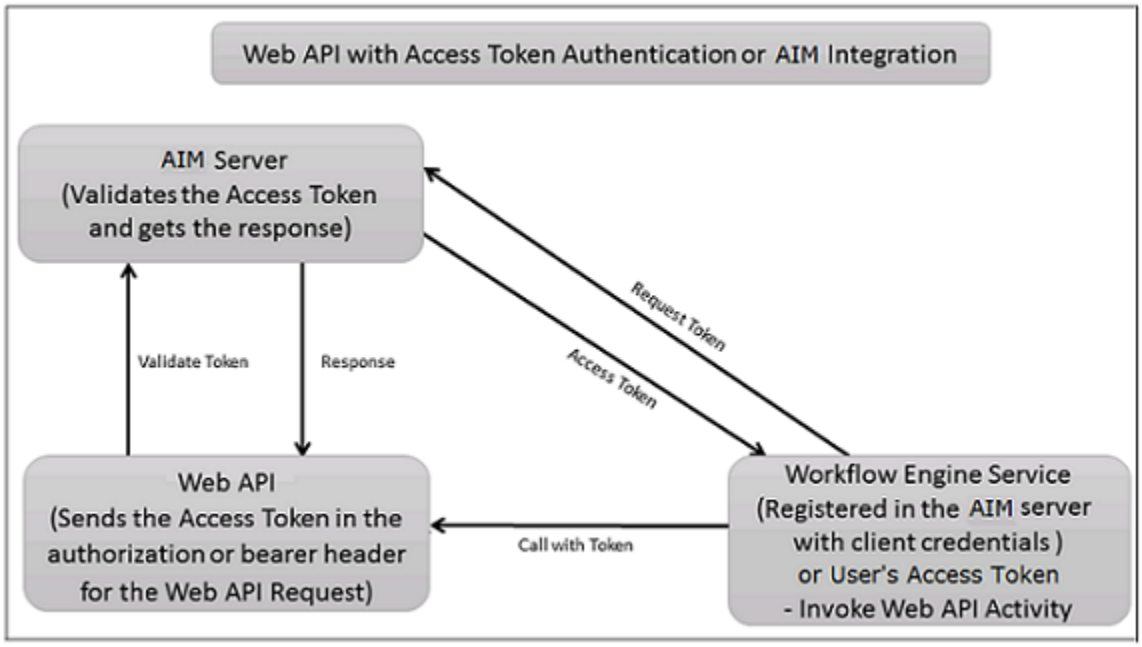
- Configure Web API with access token authentication from the Web API list.
- Invoke the Web API with access token authentication from the Invoke Web API Activity. For more information about configuring and invoking a Web API, see Web API with AIM Authentication.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Web API with AIM Authentication

Web API with Access Token Authentication or AIM Integration

The following image shows the sequence of the Web API Authentication :



1. The workflow engine server is registered in the AIM server. The registered workflow engine server acts as a client and requests the AIM server for an access token through the Invoke Web API Activity.

   This is applicable only if the **Use System Authentication** option in **Web API Activity Configuration** page is selected. If this option is not selected, then the current logged in user's access token is used.
2. The AIM server sends the access token to the client through the Invoke Web API Activity.
3. The Invoke Web API Activity invokes the Web API using the access token. It sends the access token in the authorization or bearer header for the Web API request.
4. The access token passed through the Web API request is validated internally in the AIM server.If the access token is valid, then the Web API request is executed successfully.
5. If the access token is invalid, then the AIM server responds with a 401 Unauthorized Error.

For information about registering a client in the AIM server, see Token Service.

For more information about Web API authentication, see *Web APIs* topic in the *Use Forms and Lists* section.

Consume Access Token from InTouch OMI App in AVEVA Work Tasks Controls

You can pass the access token from InTouch OMI App to AVEVA Work Tasks Controls to consume it for AIM authentication.

To consume access token from InTouch OMI App, update the **view.exe.config** file located at **<System Platform**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

IDE Install path>\Framework\Bin\ViewAppFramework and

<System Platform IDE Install path>\Framework\Bin\ModernEditors\Preview\ViewAppFramework with the following tags:

```
<add key="TokenServiceBase" value="<URL of AVEVA Identity Manager>" />
<add key="ClientId" value="SE_WSP_NG" />
<add key="Scope" value="openid profile system offline_access" />
```

Embedding AVEVA Work Tasks

You can package the AVEVA Work Tasks components within a custom installation.

Before continuing with the customization, ensure that you have installed AVEVA Work Tasks on your computer.

Example

A vendor (OEM customer) may need to package a AVEVA Work Tasks product along with their product installer to avoid a separate installation of AVEVA Work Tasks. This will ease the installation experience in the client environment since AVEVA Work Tasks will get installed along with the vendor's product.

Note: For Custom build application on AVEVA Work Tasks 2023 SP1, or when you upgrade the product to AVEVA Work Tasks 2023 SP1, it is recommended not to use Enterprise Console **appSettings** in web.config file.
For a custom build application, you need to bind the application to the **AVEVA Work Tasks Integrated** pool.

Add an Installation Path

You can define a custom installation path where AVEVA Work Tasks components will be copied during custom installation process. To enable AVEVA Work Tasks on a machine it is mandatory to have the components, UI related files copied to a folder and required libraries added to the global assembly cache.

To add an Installation Path

1. Create a directory in the machine where all AVEVA Work Tasks components and UI files will be copied.

   Example

   <drive>:\<custom_folder path>

   Note: The <drive>:\<custom_folder path> mentioned above can be any drive path on the machine where you would like to install AVEVA Work Tasks in custom installation mode.
2. During installation process, copy AVEVA Work Tasks libraries, UI related folders and files to the custom folder created. The files and folders required to be copied are:
   - BAMReports
   - Bin
   - DatabaseScripts
   - Locales
   - Logs
   - Documentation
   - WorkflowManagementWebService
   - Web

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- WorkflowElements

  Copying contents of Logs folder are an exemption.

  Copy security_approval_workflow.xml from AVEVA Work Tasks installation path to custom path, <drive>:\<custom_folder_path>

3. Add the libraries and dlls from the <drive>:\<custom_folder_path>\Bin folder to Global Assembly Cache (GAC). This can be performed by the installer or using the Visual Studio Command Prompt.

   Use the following command to add the Dlls to GAC:

   gacutil /i "<drive>:\<custom_folder_path>\Bin\Workflow.NET.NET2.dll"

   Example

   Considering that the path <drive>:\<custom_folder_path> created in step 1 is *C:\Program files\AVEVA\Work Tasks*, the following command must be used to add Workflow.NET.NET2.dll to GAC using Visual Studio Command Prompt.

   gacutil /i "*C:\Program files\AVEVA\Work Tasks*\\Bin\Workflow.NET.NET2.dll".

4. Repeat step 3 to add all the dlls to the GAC from the Bin folder except for the following dlls:
   - ASmsCtrl.dll
   - CustomAction.NET2.dll
   - LumiSoft.Net.dll
   - Skelta.HWS.SampleApp.CustomDocView.dll
   - Workflow.NET.ActionHandler.MSOfficeActions.NET2.dll
   - Workflow.NET.SMS.Messenger.NET2.dll
   - Workflow.NET.SMTPManagedSink.NET2.dll
   - Interop.AxMmCtlLib.dll
   - Microsoft.WindowsAzure.Diagnostics.dll
   - Microsoft.WindowsAzure.ServiceRuntime.dll
   - Microsoft.WindowsAzure.StorageClient.dll

To add the installation path mentioned in step 1 to the registry, refer section Entry for AVEVA Work Tasks Folders.

Add Registry Entries

This section provides information on how you can define your custom installation path in the registry, change background images for Farm Wizard, and create a shortcut path for the Central Configuration.

Entry for AVEVA Work Tasks Folders

You can define your custom installation path in the registry.

To define Entry for Folders

1. On the Windows **Start** menu, click **Run**.
2. Type **Regedit**.
3. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

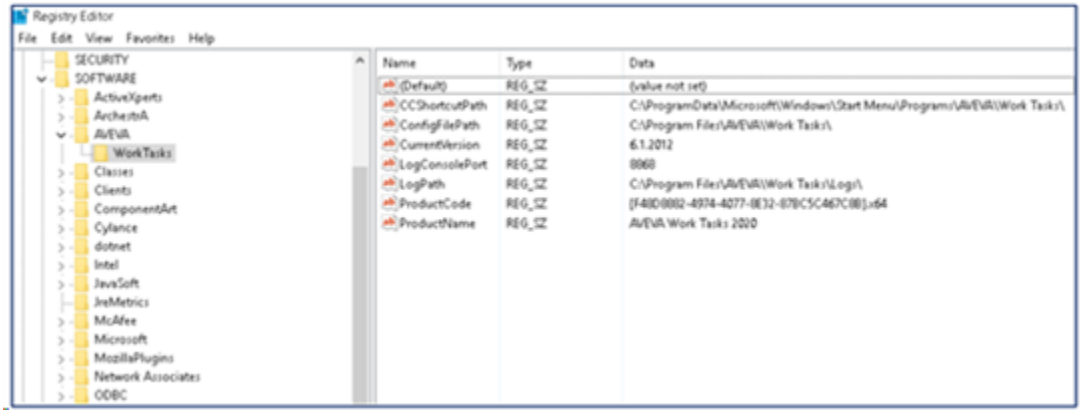
The Registry Editor screen appears.

4. Create a registry entry **HKEY_LOCAL_MACHINE\SOFTWARE\AVEVA\WorkTasks**. The keys required under this entry are *ConfigFilePath, LogConsolePort, LogPath*. The *ConfigFilePath* entry will be the same as the path defined in Add an Installation Path. The *LogPath* entry will save logs in a new folder under the defined installation path.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\AVEVA\WorkTasks]
"ConfigFilePath"="C:\\Program Files\\AVEVA\Work Tasks\\" (This would be the custom
installation path of the software)
"LogConsolePort"="8868"
"LogPath"="C:\\Program Files\\AVEVA\Work Tasks\\Logs\\"  (This would be the custom
installation path of the software)
"CurrentVersion"="x.x.xxxx"   (Specify the current build version)
"ProductName"="\AVEVA\AVEVA Work Tasks 2023 SP1"
"ProductCode"="{F48D8882-4974-4077-8E32-87BC5C467C8B}.x64"   -- for 64 bit machine
```

Note: In all the .txt files in the Locales folder which contains the keyword "AVEVA Work Tasks", change the "AVEVA Work Tasks" to <Your Product Name>.

5. After the entry is created, the registry details will show the AVEVA Work Tasks folder details.



Entry for Farm Wizard Background Image

You change the background images for the Farm Wizard with custom Product Images. The Farm Configuration Wizard is used to install the Enterprise Console, the Central Configuration, define and configure the Service settings.

Add the following string-value pair in registry keys at HKEY_LOCAL_MACHINE\SOFTWARE\WorkTasksSilentInstallation.

To change the Background image for the Farm Wizard

1. On the Windows **Start** menu, click **Run**.
2. Type **Regedit**.
3. Click **OK**.
4. The Registry Editor screen appears.
5. Navigate to \HKEY_LOCAL_MACHINE\SOFTWARE\WorkTasksSilentInstallation
6. Add a String Value, *ConfigWizardImage*.
7. Update the string value "ConfigWizadImage" data value with the new image path.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Image width = 624 and Height = 513

To configure the AVEVA Work Tasks, see Start the Farm Configuration Wizard.

Entry for Central Configuration Shortcut Path

This section provides information on creating a shortcut path for the Central Configuration, if the application is installed in a different path apart from AVEVA Work Tasks installation path.

If the application is installed in default path, it will create a shortcut path based on the operating system version. For example, *C:\ProgramData\Microsoft\Windows\Start Menu\Programs\AVEVA\Work Tasks*

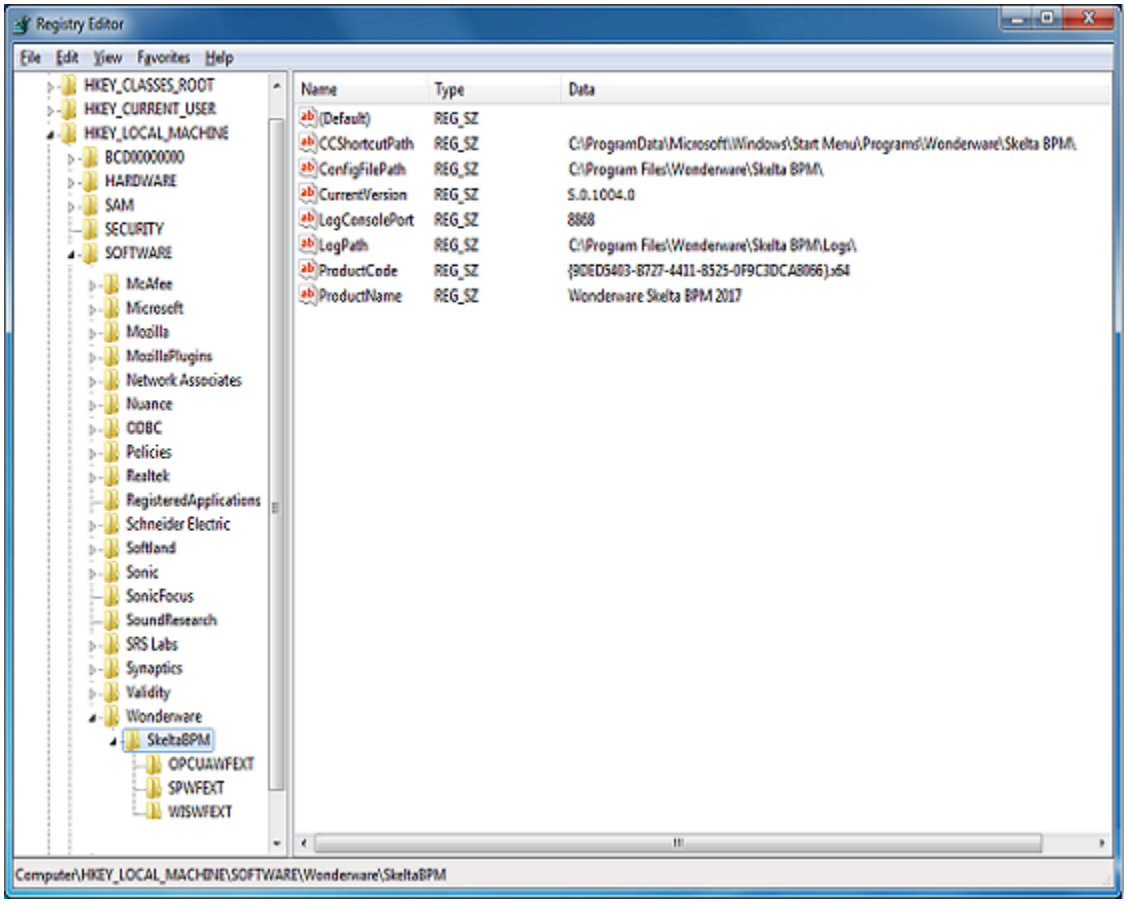
If the custom application is installed in a different path, say Start > Programs > Telematics, then create a shortcut path for the Central Configuration. You need to make a registry entry at:

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Telematics

User need to do the changes accordingly for other Operating Systems.

Add the following string-value pair in registry key.

- Key Name: CCShortcutPath
- Value need to be set as: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Telematics



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Customize UI Text

You can customize the text that appears on the AVEVA Work Tasks user interface.

To customize UI text

1. Navigate to the path **<drive>:\<custom_folder_path>** created in step 1 of the section Add an Installation Path.
2. Navigate to **Locales** > **en-US** folder.
3. Open the files listed in en-US folder in a text editor and make changes to the Value part. Ensure that you change only the Value part of the file.

Install Client Service

This topic provides information to install SKClientSrv.exe (AVEVA Work Tasks Client Service) in services.msc. This installation is required for the Services to function. In the case of default AVEVA Work Tasks installation, the Client service is installed by default. However, for custom installation the Client service have to be manually installed.

To install the Client Service

1. Open Microsoft Visual Studio Command Prompt.
2. Type **installutil -i** "**<drive>:\<custom_folder_path>\Bin\SKClientSrv.exe**".
3. Press **Enter**.

The *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin\SKClientSrv.exe* path displayed in the image above may vary and depends on the Custom installation path selected in Step 1 of the section Add an Installation Path. This service can start only after installing the Farm Server/Client.

Implement FarmEvents Interface

Overview

The IFarmEvent Interface exposes four events to capture the Farm Configuration file's Add/Edit event. This event is triggered when the CentralConfig.xml file is created or edited, before installing the service, and after the services are installed.

The steps mentioned in this section are not mandatory steps. If the user wants to implement IFarmEvents then the user can use the steps mentioned .

CentralConfig.xml file contains database user details and the Farm database connection string details.

Following is the description of event triggers, when a particular method is called:

- **OnCentralConfigFileCreation**: When CentralConfig.xml file is created.
- **OnCentralConfigFileEdit**: When CentralConfig.xml file is edited.
- **OnServiceInstallationStart**: When Service Installation is about to start.
- **OnServiceInstallationComplete**: When service installation is completed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To create a Custom IFarmEvent

1. Create a Custom class library implementing the IFarmEvent.

   DLL Reference

   Skelta.FarmConfiguration

   Workflow.NET.NET2

   Sample Code: A sample implementation code is shown below. This code notifies in the logger file after all the events have occurred.

```
using System;
using System.Collections.Generic;
using System.Text;
using Skelta.Core;
namespace FarmEvents
{
class MyFarmEvents: Skelta.FarmConfiguration.IFarmEvent
{
#region IFarmEventMembers
// Product key should be passed as Empty.
string productKey = "";
public void OnCentralConfigFileCreation(string productKey,
Skelta.FarmConfiguration.HandleSkeltaFarm.DatabaseUserCredentials databaseUser,
string databaseType, string connectionString)
{
Workflow.NET.Log log = new Workflow.NET.Log();
log.LogInformation("************** In OnCentralConfigFileCreation ***********");
log.Close();
}
public void OnCentralConfigFileEdit(string productKey,
Skelta.FarmConfiguration.HandleSkeltaFarm.DatabaseUserCredentials databaseUser,
string connectionString, string databaseType,
Skelta.FarmConfiguration.HandleSkeltaFarm.EditedConfiguration editedConfiguration)
{
Workflow.NET.Log log = new Workflow.NET.Log();
log.LogInformation("************** In OnCentralConfigFileEdit ***********");
log.Close();
}
public bool OnServiceInstallationComplete(Skelta.FarmManager.ServiceMappingCollection
serviceMappings)
{
Workflow.NET.Log log = new Workflow.NET.Log();
log.LogInformation("************** In OnServiceInstallationComplete ***********");
log.Close();
return true;
}
public bool OnServiceInstallationStart(Skelta.FarmManager.ServiceCollection services)
{
Workflow.NET.Log log = new Workflow.NET.Log();
log.LogInformation("************** In OnServiceInstallationStart ***********");
log.Close();
return true;
}
#endregion
#region ISkeltaAddInProvider Members
string _Settings;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Guid _Id;
void ISkeltaAddInProvider.InitializeProvider(string settings, Guid id)
{
this._Settings = settings;
this._Id = id;
}
string ISkeltaAddInProvider.Settings
{
get { return _Settings; }
}
Guid ISkeltaAddInProvider.Id
{
get
{
return _Id;
}
}
#endregion
}
}
```

2. Compile the class and add the dll in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\bin* folder. After this make an entry in the SKAddInProviders table mentioning the type as IFarmEvent. The name can be anything. The class name and the assembly path should be provided accordingly. The class name should be the name of the Custom Class (Namespace.ClassName).

```
INSERT INTO SKAddInProviders
(Type,[Name],ClassName,[Assembly],Settings,IsGacAssembly) VALUES
('IFarmEvent','EventName','NameSpace.ClassName','bin\<DLLName>.dll','',false)
```

Verify Embedded AVEVA Work Tasks

Installation of Embedded AVEVA Work Tasks can be verified by checking if the product name, title, label names, or window names of the product are replaced with the customized names/titles as specified in registry (see Entry for Activating and Deactivating Background Image) (HKEY_LOCAL_MACHINE > SOFTWARE > WorkTasksSilentInstallation > ProductNamePath) or the locales folder. The customized name or title can be seen in different locations in the Farm Configuration Wizard like: Activation and Deactivation windows, Product Name watermark in the Welcome to Farm Configuration wizard, Activate Key window, and the names of the services.

In the earlier sections we saw that the product name was changed from AVEVA Work Tasks to Telematics. Hence, in the Farm Configuration wizard the product name should appear as Telematics.

Farm Recovery

The Farm Recovery is performed to prevent problems that might occur (like data loss or interruptions to organizational operations) and to allow quick recovery.

The following are the files required to be backed up from the folder:

- *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin*: All the Work Tasks dlls and any custom dlls (if added)
- *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Locales* folder: For UI Localization

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\BPMUITemplates* folder
- *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements* folder
- Farm Database
- Repository Data Source Database

Note:
- The Farm Recovery is supported for both HTTP and HTTPS protocol based sites. The HTTPS sites get recovered if the target machine has the same SSL certificate as in the source machine. Otherwise, the sites will get recovered in HTTP mode. After the recovery, edit the **Security Settings** page in **Farm Configuration Wizard** and update the certificate to configure the sites in HTTPS mode.

- After the Farm Backup and Recovery, you need to do the following:
-- Register the token service again and then redeploy the Quickflow application. For information on deploying Quickflow, see Deploy Quickflow. Ensure the primary System Management Server (SMS) is online during token service registration and redeploying the Quickflow application.
-- For OAuth2 authentication, update the **Redirection URI** in **SMTP Settings** and **POP3 Settings** in the **Mail Server** page with the server name on which the Farm Recovery was performed.

Overview

All AVEVA Work Tasks files are located in the AVEVA Work Tasks folder. If a backup is required for the AVEVA Work Tasks installed server machine then copy all the files located in the AVEVA Work Tasks folder. Before taking a backup, ensure that all the Services (Engine, Task Scheduler, and Communication Service) have been stopped and the product has been deactivated.

Restoring AVEVA Work Tasks Files

Before you restore AVEVA Work Tasks files to a different or on the same machine (from which a backup was taken), the AVEVA Work Tasks has to be installed again and activated. During the installation, AVEVA Work Tasks Web sites will be created again. Ensure that the customized data is replaced correctly.

The Farm database can be connected using the Farm Manager utility and the data source can be updated using the Central Configuration Web site.

After the installation, replace all the existing AVEVA Work Tasks files with the backup files. This will ensure that any customization done for XML files, images and Html templates have been replaced as before. Replacing existing AVEVA Work Tasks files will also ensure that any custom actions or properties or dlls added are replaced correctly. The BPMUITemplates folder must be replaced under the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web* path.

Note:
- Ensure that the dlls present in GAC and in *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* are the same.

-The Central Configuration and the Enterprise Console will always be created under *DefaultWebsite/AVEVAWorkTasksSites* after the recovery of the Farm database.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Farm Recovery Scenarios

The scenario descriptions have been described under the following sections:

Server

- when the Server machine goes down
- when the Farm Database goes down
- when both machines go down
- when any one machine goes down

Client

- when the Client machine goes down

Recover Client

- Repository database Recovery

Server

- When the Server machine goes down

Recovery of the Services and the Farm database that are on the same machine

Prerequisites: Take a backup the Farm database.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
2. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.
   The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** screen appears.
3. Click **AVEVA Work Tasks Farm Recovery**.
   The **Recovery Type** screen appears.
4. Select the **Server** option to be recovered from Server or Client options displayed on the screen.
5. Select all the recovery types displayed under the **Server** option:
   - **Database Settings**
   - **Services**
   - **Central Configuration and Enterprise Console**
6. Click **Next**.
7. After a successful connection to the database, if the current server's name does not exist in the database, enter the Server name to be replaced.
   The Services will get installed and the Central Configuration and the Enterprise Console get created.
   A confirmation message appears.

   Note: Central Configuration and the Enterprise Console are always created under **DefaultWebsite/AVEVAWorkTasksSites**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Recovery of the Services and the Farm database on different machines

The Services are installed in machine1 and the Farm database exists in machine2.

Server machine goes down: In this case, only server machine containing the Services is crashed but the Farm database is up.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
2. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** screen appears.
3. Click **AVEVA Work Tasks Farm Recovery**.
4. Select **Server** from **Server** and **Client** options displayed on the screen.
5. Check all the recovery types displayed under the Server option: Database Settings, Services, and Central Configuration and Enterprise Console.
6. Click **Next**.

   Note: Central Configuration and Enterprise Console will always be created under **DefaultWebsite/AVEVAWorkTasksSites**.
7. Under Connection string, enter the details for the Farm database stored on machine2.

   After a successful connection to the database, if the current server's name does not exist in the database, enter the Server name to be replaced. The services get installed and the Central Configuration and the Enterprise Console get created. A confirmation message appears.

Farm Database machine goes down

This section describes the scenario when the machine, on which the Farm database resides, crashes while machine1, where the Services is running, is fine.

Prerequisites: Take a backup of the Farm database.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
2. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** screen appears.
3. Click **AVEVA Work Tasks Farm Recovery**.
4. From the **Server** and **Client** options, select the **Database recovery** check box.
5. Click **Next**.
6. Provide Connection string details for the Farm database that needs to be restored.

When both the machines go down

Refer to the scenario, *Recovery of AVEVA Work Tasks Services and Farm database that are on the same machine*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Moving the Farm database from one machine to another machine

This section describes the recovery in the scenario when you move the Farm Database from one machine (source) to another machine (the destination).

Prerequisites: Take a backup of the Farm Database (from the source machine) and restore it to the destination machine.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
2. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** screen appears.
3. Click **AVEVA Work Tasks Farm Recovery**.
4. From **Server** or **Client** options**,** select the server to be recovered.
5. From the **Server options,** select the **Database recovery** check box.
6. Click **Next**.
7. Provide Connection String details for the Farm Database that needs to be restored.

Any Server machine goes down

In this scenario, temporarily shift the machine to some other machine. After the original server is up, shift back to the original server from the temporary machine.

To recover any server machine

Refer to the scenario, *AVEVA Work Tasks Services and Farm database on one machine for Solution*.

You must run the Farm Recovery options twice in this situation:

- The first time will be in the temporary machine
- The second time will be on the original machine

Client

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
2. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** screen appears.
3. Click **AVEVA Work Tasks Farm Recovery**.
4. From **Server** and **Client** options, select the Client to be recovered.
5. Provide Connection String details for the Farm database.

   The Central Configuration and the Enterprise Console get created.

Note: Central Configuration and Enterprise Console always get created under DefaultWebsite/AVEVAWorkTasksSites.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Farm Recovery

This section provides technical specification details for developers in order to provide a solution for Farm Disaster Recovery.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
2. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** screen appears.
3. Click **AVEVA Work Tasks Farm Recovery**.
4. Choose one of the options displayed on the screen. If you choose server then check one of the options as described below:
   - Recover Database: The **Database User Settings** page appears. On successful database connection this will create the CentralConfig.xml file.

     If the current server's name does not exist in the database, choose the Server Name to be replaced with any server from the Farm Database's Server Collection. This collection will only contain the servers that are currently not running.
     - Database changes
     - SKServer table – Name and Address will get replaced with current server's name.
     - SKService table – WorkflowManagementGlobalSettings – property smtphost, filewatcherip.
     - SKWebsite table – property localURL of each website.
     - SKServicemapping table - SMTPNotificationEngine2 – property listnerip, smtpserverip, pophost.
   - Recover Services: Load the properties of Services from the Farm database and install services.
   - Recover Central Configuration and Enterprise Console.
     - Load the AVEVA Work Tasks Web site properties.
     - If the path of the Web site does not exist then the local URL of the site is changed to the path of DefaultWebSite/AVEVAWorkTasksSites and other properties of the site are updated accordingly.
     - Create Central Configuration and Enterprise Console according to properties from the Farm database.

To recover Client

1. Provide the database connection option.
2. If the current client's name does not exist in the database, choose Client Name to be replaced with any server from Farm Database's Client Collection. SKServer table entry for client will get changed.
3. Recover Central Configuration and Enterprise Console.
   a. Load AVEVA Work Tasks Web site properties.
   If the path of the website does not exist then the local URL of the site is changed to the path of DefaultWebSite/AVEVAWorkTasksSites and other properties of the site are updated accordingly.
   b. Update AVEVA Work Tasks website properties accordingly.
   c. Create Central Configuration and Enterprise Console according to properties from Farm Database.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To recover Repository Database

Repository database information is stored in the Farm database. If the database server is changed, edit the changes from the Central Configuration Web site and update the changes.

To recover Quickflow

Deploy Quickflow again from the Central Configuration Web site to have the Quickflows running.

Move from Development Environment to Production Environment

The following options are available when you to move from development environment to production environment:

- Option 1: To move the Farm database and Repository database
- Option 2: To recover only product database

Option 1: To move the Farm database and Repository database

Development Environment:

1. Stop all AVEVA Work Tasks Services.
2. Create backup copies of the following files from the development environment
   - Any custom dlls if added.
   - *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Locales* folder for UI Localization
   - *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Logs* folder that has all the log files
   - *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**BPMUITemplates* folder
   - *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Workflow Elements* folder
   - Farm Database
   - Repository Database

Production Environment:

1. Restore the backed up Farm database of development environment to production environment.
2. Install AVEVA Work Tasks in production environment.
3. Start the Farm Configuration Wizard on the production computer, and choose Farm Recovery.
4. Select Server or Client to be recovered.
5. Provide the database settings which will connect to restored the Farm database in production environment.
6. Install all the AVEVA Work Tasks Services, Central Configuration, and Enterprise Console.
7. In the Farm Configuration Wizard, go to the Enterprise Console Settings page, and update the URL of all the existing Enterprise Console to the URLs of the production environment.
8. Similarly, update the URLs in the Central Configuration with the production environment URLs.
9. In the Central Configuration, edit all the DataSources, and update the Server instance name, database name, connection settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Repository Data Source recovery:

Repository DataSource information is stored in the Farm database. If the database server is changed, you can edit the Server instance name, database name, and connection settings of Repository DataSource from the Central Configuration web site, and update the changes.

Recovery of AVEVA Work Tasks files:

During the installation of the product, AVEVA Work Tasks web sites will be created again. So here you need to consider the customized files are replaced correctly.

After the installation, replace all the existing AVEVA Work Tasks files with the original files. So that any customization done for XML file (for example, Actions.xml), images, html templates are replaced as before and if any custom actions or properties dlls added are replaced correctly. BPMUITemplates folder must be replaced under *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web* path.

Option 2: To recover only product database (With fresh Farm creation or recovered Farm, and if the Repository Data source does not exist in recovered Farm):

1. Creating new Repository DataSource:
   a. Export the workflow definition from development machine using Import or Export option in the Process Designer.
   b. Create a new Repository DataSource in production environment from the Central Configuration.
   c. Create a new Repository in production environment with the same Repository name as in the development environment.
   d. Import the exported (exported in step a.) workflow definition in production environment using Import or Export option in the Process Designer.
   e. Queues have to be re-configured if present in development environment. Queue will not get exported automatically.
2. Attaching existing Repository DataSource:
   a. Create a backup copy of the Repository DataSource database from the development computer.
   b. Recover this DataSource database in production computer.
   c. Create a new DataSource in production computer and use the same database created in step a. (For example, DataSource is created with name ProductionDatasource. Restored database named 'BackedUpdatabase' is used to create this data source.)
   d. Create a new repository. Choose Attach DataSource option.
   e. Select DataSource created in step b. This lists all the repositories in selected data source. Create the repository.(Example: Check Attach Data Source option. Select ProductionDatasource. This will list all repositories of ProductionDatasource. Now select the repository which needs to be copied and go ahead with Repository creation.)
   f. This will copy the repository details of the development environment to the production environment.

Repository Data Cleaning

1. If you do not want to keep Workflow runtime data, then run following script on the production database:

```
truncate table SKWorkItem
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
truncate table SKHWSActivity
truncate table SKWorkItemTransaction
Delete SWTaskList where cast(ClassName as nvarchar(max)) <>
'Skelta.WorkFlow.Scheduler.WorkFlowScheduler'
truncate table SKMultipleTimeoutWarning
truncate table SWActionEventLog
truncate table SWActionEventLong
truncate table SKWorkItemCounter
truncate table SKChannelWorkItem
truncate table SKWorkItemData
truncate table SKWorkItemFields
truncate table SWExecutionDetails
truncate table SWExecute
truncate table SWVariables
truncate table SKWorkflowEvents
truncate table SKChildWorkflows
truncate table SWCustomTimerDetails
truncate table SWClientTokenData
```

2. Queue Data:

   Queues will work fine only in case where Queue Participants are same in development and production environment. If different users need to be used in Queue on the production environment, user needs to create new Queue.

   To remove particular Queue data, see the following Queue tables:

   - SKQueue
   - SKQueueAccessRights
   - SKQueueParticipants

Reuse Repository Data

If you want to reuse existing repository data, such as workflows, forms into a new repository, complete the following steps:

1. Create new Repository Data source
   a. Export the workflow definition from development machine using Import or Export option in the Process Designer.
   b. Create a new Repository Data source in production environment from Central Configuration.
   c. Create a new Repository in production environment with the same Repository name as in development environment.
   d. Import the exported (exported in step a.) workflow definition in production environment using Import or Export option in the Process Designer.
   e. Queues have to be re-configured if present in development environment. Queue will not get exported automatically.
2. Attach existing Repository Data source
   a. Take a backup of Repository Data Source database from Development machine and recover this data source database in Production machine. (Let us say that the restored database's name is 'BackedUpdatabase')
   b. Create a new Data source in Production machine and use the same database created in step c. (Example:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Data source is created with name ProductionDatasource. Restored database named 'BackedUpdatabase' is used to create this data source.)

c. Create a new repository – Choose attach data source option.

d. Select Data source created in step c. This will list all the repositories in selected data source. Give name to the Repository and continue with creating the repository. (Example: Check Attach Data Source option. Select ProductionDatasource. This will list all repositories of ProductionDatasource. Now select the repository which needs to be copied and proceed with Repository creation.)

e. You can retain the resource providers in the attached data source by checking the Attach Providers option or create your own resource provider. Ensure that it is the same as the resource provider of the attached repository.

Note: The product installed version and data source version should be same for attaching data source.

Example:

Attaching a data source that was created in 5.0.0.0 build should not be performed in the 6.0.0.0 build.

Note: Choose the version number appropriately based on the environment.

To attach such a data source:

1. If the data source was created in 5.0.0.0 build, then the entire 5.0.0.0 build should be first upgraded to 6.0.0.0 build.
2. The 5.0.0.0 build should be upgraded to 6.0.0.0 build by running the latest msi file.
3. The Farm, Data Sources , Repositories, and Lists of 5.0.0.0 build should be upgraded to 6.0.0.0 by running the AVEVAWorkTasksUpgradeManager.exe located in the BIN folder.
4. After upgrading all the components to 6.0.0.0 build, that data source can be used in the 6.0.0.0 build that is installed in any machine by using this feature.
5. This will Copy the repository details of Development environment to Production environment.

Repository Data Cleaning

If you do not want to keep Workflow runtime data, then run the following script on Production database.

```
truncate table SKWorkItem
truncate table SKHWSActivity
truncate table SKWorkItemTransaction
truncate table SWTaskList
truncate table SKMultipleTimeoutWarning
truncate table SWActionEventLog
truncate table SWActionEventLong
truncate table SKWorkItemCounter
truncate table SKChannelWorkItem
truncate table SKWorkItemData
truncate table SKWorkItemFields
truncate table SWExecutionDetails
truncate table SWExecute
truncate table SWVariables
truncate table SKWorkflowEvents
truncate table SKChildWorkflows
truncate table SWCustomTimerDetails
truncate table SWFECount
truncate table SWRaw
truncate table SKScheduler
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
truncate table SKEventBindings
truncate table SKEEventAssociations
truncate table SKExecutionInstanceData
```

Note: You need to recreate all associations and BAM scheduling.

Queue Data:

Queues will work fine only when the Queue Participants are same in both development and production environment. If different users need to be used in Queue on the production environment, the user needs to create a new Queue.

To remove particular Queue data, see the following Queue tables:

- SKQueue
- SKQueueAccessRights
- SKQueueParticipants

Deployment Options

Deployment is the process of moving the solution from one machine to the another machine.

You can deploy the solution developed in **AVEVA Work Tasks** using multiple methods:

- Packages
- Farm Recovery
- Attach DataSource

The preferred deployment option is exporting and importing the packages.

Attach DataSource

Attach DataSource feature in **AVEVA Work Tasks** allows you to move a datasource from one environment to another. The **Attach DataSource** option is available in the **Central Configuration**.

A DataSource can have multiple repositories. For more information on repositories, see Create a Repository.

Refer Farm Recovery, if you want to move Farm Database.

Note: Ensure to use the same passphrase across the DataSource being moved from one environment to another.

Steps to Perform Attach DataSource

The following are steps to perform attach datasource:

1. Backup and restore the repository database from source to destination database server.
2. Navigate to **Central Configuration**.
3. Create the datasource with the backed up database. For more information, see Create a Repository DataSource.
4. Create the repository with the datasource created in **Step 3**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Select the **Attach DataSource** option.

   All the repositories associated to the selected datasource will be listed under **Repositories in the selected datasource** option.
6. Select a repository and enter name for the new repository.

   This name can be same as the name in the source machine or different from the source machine which is not listed in the destination machine.
7. Select a resource provider for the new repository.

   The resource provider should be same as of the previous one.

   If the repository is associated with multiple providers or a single provider, then select **Attach Providers** check box from the **Resource Provider**.
8. Select other default options and click **OK**.

   This will copy the repository details of the source to destination machine.

Note:
-It is recommended not to change the resource provider while performing **Attach DataSource**.

-AVEVA Work Tasks product version, including Hotfix, if any, should be same between Source and Destination.

-It is recommended not to perform **Attach DataSource** when the backup database has large transactional records. The completed records must be archived periodically.

User Scenarios

This section provides information on the following user scenarios and their solutions:

- A user develops a solution using AVEVA Work Tasks in the source machine and then deploys the solution to the destination machine.
- An operational production solution on AVEVA Work Tasks requires an upgrade to a new SQL server or supported operating system.
- Disaster recovery of an operational production solution in the event of system crash.
- For regulatory requirement, user needs BAM reports from an archived database.

Deploys a Solution from One Machine to Another

Description: A user develops a solution using AVEVA Work Tasks in the Source machine and then deploys the solution to the Destination machine.

Solution:

1. Create a backup of the repository database from the source machine.

   Example: Consider Database Name as “**TelematicsDB**”, Backup Filename as “**TelematicsDBBackup.bak**”, and Repository name as “**TelematicsRepo**”.
2. Restore the backed up database in the destination machine.

   Example: After restoring “**TelematicsDBBackup.bak**” in the destination machine, “**TelematicsDB**” database

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

will be created.

3. Create a new DataSource in the destination machine and use the same database created in **Step 2**.

   Example: Create a DataSource with name "**TelematicDataSource**" with the "**TelematicsDB**" database.
4. Create a new repository and select the DataSource created in **Step 3**.
5. Choose the **Attach DataSource** option.

   This lists all the repositories in selected datasource.
6. Select the repository which needs to be copied and proceed with creating a repository.

   Example: Select "**TelematicsRepo**".

   This will copy the repository details of the source to the destination machine.

Also, refer Steps to Perform Attach DataSource.

Solution Requires an Upgrade

Description: An operational production solution on AVEVA Work Tasks requires an upgrade to a new SQL server or supported operating system.

Solution: For solution to this scenario, refer Deploys a Solution from One Machine to Another.

Disaster Recovery of a Solution

Description: Disaster recovery of an operational production solution in the event of system crash.

Solution: For solution to this case, refer Best Practices for Maintaining AVEVA Work Tasks Databases and Steps to Perform Attach DataSource.

BAM Reports from an Archived Database

Description: For regulatory requirement, user needs BAM reports from an archived database.

Solution:

1. Create a backup of the repository database from the source machine.

   Example: Consider Database Name as "**TelematicsDB**", Backup Filename as "**TelematicsDBBackup.bak**", and Repository name as "**TelematicsRepo**".
2. Restore the backed up database in the destination machine.

   Example: After restoring "**TelematicsDBBackup.bak**" in the destination machine, "**TelematicsDB**" database will be created.
3. Create a new DataSource in the destination machine and use the same database created in **Step 2**.

   Example: Create a DataSource with name "**ArchiveDataSource**" with the "**TelematicsDB**" database.
4. Create a new repository and select the DataSource created in **Step 3**.
5. Choose the **Attach DataSource** option.

   This lists all the repositories in selected datasource.
6. Select the repository which needs to be copied and proceed with creating a repository.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example: Select “**TelematicsRepo**”.

This will copy the repository details of the source to the destination machine.

Also, refer Steps to Perform Attach DataSource.

Uninstall AVEVA Work Tasks

This section provides information on uninstalling Client, Server, Farm, and AVEVA Work Tasks.

- Uninstall Client
- Uninstall Server
- Uninstall Farm
- Uninstall AVEVA Work Tasks

Uninstall Client

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
2. Right-click Farm Configuration Wizard, and then click **Run as administrator**.
3. Click **Next**.

   The Edit **AVEVA Work Tasks** Farm settings screen appears.
4. Click **Uninstall Client**.
5. Click **Next**.

   The message about successful uninstallation of AVEVA Work Tasks Client appears.

Note: Uninstalling the Client makes the Enterprise Console unavailable. The underlying repositories and workflows will continue to exist and function.

For more information about complete uninstallation of AVEVA Work Tasks, see Uninstall AVEVA Work Tasks.

Uninstall Server

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click Farm Configuration Wizard.

   The Welcome to the **AVEVA Work Tasks** Farm Configuration Wizard page appears.
2. Click **Next**.

   The Edit **AVEVA Work Tasks** Farm settings page appears.
3. Select **Uninstall Server**.
4. Click **Next**.

   A message appears indicating that you will not be able to connect to any Repository from the current server and all the Services will be unprovisioned.
5. Click **Yes**.

   A message indicating *Uninstalled Server successfully* appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Click **OK**.

Note: Uninstalling the Server does not remove the Farm database and the Repository database. However, it removes the Service, Server, Farm, Data Source, Repository, Service Mapping, and Repository Mapping entries from the Farm database.

For a complete information about uninstalling AVEVA Work Tasks, see Uninstall AVEVA Work Tasks.

Uninstall Farm

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then select **Farm Configuration Wizard**.
2. Right-click **Farm Configuration Wizard**, and then click **Run as administrator**.

   The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** screen appears.
3. Click **Next**. The MSDTC settings are verified.
4. Click **Yes** to continue with current MSDTC settings.

   The Edit **AVEVA Work Tasks** Farm settings window appears.
5. Select **Uninstall Server**.
6. Click **Next**.

   A message appears indicating that you will not be able to connect to any Repository from the current server and all the Services will be unprovisioned.
7. Click **Yes**.

   A message indicating *Uninstalled Server successfully* appears.
8. Click **OK**.

   The Farm is uninstalled successfully.

Uninstall AVEVA Work Tasks

Before uninstalling AVEVA Work Tasks, ensure to complete the following:

1. Uninstall all the associated connectors and extensions.
2. Uninstall the Farm. For more information, see Uninstall Farm.

   If you start the uninstallation of AVEVA Work Tasks before uninstalling the Farm, then the dialog box appears showing the steps to uninstall AVEVA Work Tasks.
3. Click **OK**.

   In this scenario, you have to uninstall the Farm and then uninstall AVEVA Work Tasks.

To uninstall AVEVA Work Tasks

1. Go to **Control Panel**, and click **Add or Remove Programs**.

   The Add or Remove Programs screen appears.

   Based on the Windows version on your computer, this screen might be different.
2. Select the appropriate version of **AVEVA Work Tasks**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Click **Uninstall** or **Remove** to begin the uninstallation.

   A message appears asking your confirmation to uninstall AVEVA Work Tasks.
4. Click **Yes**.

   The progress of uninstallation is displayed.

   A message appears indicating that the system will be restarted to update files and/or services.
5. Click **OK**.

   AVEVA Work Tasks is uninstalled successfully.
6. After uninstalling the AVEVA Work Tasks setup, the following folders will still remain in the installed folder. Remove these folders manually.
   - BIN folder (containing Licensing related files)
   - LOG folder (containing Log files)
   - BAMReports folder

Upgrading AVEVA Work Tasks

- You can upgrade to **AVEVA Work Tasks 2023 SP1** from the following version:
  - AVEVA Work Tasks 2023
- You can upgrade to **AVEVA Work Tasks 2023** from the following versions:
  - AVEVA Work Tasks 2020
  - AVEVA Work Tasks 2020 Update 1
  - AVEVA Work Tasks 2020 Update 2
- You can upgrade to **AVEVA Work Tasks 2020 Update 2** from the following versions:
  - AVEVA Work Tasks 2020
  - AVEVA Work Tasks 2020 Update 1
- You can upgrade to **AVEVA Work Tasks 2020 Update 1** from the following version:
  - AVEVA Work Tasks 2020
- You can upgrade to **AVEVA Work Tasks 2020** from the following versions:
  - Workflow Management 2019
  - Workflow Management 2019 Update 1

Upgrade Scenarios

The following are the upgrade scenarios:

- Upgrade in Standalone Environment
- Upgrade in Client - Server Environment
- Upgrade in Enterprise Server Environment

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note

- The upgrade from AVEVA Work Tasks 2023 to AVEVA Work Tasks 2023 SP1 can be performed only after upgrading to AVEVA Work Tasks 2023 from any of the previous versions.

- This upgrade executes faster on computers with Windows Installer version 5.0. However, in computers with Windows Installer version 4.5 and lower, the upgrade process might take 40 to 45 minutes approximately, based on the configuration of the computer. This is due to the InstallValidate sequence which runs during the upgrade process and compares all the AVEVA Work Tasks 2023 files with AVEVA Work Tasks 2023 SP1 files.

Upgrade in Standalone Environment

In this scenario, the following AVEVA Work Tasks components are upgraded:

- Databases schema
- Repository properties
- List definitions
- Sites
- Services

Delete **WorkTasksTemp** and **AVEVA Work Tasks Cache** files from the client computer.

For information about the upgrade procedure, see Upgrade Procedure.

Upgrade in Client - Server Environment

Upgrade on Server Machine

In Server machine, the **AVEVAWorkTasksUpgradeManager** will upgrade the following AVEVA Work Tasks components:

- Databases schema
- Repository properties
- List definitions
- Sites
- Services

Upgrade on Client Machine

In the Client computer, the **AVEVAWorkTasksUpgradeManager** upgrades the **AVEVA Work Tasks Sites**.

Upgrade in Enterprise Server Environment

Before starting the upgrade of Enterprise Server Environment, stop the AVEVA Work Tasks services on all the machines where Enterprise Server is configured.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Upgrade on Server Machine
- Upgrade on Client Machine

Upgrade on Server Machine

In an Enterprise Server computer, the **AVEVAWorkTasksUpgradeManager** upgrades the following AVEVA Work Tasks components:

- Databases schema
- Repository properties
- List definitions
- Sites
- Services

For information about the upgrade procedure, see Upgrade Procedure.

Upgrade on Client Machine

In the Client computer, the **AVEVAWorkTasksUpgradeManager** upgrades the AVEVA Work Tasks Sites of AVEVA Work Tasks.

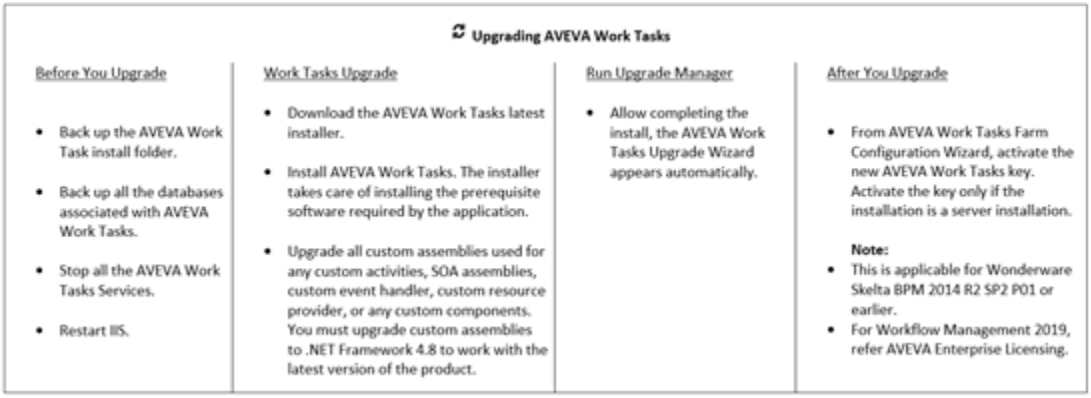
Note: In the **Enterprise Server Environment,** if a client computer upgrade is done, then the **Advance Server** service is manually installed on the client computer.

Use *installutil* command from Visual Studio command prompt to install the **Advance Server** service on the client computer.

For example, installutil –i "*C:**Program files**AVEVA**Work Tasks*\\Bin\AdvServr.NET2.exe".

Upgrade Procedure

The following image shows a graphical representation of the upgrade process of **AVEVA Work Tasks**:



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Before You Upgrade

Before you upgrade to the latest version of AVEVA Work Tasks, perform the following:

- Set up the computer to meet the requirements to run the installer of AVEVA Work Tasks.
- Back up the entire installed folder from the Install Path (For example: <Install Drive>:\Program Files\AVEVA\Work Tasks) and all the databases associated with the previous version of AVEVA Work Tasks. If there are any custom activities defined, you need to add these file details manually after the upgrade is complete.
- Create backup copies of all your existing databases, as the Database schema and content are changed during the upgrade process. Most databases provide a backup wizard or other user assistance for creating database backups. Contact your database administrator for more information.
- Upgrade all custom assemblies, such as custom activities, SOA assembles, custom event handler, custom resource provider, or any custom components to support .Net Framework 4.8 and to refer to the latest version of AVEVA Work Tasks assembly references.
- Ensure to use Connectors and Extensions, which are compatible with the latest version of AVEVA Work Tasks.
- Restart IIS.
- Perform the following tasks manually, if the **Upgrade Manager** fails to automatically complete any of these tasks. Ensure that you have Administrative privilege and sufficient rights to perform the following tasks.
  - Stop the following AVEVA Work Tasks services:
    - Client Service
    - Workflow Engine
    - Task Scheduler
    - Communication Service
    - Mobile Notification Service
    - To upgrade an Enterprise Server version, ensure to stop the Advance Server service.
    - To upgrade in a client computer, ensure to stop the Client service.
    - To upgrade in a client machine which is connected to an Enterprise Server, ensure to stop both the Advance Server service and Client service.
  - To stop IIS, run **Command Prompt** as an **Administrator**, type **iisreset /stop**, and then press **Enter**.
  - Delete the WorkTasksTemp folder from *C:\Windows\Temp*.
  - Delete the AVEVA Work Tasks Cache files and folders from *C:\Windows\Microsoft.NET\Framework64\v4.0.XXXXX\Temporary ASP.NET Files*.
  - Start the AVEVA Work Tasks related services.
  - To start IIS, run **Command Prompt** as an **Administrator**, type **iisreset /start**, and then press **Enter**.
- If you are using Oracle database for lookups, grids, or database activities, then ensure to reconfigure Oracle Data Handler.

Note: When you upgrade the product from earlier versions to AVEVA Work Tasks 2023 SP1, copy **TNSNAME.ORA** file from <Oracle Installation path>\NETWORK\ADMIN folder to the folders:
- <C:\Program Files\AVEVA\Work Tasks\Bin>
- <C:\Program Files\AVEVA\Work Tasks\Web\EnterpriseConsole\Bin>
- <C:\Program Files\AVEVA\Work Tasks\Web\EnterpriseConsole>

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Upgrading to AVEVA Work Tasks 2020

You can upgrade to AVEVA Work Tasks 2020 from Workflow Management 2019 and Workflow Management 2019 Update 1. However, if you want to upgrade from versions earlier than Workflow Management 2019, you must first upgrade to Workflow Management 2019 and then to AVEVA Work Tasks 2020.

When you upgrade the product to AVEVA Work Tasks 2020, the earlier product version is uninstalled and all the files are removed. Therefore, we recommend that you back up the files before upgrading the product.

Make sure that the user has appropriate rights for installing and upgrading software on the machine.

After the upgrade, you will see the following changes:

- The folder names are changed to reflect the new product naming.
- When you opt to back up the files during the upgrade process,
  - the CentralConfig.xml file is retained in the old Workflow Management install folder with a .bak suffix.
  - the BPM2019_Backup folder is created in the old Workflow Management install path. The existing Locales, Web, and WorkflowElements folders are moved to this backup folder.
- Workflow Management services are changed to AVEVA Work Tasks services.
- WorkflowManagementTemp and Workflow Management Cache files are removed from the machine.

Note: Customizations to files in Locales, Web, and WorkflowElements folders need to be applied again to the newly created files after upgrade.

To upgrade to AVEVA Work Tasks 2020

1. Run *AVEVA Work Tasks 2020.exe*.

   The installer splash screen, and then the Prerequisites installation screen appears.
2. Install the Prerequisites.

   Note: The system restarts after installing .NET Framework 4.8.
3. Click **Next** after installing the prerequisites.

   An alert message "*The installation process slows down when the installer backs up the Locales, Web, and WorkflowElements folders. Do you want the installer to take the backup?*" appears.
4. Perform one of the following:
   - Click **Yes** if you want the installer to take the backup.
   - Click **No** if you do not want the installer to take the backup.
   - If no option is selected, the installer waits for 20 seconds, and then continues the installation process without backing up the files.
5. Click **Next**.

   The system prompts you for the restart.
6. Click **OK** if you want to restart the machine.
7. After restart, on the InstallShield Wizard Completed dialog box, click **Finish**.
8. The **Upgrade Manager** automatically gets launched and performs the following tasks:
   - Stops the services and Application Pools.
   - Deletes the **WorkflowManagementTemp** and **Workflow Management Cache** files from the machine.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For more information on how to run the Upgrade Manager, see Run Upgrade Manager in the Administrator Guide.

You may be prompted to restart the machine. After the machine is restarted, launch the Upgrade Manager manually using AVEVAWorkTasksUpgradeManager.exe available in the <Installation Path>.

Notes:

- After the upgrade completes, the Work Tasks services and Application Pools start automatically.

- Work Tasks services names are changed. The new service name gets reflected after the restart of the machine. We recommend restarting the machine after running the Upgrade Manager.

- If the images do not appear properly in the upgraded **Enterprise Console** menu, ensure to update the images from the **Navigation Details** list.

a. Point to **admin**, and then click **Repository Settings**. The Repository Settings page appears.

b. Go to **Advanced Settings**, click **Manage Navigation List**. The Manage List appears.

c. Edit the **ImageName** in the NavigationLink Details window of the required Navigation Link item.

- The Default folder for the Attachments Storage Path in the Repository Settings is changed from C:\ProgramData\Workflow Management to C:\ProgramData\AVEVA Work Tasks.

- Clearing the browser cache on user machines where Enterprise Console is accessed is recommended for getting the (js, css, images, etc. files) changes to take effect after the upgrade. For more information on how to clear cache, see the Cache Refresh Settings section in the Administrator Guide.

Upgrading to AVEVA Work Tasks 2020 Update 1

You can upgrade to AVEVA Work Tasks 2020 Update 1 from AVEVA Work Tasks 2020. However, if you want to upgrade from versions earlier than AVEVA Work Tasks 2020, you must first upgrade to AVEVA Work Tasks 2020 and then to AVEVA Work Tasks 2020 Update 1.

When you upgrade the product to AVEVA Work Tasks 2020 Update 1, the earlier product version is uninstalled and all the files are removed. Therefore, we recommend that you back up the files before upgrading the product.

Make sure that the user has appropriate rights for installing and upgrading software on the machine.

When you upgrade the product to AVEVA Work Tasks 2020 Update 1, the existing files are overwritten by the latest files. Therefore, we recommend that you back up the files before upgrading the product.

Make sure that the user has appropriate rights for installing and upgrading software on the machine.

Note: Customizations to files in Locales, Web, and WorkflowElements folders need to be applied again to the newly created files after upgrade.

To upgrade to AVEVA Work Tasks 2020 Update 1

1. Run *AVEVA Work Tasks 2020 Update 1.exe*.

   The installer splash screen, and then the Prerequisites installation screen appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Install the Prerequisites.

   Note: The system restarts after installing .NET Framework 4.8.

3. Click **Next** after installing the prerequisites.

   An alert message "*The installation process slows down when the installer backs up the Locales, Web, and WorkflowElements folders. Do you want the installer to take the backup?*" appears.

4. Perform one of the following:
   - Click **Yes** if you want the installer to take the backup.
   - Click **No** if you do not want the installer to take the backup.
   - If no option is selected, the installer waits for 20 seconds, and then continues the installation process without backing up the files.
5. Click **Next**.

   The system prompts you for the restart.

6. Click **OK** if you want to restart the machine.
7. After restart, on the InstallShield Wizard Completed dialog box, click **Finish**.
8. The **Upgrade Manager** automatically gets launched and performs the following tasks:
   - Opens **Cryptography Settings** window. Provide the **Passphrase** to proceed.
   - Stops the services and Application Pools.
   - Deletes the **WorkTasksTemp** and **Work Tasks Cache** files from the machine.

     For more information on how to run the Upgrade Manager, see Run Upgrade Manager in the Administrator Guide.

     You may be prompted to restart the machine. After the machine is restarted, launch the Upgrade Manager manually using AVEVAWorkTasksUpgradeManager.exe available in the <Installation Path>.

Notes:

- During AVEVA Work Tasks installation, if **Platform Common Service** and **Common Licensing API Framework** prompt to restart the machine, you can ignore and do a restart after completing installation. If you opt to restart the machine, the installation does not resume after a restart. You will have to run the installer to continue the installation.

- After the upgrade completes, the Work Tasks services and Application Pools start automatically.

- Clearing the browser cache on user machines where Enterprise Console is accessed is recommended for getting the (js, css, images, etc. files) changes to take effect after the upgrade. For more information on how to clear cache, see the Cache Refresh Settings section in the Administrator Guide.

- When you upgrade the product for the existing Web APIs in Azure Active Directory and AVEVA Connect repositories, the authentication mode will change to **Service to Service Access Token** authentication. For AVEVA Identity Manager, the authentication mode will be **User Access Token** authentication.

- If User Access Token is configured for the Web API and associated with Scheduler Event Association, the workflow fails.

- Deprecated SQL Server Management Objects (SMO) will not be removed from the system automatically.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Upgrading to AVEVA Work Tasks 2020 Update 2

You can upgrade to AVEVA Work Tasks 2020 Update 2 from AVEVA Work Tasks 2020 and AVEVA Work Tasks 2020 Update 1. However, if you want to upgrade from versions earlier than AVEVA Work Tasks 2020, you must first upgrade to AVEVA Work Tasks 2020 and then to AVEVA Work Tasks 2020 Update 2.

Make sure that the user has appropriate rights for installing and upgrading software on the machine.

When you upgrade the product to AVEVA Work Tasks 2020 Update 2, the existing files are overwritten by the latest files. Therefore, we recommend that you back up the files before upgrading the product.

Make sure that the user has appropriate rights for installing and upgrading software on the machine.

Note: Customizations to files in Locales, Web, and WorkflowElements folders need to be applied again to the newly created files after upgrade.

To upgrade to AVEVA Work Tasks 2020 Update 2

1. Run *AVEVA Work Tasks 2020 Update 2.exe*.

   The installer splash screen, and then the Prerequisites installation screen appears.
2. Install the Prerequisites.

   Note: The system restarts after installing .NET Framework 4.8.
3. Click **Next** after installing the prerequisites.

   An alert message "*The installation process slows down when the installer backs up the Locales, Web, and WorkflowElements folders. Do you want the installer to take the backup?*" appears.
4. Perform one of the following:
   - Click **Yes** if you want the installer to take the backup.
   - Click **No** if you do not want the installer to take the backup.
   - If no option is selected, the installer waits for 20 seconds, and then continues the installation process without backing up the files.
5. Click **Next**.

   The system prompts you for the restart.
6. Click **OK** if you want to restart the machine.
7. After restart, on the InstallShield Wizard Completed dialog box, click **Finish**.
8. The **Upgrade Manager** automatically gets launched and performs the following tasks:
   - Opens **Cryptography Settings** window. Provide the **Passphrase** to proceed.
   - Stops the services and Application Pools.
   - Deletes the **WorkTasksTemp** and **Work Tasks Cache** files from the machine.

     For more information on how to run the Upgrade Manager, see Run Upgrade Manager in the Administrator Guide.

     You may be prompted to restart the machine. After the machine is restarted, launch the Upgrade Manager manually using AVEVAWorkTasksUpgradeManager.exe available in the <Installation Path>.

Notes:

- During AVEVA Work Tasks installation, if **Platform Common Service** and **AVEVA Enterprise Licensing API Framework** prompt to restart the machine, you can ignore and do a restart after completing installation. If you

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

opt to restart the machine, the installation does not resume after a restart. You will have to run the installer to continue the installation.

- After the upgrade completes, the Work Tasks services and Application Pools start automatically.

- Clearing the browser cache on user machines where Enterprise Console is accessed is recommended for getting the (js, css, images, etc. files) changes to take effect after the upgrade. For more information on how to clear cache, see the Cache Refresh Settings section in the Administrator Guide.

- Deprecated SQL Server Management Objects (SMO) will not be removed from the system automatically.

- SAP Connection list will not be present after upgrade. Workflows with SAP activities will be replaced with dummy activities and the workflows may not get executed properly.

Upgrading to AVEVA Work Tasks 2023

Upgrading Scenarios

You can upgrade to AVEVA Work Tasks 2023 from the following versions:

- AVEVA Work Tasks 2020
- AVEVA Work Tasks 2020 Update 1
- AVEVA Work Tasks 2020 Update 2

However, if you want to upgrade from versions earlier than AVEVA Work Tasks 2020, you must first upgrade to AVEVA Work Tasks 2020, and then to AVEVA Work Tasks 2023.

Make sure that the user has appropriate rights for installing and upgrading software on the machine.

Upgrading to AVEVA Work Tasks 2023

When you upgrade the product to AVEVA Work Tasks 2023, the earlier product version is uninstalled and all the files are removed. Therefore, we recommend that you back up the files before upgrading the product.

After the upgrade, you will see the following changes:

- When you opt to back up the files during the upgrade process,
  - the CentralConfig.xml file is retained in the old AVEVA Work Tasks install folder with a .bak suffix.
  - the AWT2020_Backup folder is created in the old AVEVA Work Tasks install path. The existing Locales, Web, and WorkflowElements folders are moved to this backup folder.
- AVEVA WorkTasksTemp and AVEVA Work Tasks Cache files are removed from the machine.

Note:
- Customizations to files in Locales, Web, and WorkflowElements folders need to be applied again to the newly created files after an upgrade.
- The SAP Connection list will not be present after an upgrade. The workflows with SAP activities will be replaced with dummy activities, and the workflows may not get executed properly.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to Upgrade

1. Run *AVEVA Work Tasks 2023.exe* to upgrade to AVEVA Work Tasks 2023.

   The installer splash screen, and then the Prerequisites installation screen appears.
2. Install the Prerequisites.

   Note: The system restarts after installing .NET Framework 4.8.
3. Click **Next** after installing the prerequisites.
4. Perform one of the following:
   - Click **Yes** if you want the installer to take the backup.
   - Click **No** if you do not want the installer to take the backup.

     The installer waits 20 seconds for any input and continues the installation process without backing up the files if no user input is provided.
5. Click **Next**.

   The system prompts you for the restart.
6. Click **OK** if you want to restart the machine.
7. After restart, on the InstallShield Wizard Completed dialog box, click **Finish**.
8. The **Upgrade Manager** automatically gets launched and performs the following tasks:
   - Opens **Cryptography Settings** window. Provide the **Passphrase** to proceed.

     Note: While upgrading the product from AVEVA Work Tasks 2020 to AVEVA Work Tasks 2023, you are prompted to provide a **Passphrase**.
     But the Passphrase is not needed while upgrading from
     - AVEVA Work Tasks 2020 Update 1 to AVEVA Work Tasks 2023
     - AVEVA Work Tasks 2020 Update 2 to AVEVA Work Tasks 2023
   - Stops the services and Application Pools.
   - Deletes the **WorkTasksTemp** and **Work Tasks Cache** files from the machine.

     For more information on how to run the Upgrade Manager, see Run Upgrade Manager in the Administrator Guide.

     You may be prompted to restart the machine. After the machine is restarted, launch the Upgrade Manager manually using AVEVAWorkTasksUpgradeManager.exe available in the <Installation Path>.

Notes:
- During AVEVA Work Tasks upgrade, if **Operations Control Logger, Platform Common Service**, and **Common Licensing API Framework** prompt to restart the machine, you can ignore and do a restart after completing installation. If you opt to restart the machine, the installation does not resume after a restart. You will have to run the installer to continue the installation.

- After the upgrade completes, the Work Tasks services and Application Pools start automatically.

- It is recommended to clear the browser cache on the machines where Enterprise Console site is accessed to get the (js, css, images, etc. files) changes to take effect after the upgrade. For more information on how to clear cache, see the Cache Refresh Settings section in the Administrator Guide.

- Deprecated SQL Server Management Objects (SMO) will not be removed from the system automatically.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Upgrading to AVEVA Work Tasks 2023 SP1

Upgrading Scenarios

You can upgrade the product to AVEVA Work Tasks 2023 SP1 from AVEVA Work Tasks 2023. However, if you want to upgrade from versions earlier than AVEVA Work Tasks 2023, you must first upgrade to AVEVA Work Tasks 2023 and then to AVEVA Work Tasks 2023 SP1.

When you upgrade the product to AVEVA Work Tasks 2023 SP1 from versions earlier than AVEVA Work Tasks 2023, you cannot proceed with the upgrade process. The prerequisites for AVEVA Work Tasks 2023 SP1 get installed which are not compatible with the earlier product version's prerequisites, and you cannot revert the changes. Hence, before you start upgrading, backup all the files of the installed product version in case you want to revert. Also, make sure earlier versions of the product are upgraded to AVEVA Work Tasks 2023 before upgrading to AVEVA Work Tasks 2023 SP1.

Upgrading to AVEVA Work Tasks 2023 SP1

The upgrade will overwrite all files. Therefore, we recommend that you back up the files before upgrading the product.

Make sure that the user has appropriate rights for installing and upgrading software on the machine.

Note:
- Customizations to files in Locales, Web, and WorkflowElements folders need to be applied again to the newly created files after an upgrade.

Steps to Upgrade

1. Run *AVEVA Work Tasks 2023 SP1.exe* to upgrade to AVEVA Work Tasks 2023 SP1.

   The installer splash screen, and then the Prerequisites installation screen appears.
2. Install the Prerequisites.

   Note: The system restarts after installing .NET Framework 4.8.
3. Click **Next** after installing the prerequisites.
4. Perform one of the following:
   - Click **Yes** if you want the installer to take the backup.
   - Click **No** if you do not want the installer to take the backup.

     The installer waits 20 seconds for any input and continues the installation process without backing up the files if no user input is provided.
5. Click **Next**.

   The system prompts you for the restart.
6. Click **OK** if you want to restart the machine.
7. After restart, on the InstallShield Wizard Completed dialog box, click **Finish**.
8. The **Upgrade Manager** automatically gets launched and performs the following tasks:
   - Stops the services and Application Pools.
   - Deletes the **WorkTasksTemp** and **Work Tasks Cache** files from the machine.

     For more information on how to run the Upgrade Manager, see Run Upgrade Manager in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Administrator Guide.

You may be prompted to restart the machine. After the machine is restarted, launch the Upgrade Manager manually using AVEVAWorkTasksUpgradeManager.exe available in the <Installation Path>.

Notes:
- During AVEVA Work Tasks upgrade, if **Platform Common Service** and **AVEVA Enterprise Licensing** prompt to restart the machine, you can ignore and do a restart after completing installation. If you opt to restart the machine, the installation does not resume after a restart. You will have to run the installer to continue the installation.

- After the upgrade completes, the Work Tasks services and Application Pools start automatically.

- Clearing the browser cache on user machines where Enterprise Console site is accessed is recommended for getting the (js, css, images, etc. files) changes to take effect after the upgrade. For more information on how to clear cache, see the Cache Refresh Settings section in the Administrator Guide.

- Deprecated SQL Server Management Objects (SMO) will not be removed from the system automatically.

Run Upgrade Manager

After completing the upgrade, the **AVEVA Work Tasks Upgrade Wizard** appears automatically.

1. In the **AVEVA Work Tasks** Upgrade Wizard, click **Next**.
   The **AVEVA Work Tasks** Upgrade Progress window appears.
2. Click **Update** to continue with the upgrade.
3. Click **Close**.
   A message appears to restart the computer.
4. Restart the computer for the changes to take effect.
5. Check if the IIS is running. If not, manually restart the IIS.

Upgrading AVEVA Work Tasks in Silent Mode

You can upgrade to AVEVA Work Tasks 2023 SP1 in silent mode from AVEVA Work Tasks 2023. However, if you want to upgrade from versions earlier than AVEVA Work Tasks 2023, you must first upgrade to AVEVA Work Tasks 2023 and then to AVEVA Work Tasks 2023 SP1.

Prerequisites to Upgrade AVEVA Work Tasks in Silent Mode

1. Stop all the services.
2. Stop IIS.
3. Delete the WorkTasksTemp and Cache files from the computer.
4. Ensure that all the prerequisites are installed. For more information about installation prerequisites, see Installation Prerequisites.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure License Server Details in Silent Mode Using XML

After installing or upgrading AVEVA Work Tasks in silent mode, you have to configure the license server.

For more information about parameters in the **WWSKBPMConfig.xml** configuration file, see Configure in Silent Mode Using XML.

1. Open the **WTLogger Console**.
2. Browse to the "Bin" folder, which is available in the path where the product is installed.
3. Open the **WWSKBPMConfig.xml** file.
4. In the XML file, pass the values for the following parameters.
   - **ConfigurationType**: Enter the value as **AVEVALicensing**.
   - **LicenseServerName**: Enter the name of the License Server.
   - **LicenseServerPort**: Enter the port number used by the License Server.
5. Save the XML file.
6. Run the **AVEVAWorkTasksConfigurator.exe**.

   The configuration progress is displayed in the **WTLogger Console**.

   Note: If there are any errors while updating the license server details in silent mode, these exceptions are not logged in the **WTLogger Console**.

Upgrading in Silent Mode Using Installer

You can upgrade to AVEVA Work Tasks 2023 SP1 in silent mode from the following version:

- AVEVA Work Tasks 2023

Upgrade Procedure

The syntax for running the AVEVA Work Tasks upgrade installer from the command line is as follows:

<Installer EXE Name>.exe /S /v/qn

For example: AVEVA Work Tasks 2023 SP1.exe /S /v/qn

where:

- /S denotes silent install
- /v denotes pass arguments to Msiexec (Basic MSI and InstallScript MSI projects). The /v option is used to pass command line options and values of public properties to Msiexec.exe.
- /qn denotes an msiexec switch indicating to run with no user interface (UI).

This command string takes the default values for parameters, such as user name, installation location, and company name.

This command considers the default settings for the upgrade. The installer automatically detects the current installation path and then performs the upgrade in the same path. For example: If the product was installed in a custom path, then the upgrade also happens in the same custom path.

The progress of the silent mode upgrade will not be displayed in the Command Prompt. However, the upgrade continues in the background. If you want the Command Prompt to be in waiting mode till the upgrade is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

complete, execute the following command in a **.bat** file.

```
echo off
AVEVA Work Tasks 2023 SP1.exe /S /v/qn
if %ERRORLEVEL% EQU 0 (
echo AVEVA Work Tasks 2023 SP1 upgrade installation completed successfully.
)
```

Note: In scenarios where the AVEVA Work Tasks binaries/files are already open or are being used by any application, then the server might be automatically restarted after silent upgrade.

Upgrade in Silent Mode

Server Machine

You can upgrade to AVEVA Work Tasks 2023 SP1 in silent mode using the installer. After the assemblies and related resources are upgraded, you have to upgrade the Farm database, Repository database, and all the required configurations in silent mode, using **AVEVAWorkTasksSilentUpgradeManager.exe**.

Client Machine

You can upgrade the client machines in silent mode using **AVEVAWorkTasksSilentUpgradeManager.exe**. This EXE automatically detects whether the machine is a server or a client and then does the changes accordingly. After the upgrade is complete, Central Configuration and Enterprise Console are upgraded with the latest changes.

For more information about the return results for Upgrade, see Return Results for AVEVA Work Tasks Silent Upgrade.

To upgrade in Silent Mode

1. Open the Command Prompt in the Administrative mode.
2. Enter the path for the "Bin" folder, which is available in the path where the product is installed, and then press **Enter**.

   For example: CD <AVEVA Work Tasks 2023 SP1 Install Path>\Bin
3. On the Command Prompt, type **AVEVAWorkTasksSilentUpgradeManager.exe**.

   The progress of the upgrade is displayed in the **WTLogger Console**.

Return Results for AVEVA Work Tasks Silent Upgrade

1. Open a Notepad file and add the following script.

   ```
   echo off
   cd <AVEVA Work Tasks 2023 SP1 Installpath>\Bin\AVEVAWorkTasksSilentUpgradeManager.exe
   echo %ERRORLEVEL%
   ```
2. Enter the appropriate value for **<AVEVA Work Tasks Install path>**.

   For example: *C:\Program files\AVEVA\Work Tasks\Bin*
3. Save the file with **.bat** extension.
4. Open the Command Prompt in the Administrative mode.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Type the path where the Batch file is saved, and then press **Enter**.
6. Type the name of the Batch file, and then press **Enter**.

   The return result of the **AVEVAWorkTasksSilentUpgradeManager.exe** is stored in the **%ERRORLEVEL%** variable. The return result will be one of the following:

   - **0** - If the upgrade procedure is successful.
   - **1** - If the upgrade procedure failed.
   - **2** - If multiple instances of **AVEVAWorkTasksSilentUpgradeManager.exe** are running.

Use an Existing Website

1. In Services/Global Settings page, select **Custom Settings**.

   By default, for custom settings, Website, Central Configuration, and Enterprise Console nodes will not be checked.
2. Select the above mentioned tree nodes, and then click **Next**.

   The Global Settings page appears.
3. Click **OK**.

   The Central Configuration Settings window appears.
4. Click the icon against **Create at** text box in the Central Configuration Settings page.

   The IIS structure screen appears.
5. Select the desired S\site and click **OK**.

   The selected URL with the desired port number appears in the Central Configuration Settings Page.
6. Click **OK**.

   The Enterprise Console Settings page appears.
7. Click the icon against **Create at** text box.

   The IIS Structure gets displayed.
8. Select the desired site and click **OK**.

   The selected URL with the desired port number appears in the Enterprise Console Settings Page.
9. Click **Save**, so that Enterprise Console will get added to the Existing Enterprise Consoles grid and then click OK button.

Best Practices and Guidelines

This section provides information on the guidelines and the best practices to be followed while using AVEVA Work Tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Best Practices for Securing AVEVA Work Tasks

Introduction

Security is a core functional requirement that protects critical information from accidental or deliberate theft, leakage, integrity compromise, and deletion.

This document provides the best practices, which will help you define your security practices, and set of security policies and processes for your organization. This will ensure that the data and assets are protected while using AVEVA Work Tasks.

- Securing Application Websites
- Securing Data
- Securing Ports
- Securing using Users and Roles
- Securing Application Configuration
- Securing Webservers
- Managing List Data

Securing Application Websites

Use of https

Central Configuration and Enterprise Console of AVEVA Work Tasks are web-based applications.

- Central Configuration is meant only for administration purposes. However, it can be accessed by any user, if it is not protected. Therefore, Central Configuration needs to be protected by securing the sites.
- Enterprise Console is meant to be used by various users. However, it can also be accessed through internet. Therefore, Enterprise Console needs to be protected.
- Recommend use of Standard SSL certificate instead of self-signed SSL certificate.
- AVEVA Work Tasks supports TLS 1.2 version.

Securing Sessions

- Use Auto log off to secure sessions.

Ways to Minimize Security Threats

- Ensure to use ‘https’ to host Enterprise Console.
- Ensure to use SSL certificates.
- Ensure to encrypt AppSettings in the **web.config** file for Central Configuration and Enterprise Console.

  Note: If you want to edit the values in the **web.config** file, run the **Command Prompt** as an **Administrator** and type *C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319.*

  For example,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To decrypt the value, type
`aspnet_regiis -pd "<place holder for section name>" -app "/<place holder for application name>" -site <place holder for site number>`

To encrypt the value, type,
`aspnet_regiis -pe "<place holder for section name>" -app "/<place holder for application name>" -site <place holder for site number> -prov "DpapiProtectedConfigurationProvider"`

where,
`<place holder for section name>` is section of the **web.config** file to be encrypted or decrypted
`<place holder for application name>` is application name (e.g. CentralConfig, EnterpriseConsole)
`<place holder for site number>` is site number to which AVEVA Work Tasks is bound to (e.g. site number is 2 for Enterprise Console)

For more information, refer Protected Configuration Provider.

Securing Data

Eliminating Database Security Threats

AVEVA Work Tasks is a highly database dependent application. The back end database is SQL Server.

- Ensure to protect the database with proper security measures.
- Keep periodic backups for any eventuality.

Ways to Minimize Security Threats on Database

- Protect the database server with limited access.
- Ensure to limit the database access rights only to the authorized users.
- Provide only required rights to the maintenance team.

Securing Ports

Eliminating Threat on Ports

AVEVA Work Tasks application depends on certain ports for communication. System uses TCP / UDP for communication. Packets of data move from or towards these ports.

Hackers are constantly trying to discover new ways to connect to machines so that they can install backdoors for later re-entry, trojans to collect financial details, or botnet clients.

There are different kinds of threats such as denial of service, men in the middle attack, and so on. The entry point for each of these attacks is a port.

Therefore, protecting these ports without affecting the functionality is the responsibility of the user.

Ways to Minimize Threat on Ports

- There are multiple ways to minimize threats on ports. These details are not in the scope of this document. It is recommended to refer to the industry standard mechanisms to minimize the threats on ports.

Note: For more information about the ports used by the system, refer the ‘Setting up Ports’ section in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Administrator Guide.

Securing Access using Users and Roles

Eliminating Unauthorized Access

AVEVA Work Tasks provides authorization to secure data on every level. You can protect data with proper authorization.

Forms customization through scripting can introduce security vulnerabilities if the End-Users who author those scripts are unfamiliar with Secure Coding principles or are acting maliciously. Forms Designer permission role should only be assigned to users who are trusted not to abuse the system and are familiar with secure coding principles.

Ways to Secure Access

- Provide only accurate access to different users based on their usage.
- Provide role-based access to features.

Note: For Security-related information, refer the 'Security Framework' section in the *User Guide* .

Securing Application Configuration

Eliminating Threats on Application Configuration

Farm Configuration is the base configuration for AVEVA Work Tasks. Ensure that the system and the access to configuration are protected, by providing access to the Farm Configuration only to Administrators.

Ways to Protect Farm

- Ensure that the system can be access only by authorized users with Administrative privileges.

Ways to Protect Enterprise Console

- Ensure use of Active Directory provider or AIM configured with appropriate policy instead of AVEVA Work Tasks Repository provider.

Securing Webservers

To improve the AVEVA Work Tasks WebServers security, disable the weak ciphers and protocols .

You can use the tool IIS Crypto for enabling/disabling the protocols and weak ciphers. You need to restart the webserver after making the changes.

It is recommended to enable the following protocol and ciphers:

- Protocol - TLS 1.2 for server and client
- Ciphers - AES 128, AES 256

  Note: Assess the system for any impact on the other running applications before you enable/disable the protocols and ciphers.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Managing List Data

Eliminating threats on List Data

AVEVA Work Tasks provides List infrastructure to store data. Sensitive data can also be stored in the Lists. However, protecting this sensitive data is the responsibility of the organization and is important based on the business scenario.

- Ensure to protect sensitive and confidential information using adequate measures.

Ways to protect sensitive information

- Ensure to use password characters to show sensitive information.
- Enable audit trail to track repudiation.

Enhancing Security for Service Bus of Notification Bus

You can enable enhanced security for Azure Connections to enforce additional security at the service bus level.

For more information about enabling enhanced security, see Enabling Enhanced Security.

Conclusion

Ensure that the organization takes adequate measures to:

- Protect against information loss or leakage.
- Secure data, resources, and application.

Key Points

Ensure the following:

- Use https to host enterprise console.
- Protect the database with proper security measures.
- Provide role-based access to features.
- Protect the ports.
- Protect the system where services are running.

Enhancing Security of AVEVA Work Tasks Client Service running as Local System

AVEVA Work Tasks Client Service runs as a Local System and has higher privileges. It is used for:

- Quickflow configuration and deployment
- Service monitoring
- Package installation

To secure and run Client Service with the least privileges, you need to elevate the permissions through security settings for the Client Service.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To elevate the Client Service to run on Network Service,

1. Run the below command in the Command Prompt:

   ```
   netsh http add urlacl url=http://machinename:8863/ user="NT AUTHORITY\NETWORK SERVICE"
   ```

2. Restart the Client Service.

Note: You cannot configure and deploy Quickflow, perform Service Monitoring (or start other services), import and install the package if the Client Service is running as Network Service.

Improper Access Control List (ACL) in Custom Installation

In case AVEVA Work Tasks is installed in a custom path, the installation folder will inherit the access control restrictions of Program Files folder.

As a best practice to maintain security,

- Use the default path for installation
- Set ACL in case custom path is used for installation

Best Practices for Maintaining AVEVA Work Tasks Databases

A database must be maintained well in order to achieve optimum performance and reliability and high availability. A Database Professional has to perform the required maintenance tasks to achieve this. This document gives a brief of essential maintenance tasks to be performed for AVEVA Work Tasks software's database.

Note: It is highly recommended that a Database Professional performs the tasks mentioned in this document. The purpose of the document is to provide guidelines for maintenance activities. Essential maintenance tasks may vary based on specific implementation.

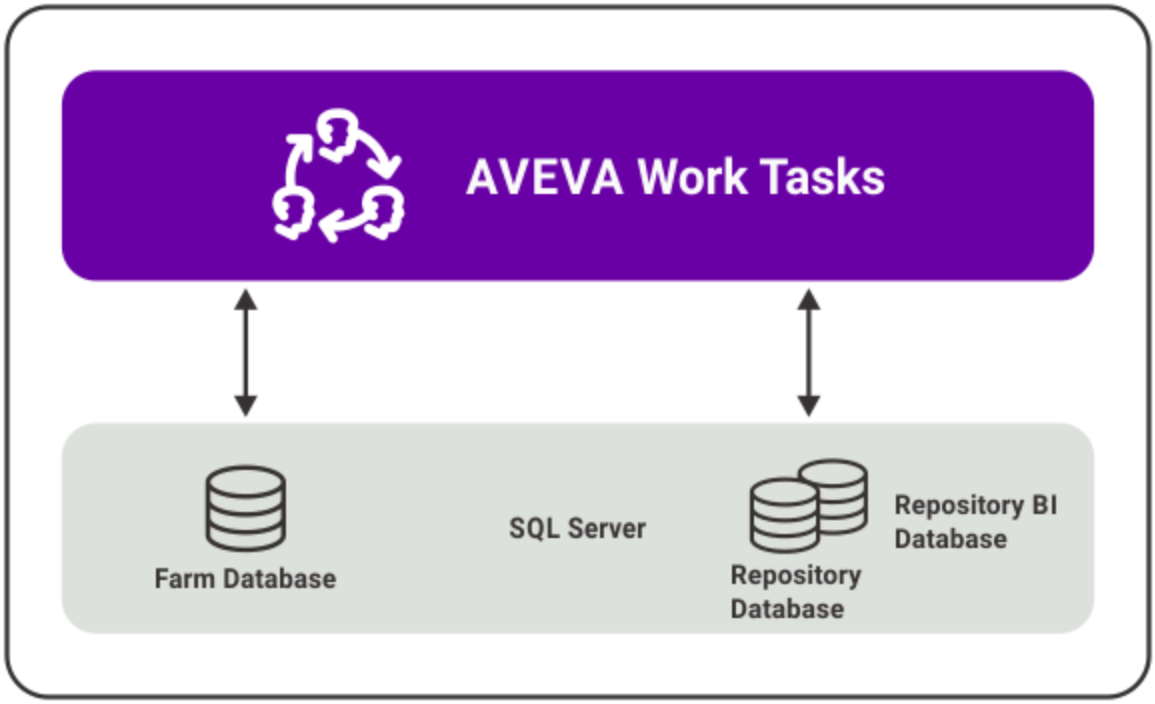
Databases in AVEVA Work Tasks

AVEVA Work Tasks uses the following databases.

- Farm Database in which the Farm Configuration settings are stored.
- Repository Database with tables pertaining to each repository.

The following figure displays the databases used by AVEVA Work Tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Maintenance Plan

Database maintenance includes a defined set of proactive tasks that a Database Professional needs to perform on a periodic basis to help ensure that their databases perform optimally and maintain high availability. In many cases, the individual maintenance steps taken might not, by themselves, seem all that important, but the accumulative effect of performing database maintenance tasks can be substantial.

Note: Under normal circumstances, it is recommended to perform the database maintenance tasks depending on the environment, intensity of transactions, and regulatory and organizational requirements.

The following sections describe some of the very important database maintenance tasks, which are categorized into high availability and optimum performance.

High Availability and Disaster Recovery

Backup and Restore

Protection of data is of prime importance to any organization. To protect the database, a proper disaster recovery plan and backup process should be in place. Based on the daily transactions count, full backup should be taken periodically, i.e. either Weekly or Daily.

Note: It is recommended to schedule a full backup and a transaction backup periodically, based on the transactions.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Managing MDF Files

An MDF file is a primary file to store the data of the database. Maintenance of the MDF file is important to get optimum performance and availability of the database. Ideally, the MDF file should be placed in an IO Device with the best IO throughput.

Pre-size the database based on the size requirement. For more information about database sizing, see Guidelines for Sizing Database.

You can also find out the size requirement by performing the following:

1. Watching the database growth for multiple weeks/months
2. Calculating the size for a longer period of time
3. Setting the database size in advance

The two major benefits of pre-sizing databases are:

- By pre-sizing databases, SQL Server would not have to depend on autogrowth to grow the database. This ensures that the performance is not affected when autogrowth occurs.
- Each autogrowth can contribute to physical file growth.

Managing LDF Files

An LDF file stores the database transactions that allow a database to be restored back to a specific point in time on any eventuality. SQL Server transaction log files are NTFS files that have a file extension of .ldf. A database can have multiple log files that reside on one or more drives. The growth of the file is based on transactions on to the database and maintenance strategies.

The backup strategy affects the growth of the LDF files. The LDF file size might increase when the database Recovery Mode is “Full”. Therefore, to reduce the LDF file size, it is recommended to schedule a Full Backup followed by incremental backups or to shrink the file size.

A Database Professional can achieve this using internal database commands such as **DBCC SHRINKFILE**.

Note: Ensure not to place the LDF file in the same drive as that of the MDF file.

Checking for Corruption

Corrupt databases are the worst nightmare of any Database professional. In any environment, from small business to enterprise, the compromise of integrity and availability of the data can constitute a business emergency or financial and reputation loss. Therefore, early detection and prevention of data corruption is essential, which can be achieved by meticulous planning. Ideally, the DBCC CHECKDB command should be run periodically to monitor the database corruption.

Data Corruption Detection

Following is a standard checklist used for data corruption and detection:

- List the “Page Verify Setting” for all of your Production and System Databases.
- Do you regularly and proactively check for potential database corruption.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- What method(s) do you use to identify potential database corruption.
- How often do you check for potential database corruption in each Production and System Database.
- List the average time taken to perform database corruption detection per database.
- Have you documented what you would do if data corruption is detected.

Eliminating Physical File Fragmentation

When the OS writes a file to disk, if continuous clusters are not available, the content is written elsewhere on the disk.

- When a file is stored in a non-contiguous manner on disk, the file is considered to be physically fragmented.
- Physical file fragmentation can contribute to an additional load on your I/O subsystem and reduce I/O performance because the disk has to work harder (thrash) to read and write data.

Ways to minimize physical file fragmentation:

- Ensure there is no physical file fragmentation before creating new database and log files.
- Pre-size MDF and LDF files instead of allowing autogrowth to automatically size files. This prevents physical file fragmentation.
- Regularly defragment arrays used for backups as they can become fragmented over time.

Physical File Fragmentation

The following need to be checked periodically.

- List the physical file fragmentation for every array with MDF, LDF, BAK, and TRN files.
- Monitor File Fragmentation
- Regularly perform Physical File Defragmentation
- What tool do you use to perform physical file fragmentation.
- Before creating or growing MDF or LDF files, do you check for Physical Fragmentation first.

Managing Indexes

AVEVA Work Tasks is a highly database dependent application. AVEVA Work Tasks has all the required indexes for ensuring optimum performance of the Workflow Engine. Since AVEVA Work Tasks can be extended by customization, the customized component may degrade the performance of AVEVA Work Tasks and its components. Therefore, it is advised that the indexing needs of the customized components and the extent of usage are looked into, from the performance perspective.

Custom tables are the tables created by the user for specific needs such as new custom Lists, custom application specific tables, and so on. For example, if an application uses AVEVA Work Tasks List infrastructure, then the users have to create indexes based on the custom tables.

You can use the database engine tuning advisor to arrive at the index requirement. Understanding the characteristics of your database, queries, and data columns can help you design optimal indexes.

It is also recommended to perform the following actions periodically:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Detect Index Fragmentation
- Fix Index Fragmentation
- Reorganize or Rebuild Indexes
- Create missing index
- Filter out unused indexes

Note: It is recommended to add only the required indexes, and avoid adding too many indexes.

Maintaining Statistics

SQL Server collects statistical information about indexes and column data stored in the database. These statistics are used by the SQL Server Query Optimizer to choose the most efficient plan for performing INSERT, SELECT, DELETE or UPDATE queries.

The statistical maintenance functionality of SQL Server, AutoStat, may generate unwanted overhead on a production system by performing one of the following actions:

- Initiating statistical updates during heavy production periods.
- Initiating an excessively high number of UPDATE STATISTICS processes at a given point in time.

Note: It is recommended to analyze the pros and cons before setting the statistics to the best desired mode and maintain it based on the requirement.

Statistics Maintenance

Following is a standard checklist used for index and statistics maintenance:

- What method(s) do you use to maintain Index and Column Statistics.
- How often do you run jobs to maintain Index and Column Statistics.
- If you use the REBUILD Method to Defragment Indexes, do you also update Statistics.
- If you use the REORGANIZE Method to Defragment Indexes, do you also update Statistics.
- If you update Statistics independent of Index Defragmentation, list the average time taken per database.
- Do you have “AUTO_UPDATE_STATISTICS_ASYNC” turned “On”. If Yes, list those databases.
- Do you use Filtered Statistics. If Yes, are they documented (SQL Server 2008).

Archiving Data

Increase in the AVEVA Work Tasks Database size inversely affects the performance of workflow execution. The database size can be reduced by archiving the data.

Note: It is recommended to archive the data based on year, financial year, or so on depending on business and regulatory requirements.

AVEVA Work Tasks archival script is available for SQL Server. Check with customer support for more details and for script.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Table Partitioning

SQL Server supports table and index partitioning. The data of partitioned tables and indexes is divided into units that can be spread across more than one file group in a database. The data is partitioned horizontally, so that groups of rows are mapped into individual partitions. All partitions of a single index or table must reside in the same database. The table or index is treated as a single logical entity when queries or updates are performed on the DB. This will provide better performance.

A remarkable performance improvement of AVEVA Work Tasks engine can be achieved in the SQL Server Enterprise versions using the SQL Server Table Partition feature.

Conclusion

These are the best practices for maintaining AVEVA Work Tasks Database. The implementation can be customized as per the requirement. Periodic database maintenance by a Database Professional or Expert is required to get optimum performance, reliability, and availability of the database.

The following are the key points for maintaining AVEVA Work Tasks Databases:

- Pre-size database based on transactions.
- Plan periodic database and log file backups.
- Manage indexes and maintain statistics.
- Archive database for exponential data growth if performance is affected.
- Follow a periodic and systematic backup strategy.
- Eliminate index fragmentation.
- Detect corrupted database pages.

Best Practice for Disabling Active Directory Users

Disable the user account in the Active Directory when the user is no longer associated with an organization or process. This makes the user account unusable until the user returns. As a best practice, the user accounts are often disabled for a period before deleting when the user leaves the organization.

Consider a scenario when Active Directory provider or AIM is used in the product for work item assignments, administrative, or system user accounts. The system considers only enabled users for Active Directory search and if the user information is used, the system may fail to access certain features. So before disabling a user in Active Directory, replace the user with an alternate user. To replace user with alternate user, see Replace User topic.

Best Practices for Handling Different Regional/Cultural Settings

The best practice to avoid issues related to different regional/cultural settings is to use the same collation and settings for all the instances used if you intended to develop content to be rolled out globally.

Consider a scenario where you have a centralized engineering team located in one region/country and the production sites globally. When you configure the Development or Test servers with specific regional and culture settings and roll out the solution to production site servers where the regional/culture setting is specific with that region, you may have some issues while working with AVEVA Work Tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Best Practice for Handling Cross Site Request Forgery in Custom Application

To handle Cross Site Request Forgery in custom applications, add the below code if you are using AVEVA Work Tasks 2020 or a later version:

- In case the webapp is not anonymous,

  ```
  AntiForgeryConfig.UniqueClaimTypeIdentifier = ClaimTypes.NameIdentifier;
  ```

- In the aspx page, under the form tag,

  ```
  <%=System.Web.Helpers.AntiForgery.GetHtml()%>
  ```

Guidelines for Sizing Database

This document explains the database growth pattern of AVEVA Work Tasks databases. It is intended to help the user understand about the disk space, memory, and processor speed requirements for best performance of AVEVA Work Tasks.

Note: The performance characteristics presented in this document are not to be interpreted as a benchmark measurement for all the supporting systems. Only empirical testing on the target system can provide accurate data. The actual database size, memory used, I/O required is based on the respective implementation, concurrent user access, and so on.

Overview

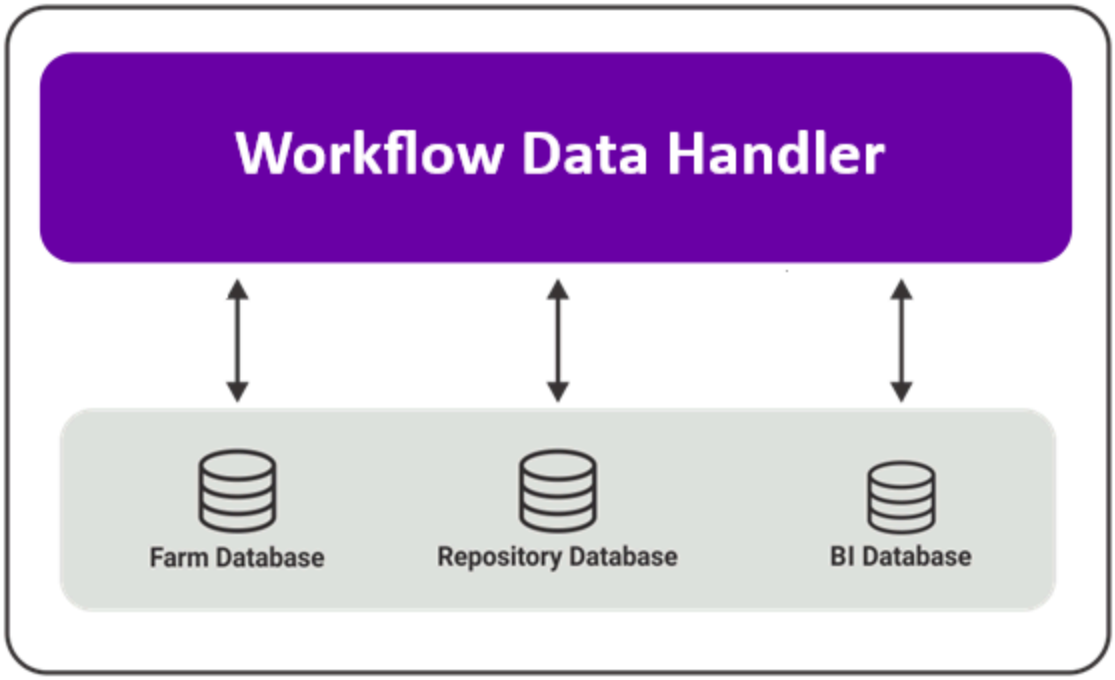
AVEVA Work Tasks supports different versions of Microsoft SQL Server to store operational data. The analysis for database sizing is done on Microsoft SQL Server 2008 (SP3) - 10.0.5512.0 (X64).

By default, AVEVA Work Tasks databases are set to grow in an "Auto growth" mode as SQL Server default settings. Based on the best practices recommended by Microsoft and database experts, you should set the size of the repository database depending on the transactions and growth pattern.

A database should be sized appropriately so that the database does not need to auto grow, but should be configured by the user when required. It is recommended to configure the **File Growth** value to fix growth, based on the database growth pattern.

A desirable approach is to have a separate disk for repository and farm database.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Note: It is highly recommended that a Database Professional performs the tasks mentioned in this document. The purpose of the document is to provide guidelines for sizing AVEVA Work Tasks database. Essential maintenance tasks may vary based on specific implementation.

Farm Database

The farm database is a configuration database, basically used to map repositories, support Enterprise Server configuration, and so on. The database may grow when there is a massive usage of communication activities, such as email and SMS.

Repository Database

The repository database is a database for the repository related data. All the operational data required for the application exists in the database to keep the database healthy. Data such as various activities related information, the list data, forms definition, and workflow definitions is stored in the repository database. The database growth always relates to the features, such as list and queue, used by the application.

BI Database

The BI database is a placeholder for future development on business intelligence reports and does not occupy any space.

Parameters that affect the performance of an application include the environment, various databases involved, kind of workflow, and number and types of activities. For database sizing, the prospective data is captured to show the database growth pattern based on a typical workflow having 72 activities with a combination of engine and human activities. The information of the environment on which the data is captured along with the workflow used is included.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Database Growth Pattern

Table Level Growth

Table level growth is purely based on the features and Workflow used. Based on the sample workflow, the table level record growth is as follows:

Table Name	Number of Records per 5000 instances	Number of Records per Instance	Remarks
SKChannelWorkItem	20000	4	
SKHWSActivity	20000	4	
SKWorkItem	20000	4	
SKWorkItemTransaction	20000	4	
SWActionEventLog	500000	100	
SWActionEventLong	500000	100	
SWExecute	5000	1	
SWExecutionDetails	360000	72	There are 72 activities for the workflow. Each activity has a record.
SWLong	5000	1	
SWVariables	45000	9	

Table Size

Farm Database

The details regarding the table size help in arriving at the maximum database growth pattern.

To calculate the storage needs, multiply the average row length and the number of records per month that will be entered into the database. This provides the details of the amount of storage required by each table. If a table has any variable length string fields, add 4 bytes to the average row length to account for overhead, which SQL Server requires when maintaining the table.

The following table shows different tables of the repository database and their size:

Table_name	Byte_Length
SKActivityNameConcurrency	1220
SKActivityTypeConcurrency	720

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Table_name	Byte_Length
SKAddInProviders	1983
SKASServerStatus	1057
SKAzureServiceBusSettings	2776
SKBizTalkMessages	77
SKBizTalkRules	1109
SKCacheInfo	224
SKConcurrentThreadPools	520
SKConcurrentUsers	32
SKDataSource	1106
SKDBScriptVersionStamp	204
SKDeviceRegistration	3610
SKEventServicePorts	548
SKFarm	1244
SKInMemoryActivityCount	1020
SKNotificationMessage	1261
SKProduct	553
SKRepository	5357
SKRepositoryMapping	72
SKServer	1861
SKService	1276
SKServiceMapping	1336
SKSiteUserGroups	1124
SKWebSite	3762
SWProcessedMails	16
SWProcessedSMS	16
SWSMSException	20

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Table_name	Byte_Length
SWSMSFilters	2740
SWSMSs	48
SWSMTPMailException	20
SWSMTPMailFilters	2740
SWSMTPMails	48
SWTaskBizTalkMsgTimeout	542
SWTaskEmailTimeout	1042
SWTaskSMSTimeout	1030
All_Tables	42314

Repository Database

Default table size of the repository database is as follows and this may help to arrive at maximum database growth pattern.

Table_name	Byte_Length in KB
SKAddInProviders	1983
SKAddInProviderSettings	1057
SKAlternateActorRules	1175
SKBamAlertManager	1040
SKBAMConditionWorkItemFields	808
SKBAMConditionWorkItems	596
SKBamDashboard	1042
SKBamMenu	3640
SKBAMResourceWorkAnalysis	636
SKBAMTypeofWork	24814
SKBAMUniqueTypeOfWork	2121
SKBAMWebService	6516
SKCalendar	1114

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Table_name	Byte_Length in KB
SKCalendarDay	764
SKCalendarWeekDays	560
SKChannelWorkItem	6820
SKChildWorkflows	2454
SKCrossDomainUserRoles	2524
SKDBScriptVersionStamp	204
SKDeviceSubscription	2403
SKEAssemblyList	8093
SKEAuditTrail	5283
SKEBizTalkConnections	5539
SKEConnection	10335
SKECustomControls	2478
SKEDynamicLists	5508
SKEEventAssociations	11103
SKEEventProviders	5961
SKEFolderList	9653
SKEForm	6007
SKEFormDataGridViews	2592
SKEFormDefinition	1526
SKEGridConfigurations	3247
SKEImpersonationUsers	6527
SKEListDefinition	1907
SKEListFormAssociation	1551
SKEListItemDataNV	99599
SKEListItemDataTemp	1064
SkeListItemWorkflow	1341

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Table_name	Byte_Length in KB
SKEListOperations	3284
SKEListTemplateTable	1411
SKEListViews	2245
SKEListWIPItems	673
SKELookup	6223
SKENavigationDetails	8238
SKEntity	1272
SKEntityDataSourceProvider	1756
SKEPopupTemp	56
SKERightsControlProviders	1604
SKERightsListMapping	548
SKEscalation	1825
SKESecurityGroup	5558
SKESecurityRights	6727
SKESGResourceMapping	5667
SKESGRightsMapping	628
SKESGViewRightsMapping	617
SKESSRSConnection	4412
SKESSRSReports	3639
SKESSRSTemplate	3411
SKEUsers	6663
SKEventBindings	16436
SKEventServicePorts	548
SKEVirtualTable	1773
SKEWebPartLayouts	1056
SKEWebParts	1566

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Table_name	Byte_Length in KB
SKEWebPartTemplates	2036
SKEWorkflow	5507
SKEWorkflowDefinition	1348
SKEWPApplications	1552
SKEWPPaths	1056
SKEWPPersonalizationAllUsers	40
SKEWPPersonalizationPerUser	72
SKEWPRoles	1568
SKEWPSchemaVersions	513
SKEWPUsers	1097
SKEWPUsersInRoles	32
SKExecutionInstanceData	544
SKExternalConnectionDetails	2587
SKExternalConnections	1711
SKFavouritesURI	1816
SKFWFolderList	5707
SKHWSActivity	7881
SKInvokeWebServices	3141
SKListDefinitionVersionStamp	20
SKListTables	1025
SKMultipleTimeoutWarning	833
SKPackage	5507
SKPackageAttachment	2682
SKPackageConfigurations	1148
SKPackageDataTemp	540
SKPackageDetails	977

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Table_name	Byte_Length in KB
SKPackageHistory	3361
SKPackageTemplate	5507
SKPackageTemplateAttachment	2683
SKPackageTemplateDetails	977
SKPersistedException	3912
SKPersonalization	640
SKQueue	1389
SKQueueAccessRights	1042
SKQueueParticipants	588
SKReplaceUserLog	4623
SKRibbonBarXmlInfo	1832
SKRoleRefresh	559
SKScheduler	1661
SKSchedulerHistory	9728
SKSchedulerProcessInfo	1128
SKSPAlertedWorkflows	5062
SKSPWebSettings	3216
SKTaskStatusTypes	1341
SKTempFormPrint	556
SKTemplates	2833
SKTemplatesLong	48
SKTimeZones	1016
SKUserReports	1733
SKUserReportsDetails	1227
SKVirtualActor	6126
SKVirtualRole	5386

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Table_name	Byte_Length in KB
SKWorkflowCorrection	1101
SKWorkflowEvents	882
SKWorkItem	1521
SKWorkItemCounter	1268
SKWorkItemData	540
SKWorkItemFields	4392
SKWorkItemFieldsDefinition	2986
SKWorkItemFieldTypeCounts	804
SKWorkItemTransaction	946
SKWorkItemTransactionTypes	522
SKWorkItemView	1288
SWActionEventLog	20
SWActionEventLong	20
SWCode	9648
SWCustomTimerDetails	598
SWExecute	1276
SWExecutionDetails	946
SWFECount	1032
SWFileMonitor	721
SWLong	20
SWProduction	1020
SWProductionLong	20
SWRaw	20
SWReportSets	1424
SWScriptCode	2248
SWStaging	1076

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

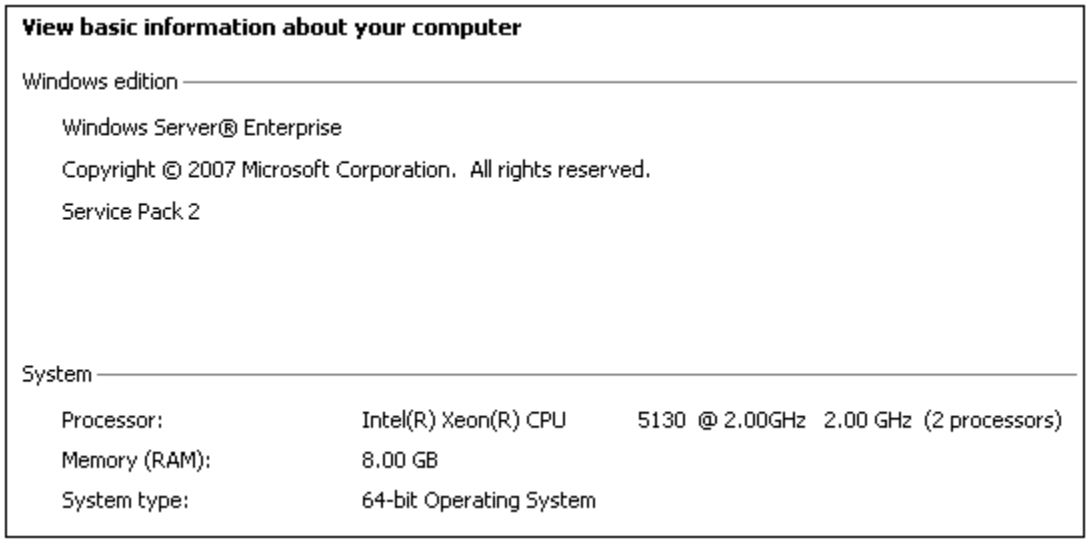
Table_name	Byte_Length in KB
SWStagingLong	20
SWTaskForResuming	532
SWTaskList	1332
SWVariables	292
SWWebServiceDetails	4648
SWWorkflowEventLog	20
All_Tables	504837

Bytes	KB	MB
504837	493.0049	0.48145

Environment

The environment used to capture the database sizing related information is the Enterprise server with two web server nodes, two workflow engine servers, and an SQL Server mentioned as Database Server. The system configuration of the environment is as follows:

Web Server Configuration

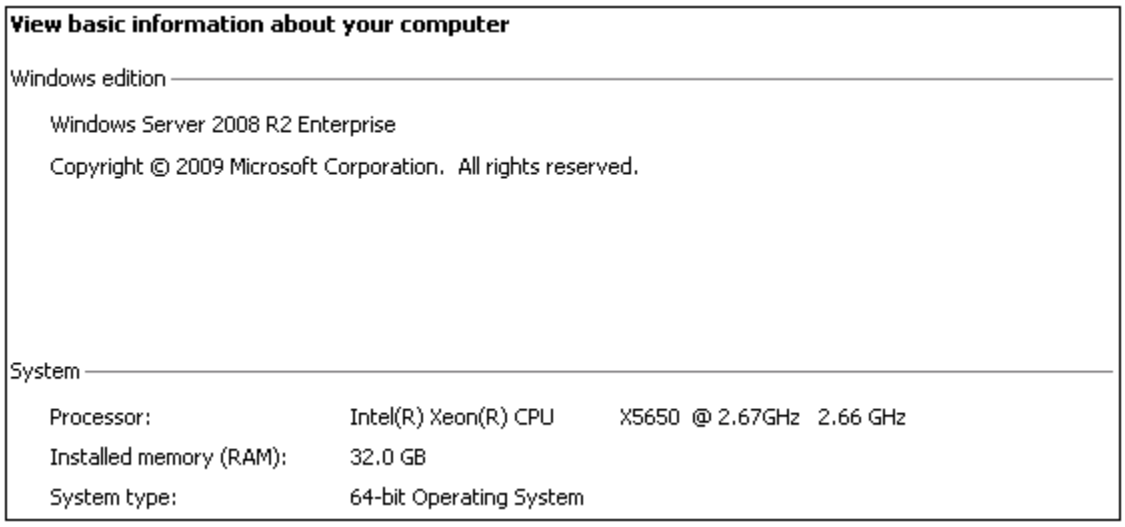


Engine Servers

The configurations for the engine servers are same as that of the Web Server.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Database Server

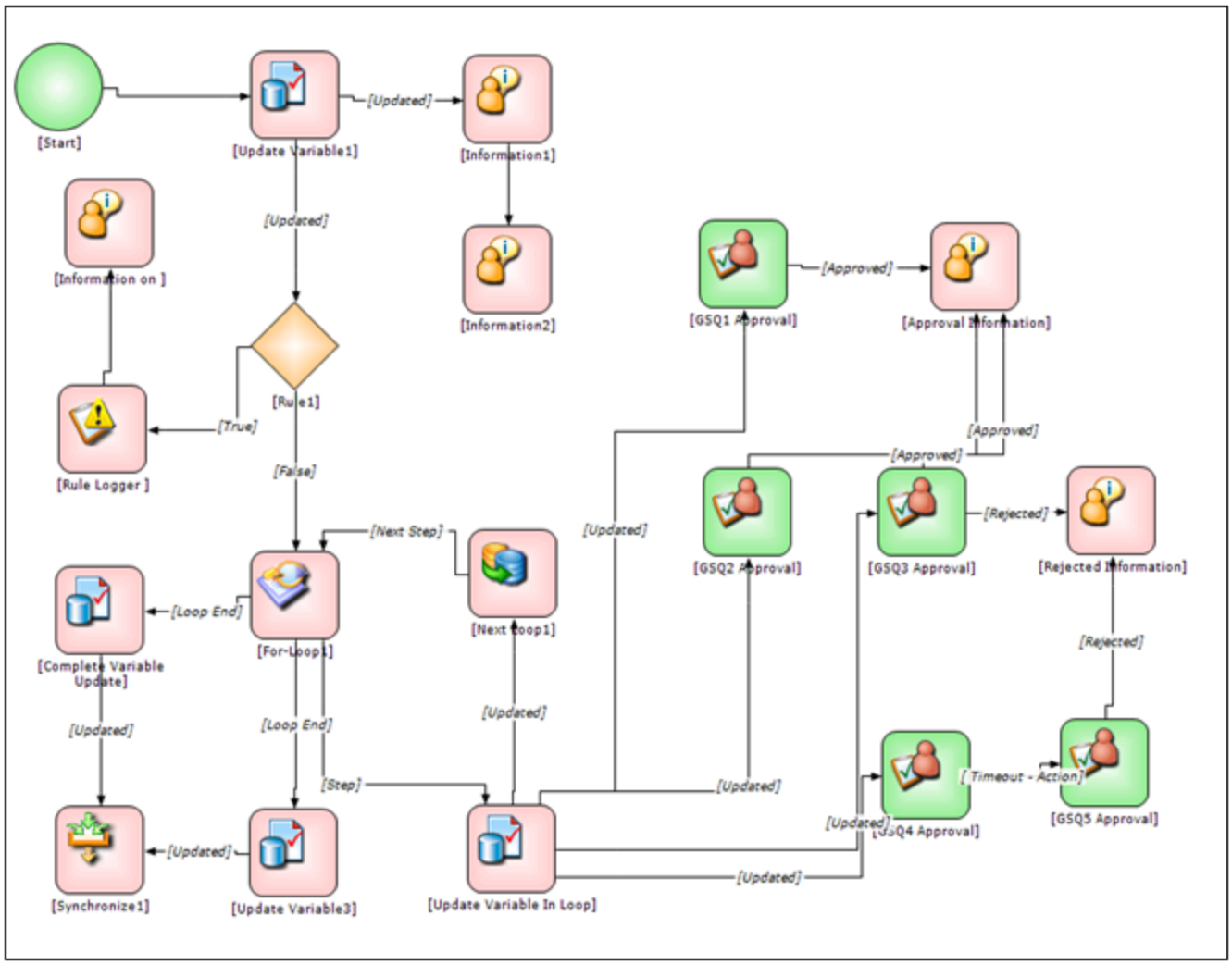


Server Startup Time	Aug 16 2013 4:30PM
Server Instance Name	SKELTARND2
Product Version	10.0.5512.0
Edition	Developer Edition (64-bit)
Windows ProcessID	2232
Active Sessions	1
Active Transactions	9
Active Databases	50
Total Server Memory (KB)	29810440

Workflow Used

The database growth pattern is captured based on the performance test workflows, which include 72 activities altogether and 5 queues with a combination of automatic dispatch pattern and manual pick. In parallel, work item tasks are assigned to the queues. The workflow diagram is as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



LogFile Size

Every SQL Server database has a transaction log, which records all the transactions and the database modifications made by each transaction. The transaction log must be truncated on a regular basis to keep it from filling up. However, some factors can delay log truncation. Therefore, monitoring log size is important. This can be achieved by logging some operations minimally to reduce their impact on transaction log size.

The transaction log is a critical component of the database. If there is a system failure, the transaction log is required for database recovery to bring the database back to a consistent state.

Note: Ensure that the transaction log is never deleted or moved without having a clear understanding of its effects.

For more information regarding management of log files, it is recommended to consult a Database Professional.

Number of Workflow instances	Log file Growth MB
433000	9697
438000	9697

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Number of Workflow instances	Log file Growth MB
443000	9697
448000	9697
453000	9697
458000	9697
463000	9697
468000	9697
473000	9697
478000	9697
483000	9697

TempDB

A temporary database is not directly used by AVEVA Work Tasks. However, the SQL Server internally uses this database for operations, such as sorting indexing. Therefore, it is recommended to maintain an ideal space for this database, to facilitate the above mentioned SQL server internal operations.

A default installation of any SQL Server edition will create a TempDB database with an 8MB data file and a 1MB transaction log file. For a lot of SQL Server installations, these file sizes are not sufficient. However, they are configured to auto grow by 10% as needed.

The auto grow feature enables a more hands-off approach to maintaining many SQL Server installations. However, this approach is not recommended because the database storage files cannot be used while they are autogrowing. This can lead to fragmentation of the files on the hard disk; thereby, leading to poor performance.

This recommendation is applicable to any SQL Server database, but it is even more relevant for the TempDB. When you restart your SQL Server instance, the TempDB is re-created (files will be reused if they already exist) and sized to the value specified in the database properties. The default values are 8MB for the data file and 1MB for the log file.

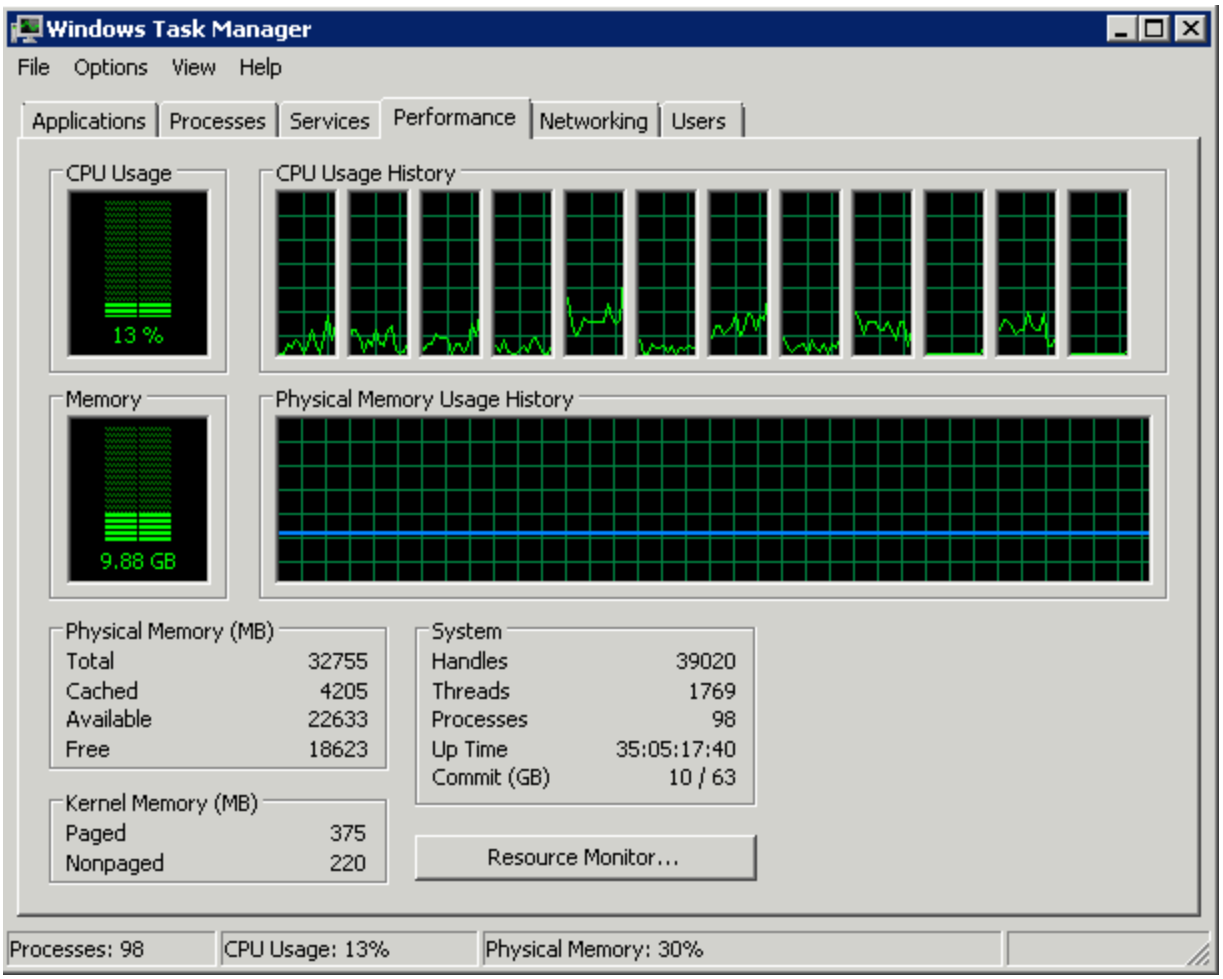
Note: The I/O operations are the most time consuming processes and the TempDb is being used exclusively by SQL Server for most of its operations. Therefore, it is recommended to have a separate disk for the TempDb than the operational database. A managed growth gives better performance. Please consult the Database Professional for the desired settings.

CPU Utilization

The CPU utilization on the database server is based on the queries or updates considered as a part of the workflow engine database activities. The CPU utilization varies with user interaction taken place as part of the UI activities. Various other parameters such as file fragmentation, I/O overheads, and so on also affect the CPU utilization. For more realistic information on the implementation, please consult a Database Professional.

Graph shows the CPU utilization.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



I/O Statistics

Reading and writing to the disk is the heart of any database management system. Input/Output (I/O) performance is critical for databases and can make or break an application. In many environments, I/O is the main bottleneck for the database performance. The main task of I/O is to retrieve and to store records within files. The I/O performance can be improved by analyzing the specific issues by a Database Professional. Various SQL Server reports are available for analyzing the I/O related performance. The I/O performance purely depends on the specific environment. Thus, providing a snapshot of the I/O statistics may not be helpful.

http://blogs.msdn.com/b/sqljourney/archive/2013/06/04/how-to-troubleshooting-sql-server-i-o-bottlenecks.aspx

http://www.quest.com/backstage/images/promotions/SQLServer-Perfmonance-Poster.pdf

Performance Matrix

Following are some of the important parameters to consider for database sizing. Ideally, an implementation should provide the following performance. Count values above the following are not advisable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Area	Maximum Value	Type	Remarks
Number of Unattended Work items across queues	60000	Supported	Ideally there should not be this many items
Number of not acted Items in a Queue	15000	Supported	In the test environ 4 queues populated with 15000 items each
Execution Time in ms for an action on 100GB data	65	Threshold	Execution time in ms for an activity
Time taken in ms for 72 Actions on 100GB	4750	Threshold	Execution time in ms for a workflow instance
Number of execution of Workflow instance at any point of time on a Server	32	Threshold	At any point of time on any server, 32 instances of workflow can be running

References

For further information, refer the following links.

http://blogs.technet.com/b/patricg/archive/2007/10/18/sql-server-tips-n-5-tempdb-best-practices.aspx?Redirected=true

http://technet.microsoft.com/en-us/magazine/2008.08.database.aspx

Definitions

Term	Definition
Logfile	Database log file
TB	Tera Bytes
GB	Giga Bytes
MB	Mega Bytes
DB	Database

Special Notes

- Actual growth pattern will be based on the implementation.
- Set the size of the database for a better performance.
- Keep enough space in the SQL Server TempDB.
- Use 64-bit SQL server environment for better performance based on business requirement.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Actual server configuration purely depends on business requirement, such as concurrent users and workflow instances.
- Every implementation should have proper database management monitored by a Database Professional.
- I/O statistics must be monitored to check if it affects the actual performance an application.

Conclusion

The following values are approximate values, and might vary based on the implementation:

- Average Growth of Database per instance is **118.1726 KB**.
- Average Growth of Database per instance per Action is **1.641286 KB**.
- Database size requirement per day for a workflow having 10 activities and 100 instance is 1.6412 * 10 * 100 = 1641.2 KB, that is, **1.627MB** approximately.

Guidelines for Archiving Database

Introduction

Archiving and purging data from the production database helps in maintaining a healthy system, and also in keeping the archived data for future use. An archiving strategy is a plan that ensures that the correct data is archived in a correct way. Archiving without a strategy might result in poor overall system performance. The most successful archival implementations are those that are well-planned by taking several important aspects into account.

Assess your needs for archiving by answering the following questions:

- What data do we need to archive?
- When do we need to archive this data?
- How long do we need to maintain the archived data?
- Under which circumstances and how often will we need to retrieve archived data from either table archives or file archives?
- Do we require archived data that is stored in table archives to be moved to flat-file archives (i.e., do we require a multi-tier archiving strategy)?

Note: It is highly recommended that a Database Professional performs the tasks mentioned in this document. The purpose of the document is to provide guidelines for archiving AVEVA Work Tasks database. Essential maintenance tasks may vary based on specific implementation.
The Repository database and Farm database should reside on two different machines.

Benefits of Archiving

Archiving data reclaims disk space, lowers storage costs, reduces object maintenance, and improves the performance of active data.

The major benefits of archiving are interrelated. By reclaiming disk space that was consumed by inactive data, you can effectively lower your storage costs and reduce maintenance. As a result, access to your active data

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

becomes more efficient. When less data exists, utility processing requires less time; thereby, reducing the batch processing times.

In addition, lowering costs is the most important benefit of archiving your inactive data. Also, applications might experience improved performance and use less CPU, I/O, and memory resources, which results in further cost savings.

Archival Strategy for AVEVA Work Tasks Database

The Archiving Script is a database script used for removing the completed and aborted workflow data from the AVEVA Work Tasks databases. The main goal of archiving the databases is to ensure better performance by reducing the amount of data from the databases.

AVEVA Work Tasks has two databases – Repository database and Farm database.

To archive the databases, follow the steps given in:

- Archiving Repository Database
- Archiving Farm Database

The steps provided are applicable only if the product version is 4.3.4005.0 or later.

Archiving Repository Database

1. On executing the **AWTRepositoryArchivingScript.sql** script in the Repository database, the following tables and procedures are created.

Tables and Procedures	Description
SKDeleteTable	Table to store the ExecutionID and other details of the records to be deleted from the Repository database (Production database).
SKDeleteTableFor ArchiveDB	Table to store the ExecutionID and other details of the records to be deleted from the Archived database.
proc_SKInsertIntoSK DeleteTable	Procedure to insert ExecutionID, ProductionID, Application, Name, Data, LocalVariablesId, and ExecutionDetailsID into the SKDeleteTable table. **Note:** Records related to workflows with InternalStatus "FN" or "AB" will be inserted into the **SKDeleteTable** table.
proc_SKDeleteIn Batches	Procedure to delete all records of a table in batches, where the size of each batch size is 50000 records.
proc_SKDeleteFrom Production	Procedure to delete the Repository database records, and to update the 'Isdeleted' field of the **SKDeleteTable** table to 1, only if the delete was

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Tables and Procedures	Description
	successful. The procedure will delete workflow records with InternalStatus "FN" or "AB".
proc_SKDeleteFrom ArchivedDB	Procedure to delete workflow records, which have InternalStatus other than "FN" or "AB", from the Archived database. **Note:** Ensure that this procedure is executed only in the Archived database.
proc_SKSKDelete TableInArchiveDB	Procedure to insert ExecutionID, ProductionID, Application, Name, Data, LocalVariablesId, and ExecutionDetailsID into the **SKDeleteTableForArchiveDB** table. **Note:** - Records related to workflows with InternalStatus other than "FN" or "AB" will be inserted into the **SKDeleteTableForArchiveDB** table. - Ensure that this procedure is executed only in the Archived database.

Store the workflow data to be deleted from the Repository database in the **SKDeleteTable** table.

Execute the **proc_SKInsertIntoSKDeleteTable** procedure on the Repository database.

```
Exec proc_SKInsertIntoSKDeleteTable 'fromdate', 'todate', 'applicationname',
'executionstartid', 'executionendid'
```

Pass the parameters for the above procedure based on the workflow records to be deleted from the Repository database. Workflow data with InternalStatus ‘FN’ and ‘AB’ will be stored in the database. For example, to delete workflow records executed for a period of 2 years, execute the procedure as follows:

```
Exec proc_SKInsertIntoSKDeleteTable '01-01-2013', '12-31-2014', NULL, NULL, NULL
```

The above query will insert records between 01-01-2013 and 12-31-2014 with InternalStatus ‘FN’ and ‘AB’ into the SKDeleteTable table.

The records to be deleted are stored with ExecutionID, ProductionID, Application, Name, Data, LocalVariablesId, ExecutionDetailsID, and Isdeleted details in the SKDeleteTable table. Initially, the ‘Isdeleted’ field is ‘0’ for all the records to be deleted.

2. Store the workflow data to be deleted from the Archived database in the **SKDeleteTableFor ArchiveDB** table.

   Execute the **proc_SKSKDeleteTableInArchiveDB** procedure on the Repository database.

   ```
   Exec proc_SKSKDeleteTableInArchiveDB
   ```

3. Back up the Repository database.
4. Restore the above backup to new database. This can be the archived Repository database.
5. Delete the completed and aborted workflow instances related data from the Repository database tables.

   Execute the **proc_SKDeleteFromProduction** procedure on the Repository database to delete all the records similar to the ones that exist in the **SKDeleteTable** table. Workflow records with InternalStatus ‘FN’ or ‘AB’ will be deleted from database.

   Execute the following procedure:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Exec proc_SKDeleteFromProduction
```

After the workflow records are deleted, the **Isdeleted** field of these records in the **SKDeleteTable** table is updated to '1'.

To delete 36118960 records, the system took approximately 39:25 minutes in a database of size 221121.63 MB. This time requirement will vary based on the system performance.

6. Delete the records other than that for the completed and aborted workflow from the Archived database.

   Execute the **proc_SKDeleteFromArchivedDB** procedure on the Archived database to delete all the Repository database table records that are available in the **SKDeleteTableForArchiveDB** table.

   Workflow records with InternalStatus other than 'FN' or 'AB' are deleted from the Archived database tables.

   Execute the following procedure :

   ```
   Exec proc_SKDeleteFromArchivedDB
   ```

Refer to the scenario explained in the Application Scenario section.

Archiving Farm Database

Note:
- Do not execute step 1 and step 2, if you have already executed while archiving Repository database and continue with step 3.
- If you want to archive only Farm database and do not want to archive the Repository database, then follow the archiving procedure from step 1.

1. Execute the **AWTRepositoryArchivingScript.sql** script in the Repository database.
2. Execute the **proc_SKInsertIntoSKDeleteTable** procedure in the Repository database. Refer the execution scripts given in Archiving Repository Database.
3. Execute the **proc_SKInsertIntoFarmTemptable** procedure on the Repository database.

   Note: Execute above procedure if the SQL Server system has restarted after step 2.

4. Execute the **AWTFarmArchivingScript.sql** script in the Farm database.

   The following tables and procedures are created on executing the above steps.

Tables and Procedures	Description
SKDeleteTableFarm	Table to store the ExecutionID and other details of the records to be deleted from the Farm database (Production database).
proc_SKDeleteFrom ProductionFarm	Procedure to delete the Farm database records, and to update the 'Isdeleted' field of the **SKDeleteTable** table to 1, only if the delete was successful. The procedure will delete workflow records with InternalStatus "FN" or "AB".
proc_SKDeleteFrom ArchivedFarm	Procedure to delete workflow records, which have InternalStatus other than "FN" or "AB", from the Archived database. **Note:** Ensure that this procedure is executed only in

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Tables and Procedures	Description
	the Archived database.
proc_SKDelete InBatches	Procedure to delete all records of a table in batches, where the size of each batch size is 50000 records.

5. Backup the Farm database.
6. Restore the above backup to new database. This can be the archived Farm database.
7. Execute the **proc_SKDeleteFromProductionFarm** procedure in Production Farm database.
8. Execute the **proc_SKDeleteFromArchivedFarm** procedure in Archived Farm database.

Archival Strategy for Custom Tables

The users are expected to create archiving script for custom tables.

Archival Strategy for Migrated Database

The users are expected to migrate to the appropriate versions of AVEVA Work Tasks databases before using the Archival Strategy.

Viewing BAM Reports for the Archived Database

- Use the **Attach Datasource** feature provided in Central Configuration to get the Archived database repositories.

  To attach existing archived repository datasource, do the following steps:

  1. Take a backup of the Archived database and recover this database in the Database Server. For example, the restored database name is BackedUpdatabase.
  2. Create a new Datasource and use the same database created in step a.

     Example: Datasource is created with the name ‘BackedupDatasource’. The restored database, named ‘BackedUpdatabase’, is used to create this datasource.
  3. Create a new repository for the datasource created in step b.
     a. Select the **Attach Datasource** check box. Select the datasource created in Step b. This will list all the repositories of the selected data source.
     b. Provide a name to the repository and continue with creating the repository. Ensure that the Resource Provider for the new repository is same as the Resource Provider of the archived repository selected from the drop-down list.

  Example: Select the **Attach Datasource** option. Select the BackedupDatasource from the **Repository DataSource** drop-down list. This will list all repositories of the BackedupDatasource. Select the repository which needs to be copied and then continue with repository creation. Ensure that the Resource Provider for the new repository is same as that of the Resource Provider selected for the archived repository.

  You can view the BAM reports for the newly created repositories from the Enterprise Console.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Application Scenario

Example Scenario

The following scenario explains the steps for the Archival Strategy.

Scenario Description

Consider a scenario where all workflow records related to employee leave details are stored in the database named Employees Leave Report. This database contains large amount of data, which affects the performance. Hence, all the completed and aborted workflow related data, which is not related to current financial year (2015), must be deleted from the database. A record of the deleted data must also be maintained in the Archived database.

This ensures that the Employees Leave Report database will have all the workflow related data for the current financial year (2015), and also all the executing or sleeping workflow related data for the previous financial years.

The Archived database will have only completed and aborted workflow instance records related to the previous financial years.

Scenario Application Details

The following application and workflow names are considered to describe this scenario effectively.

- Application Name/Repository Name: EmployeeRepo
- Workflow Name: LeaveReportWF

Consider that the **SWExecute** table contains the following entries. In this example, only few fields of the **SWExecute** table are considered.

ID	Production ID	Application	Name	InternalStatus	RequestedOn
1	1	EmployeeRepo	LeaveReportWF	FN	7/9/2013 17:00
2	1	EmployeeRepo	LeaveReportWF	AB	11/22/2014 11:16
3	2	EmployeeRepo	LeaveReportWF	FN	8/8/2013 19:22
4	2	EmployeeRepo	LeaveReportWF	AB	12/4/2013 9:15
5	2	EmployeeRepo	LeaveReportWF	SL	5/29/2013 15:32
6	2	EmployeeRepo	LeaveReportWF	SL	5/2/2014 6:09
7	2	EmployeeRepo	LeaveReportWF	PA	6/1/2013 20:56
8	2	EmployeeRepo	LeaveReportWF	SL	11/3/2014 14:45
9	2	EmployeeRepo	LeaveReportWF	FE	4/20/2013 10:00

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

ID	Production ID	Application	Name	InternalStatus	RequestedOn
10	2	EmployeeRepo	LeaveReportWF	FN	3/3/2015 16:35

Scenario Solution

The following section will give you a brief idea of each step in the archival strategy:

1. Run the **WSBPMArchivingScript.sql** script on the Employees Leave Report database. This will create the required tables and procedures.
2. Execute the **proc_SKInsertIntoSKDeleteTable** procedure as follows on the Employees Leave Report database to fetch the records of the previous financial years. This procedure will insert the fetched records into the **SKDeleteTable** table.

   ```
   Exec proc_SKInsertIntoSKDeleteTable '01-01-2013', '12-31-2014', NULL, NULL, NULL
   ```

   The above query will insert records between 01-01-2013 and 12-31-2014 with InternalStatus ‘FN’ and ‘AB’ into the SKDeleteTable table.

   The records to be deleted are stored with ExecutionID, ProductionID, Application, Name, Data, LocalVariablesId, ExecutionDetailsID, and Isdeleted details in the **SKDeleteTable** table. Initially, the ‘Isdeleted’ field is ‘0’ for all the records to be deleted.
3. Execute the **proc_SKSKDeleteTableInArchiveDB** procedure as follows on the Employees Leave Report database to fetch the records of the previous financial years. This procedure will insert the fetched records into the **SKDeleteTableForArchiveDB** table.

   ```
   Exec proc_SKSKDeleteTableInArchiveDB
   ```
4. Take a full backup of the Employees Leave Report database and rename it as ‘Backup Employees Leave Report’.

   Backup Employees Leave Report Database

ID	Production ID	Application	Name	InternalStatus	RequestedOn
1	1	EmployeeRepo	LeaveReportWF	FN	7/9/2013 17:00
2	1	EmployeeRepo	LeaveReportWF	AB	11/22/2014 11:16
3	2	EmployeeRepo	LeaveReportWF	FN	8/8/2013 19:22
4	2	EmployeeRepo	LeaveReportWF	AB	12/4/2013 9:15
5	2	EmployeeRepo	LeaveReportWF	SL	5/29/2013 15:32
6	2	EmployeeRepo	LeaveReportWF	SL	5/2/2014 6:09
7	2	EmployeeRepo	LeaveReportWF	PA	6/1/2013 20:56
8	2	EmployeeRepo	LeaveReportWF	SL	11/3/2014 14:45

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

ID	Production ID	Application	Name	InternalStatus	RequestedOn
9	2	EmployeeRepo	LeaveReportWF	FE	4/20/2013 10:00
10	2	EmployeeRepo	LeaveReportWF	FN	3/3/2015 16:35

5. Execute the **proc_SKDeleteFromProduction** procedure on the Employees Leave Report database to delete all the workflow records from the Repository database tables with InternalStatus 'FN' or 'AB'. After the workflow records are deleted, the 'Isdeleted' field in the **SKDeleteTable** table is updated to '1' for all deleted records. Now the 'Employees Leave Report' database contains the following records:

   Employees Leave Report Database

ID	Production ID	Application	Name	InternalStatus	RequestedOn
5	2	EmployeeRepo	LeaveReportWF	SL	5/29/2013 15:32
6	2	EmployeeRepo	LeaveReportWF	SL	5/2/2014 6:09
7	2	EmployeeRepo	LeaveReportWF	PA	6/1/2013 20:56
8	2	EmployeeRepo	LeaveReportWF	SL	11/3/2014 14:45
9	2	EmployeeRepo	LeaveReportWF	FE	4/20/2013 10:00
10	2	EmployeeRepo	LeaveReportWF	FN	3/3/2015 16:35

   Note: Even though the highlighted workflow instance record is in finished state, this row and the records related to it are not deleted because this record belongs to the current financial year (2015). As per the parameters passed in step 2, only the records from previous financial years need to be deleted.

6. Execute the **proc_SKDeleteFromArchivedDB** procedure in the Backup Employees Leave Report database to delete all the records except the records in the SKDeleteTableForArchiveDB. Workflow records with InternalStatus other than 'FN' or 'AB'. The value of the 'Isdeleted' field of the **SKDeleteTableForArchiveDB** table will remain as '0'.

Backup Employees Leave Report Database

ID	ProductionID	Application	Name	InternalStatus	RequestedOn
1	1	EmployeeRepo	LeaveReportWF	FN	7/9/2013 17:00
2	1	EmployeeRepo	LeaveReportWF	AB	11/22/2014 11:16
3	2	EmployeeRepo	LeaveReportWF	FN	8/8/2013 19:22

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

ID	ProductionID	Application	Name	InternalStatus	RequestedOn
4	2	EmployeeRepo	LeaveReportWF	AB	12/4/2013 9:15

Environment

The environment used for the sample exercise is an Intel Xenon computer with a 2.66 GHz processor and 32 GB RAM using Windows Server 2008 R2 Load Balanced Server.

Conclusion

After performing the above steps, the Employees Leave Report database will have data only related to the current financial year (2015) along with all the running workflow instances of the previous financial year.

The 'Backup Employees Leave Report' Archived database will have only those completed and aborted workflow instance records related to the previous financial year.

This is the method to reduce the amount of data from the production database to ensure better performance.

Guidelines for Partitioning Database

Overview

AVEVA Work Tasks stores application data in SQL Server. Important data cannot be destroyed due to analytical or legal requirements. Keeping the data in transactional database, results in alarming database growth. Though the latest database management system and hardware can handle large amount of data, it might still affect the performance of an application.

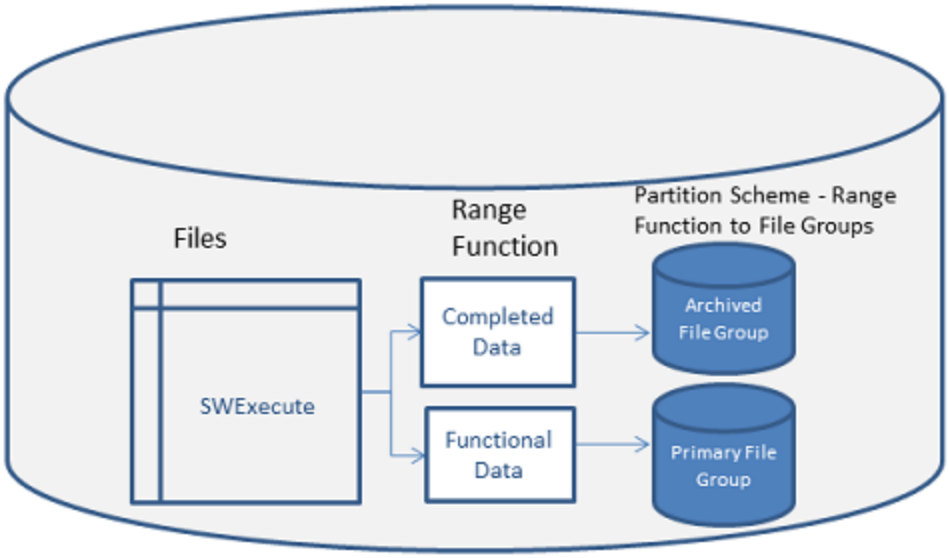
AVEVA Work Tasks can be used across different business domains. Based on the business requirement, workflows might run continuously, resulting in data table growth for a few tables. These tables are critical for the workflow engine. The alarming growth of these tables affects the performance of the application. Table partitioning capabilities provide an option to boost the performance, without affecting the other areas of the application.

The partitioning of the database is recommended for those business applications that run large number of workflow instances.

AVEVA Work Tasks uses Range Partition. Partitioning feature functions by moving data as follows:

- The completed workflow details to secondary partition.
- The running and active workflows to primary partition.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Special Note

The partition feature is available only on Microsoft SQL server versions and editions that support table partition feature. Refer to the Microsoft documentation to check the availability of the partition feature.

For more information on partitioning, see Partitioning Database.

Recommendations

- Computers with more than 16 GB RAM will require more disk space for paging, hibernation, and dump files.
- Depending on complexity and size of your database server, 30 GB or more of available hard disk space is recommended. Your hard drive for the database must have adequate available disk space to support the performance requirements of your specific deployment.

Best Practices for Features and Services

Services

- **Workflow Engine** - Workflow Engine is a Microsoft Windows service which processes the workflow actions or activities continuously. The workflow engine always connects to SQL server to pull and push data and uses thread pooling for processing the activities. Based on the system, proper thread pool settings are essential. The default, thread pool settings is 25. Refer thread pool settings <in Admin Guide>. Ensure to provide appropriate thread pool size based on the system configuration. Ensure to monitor CPU usage and Memory usage of the system <less usage of>.
- **Advance Server** - Advance server service enables load balancing in AVEVA Work Tasks. In an environment, ensure to disable the feature if the environment is not set for enterprise version.
- **Task Scheduler** - This service handles the execution of all tasks that have not taken place as scheduled (escalations). Task Scheduler does not persist any information in memory. All tasks are database persisted and are performed only when due time is reached. The Task Scheduler use a queue to process the tasks in

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

parallel. Due to queuing up of Task, you may not get accurate timeouts with seconds accuracy.

- **Client Service** - You can monitor the AVEVA Work Tasks Windows services such as, Client Monitor, Advance Server, Task Scheduler (TS), Workflow Engine (WF Engine), and Communication Service (CS).

Guidelines on using SQL AlwaysOn Availability Group

AVEVA Work Tasks supports SQL Server AlwaysOn Availability Groups for database.

For more information on setting up this feature, see:

- Microsoft documentation
- *Creating Farm Database* and *Creating Repository DataSource* topics

Guidelines for Configuring AVEVA Work Tasks in DMZ

Overview

You can setup an **Enterprise Console Web Application** with an external URL (accessible with an Internet connection) and with its content on an internal **Enterprise Console Web Server** (not connected to the Internet) hosted through an Internet Information Server (back end server). This is achieved by forwarding HTTP requests (client requests) to the Application Request Routing server (front end server), which in turn will internally route it to the back end server.

A reverse proxy in ASP.Net forwards client requests to the back end server, translates links and headers from the response received to the proxied URL, and then sends the response to the client.

Configuration

To set up this environment, configure the **Application Request Routing Front End Server**, and the **Back End Secured Server** as follows:

Configure Enterprise Web Application Proxy Server

(Application Request Routing Front End Server)

- Install Internet Information Server Extensions

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

  - Application Request Routing
  - URL Rewrite
- Configure Website on Application Request Routing Server
  - Setup Application Pool
  - Create Website
  - Enable Authentication to Anonymous
- Configure URL Rewrite
  - Setup Inbound Server
  - Setup Outbound Server
  - Setup Outbound Precondition

Configure Enterprise Console Web Server
(Back End Secured Server)

- Setup Website Authentication
- Setup Service Principal Names

Configure Application Request Routing Front End Server

Install Internet Information Server Extensions

The following Internet Information Server (IIS) extensions are prerequisites to configure the application request routing front end server:

- Application Request Routing
- URL Rewrite

Application Request Routing

Application Request Routing is an IIS extension that enables you to increase web application scalability and reliability through rule-based routing. For more information, refer to: https://www.iis.net/downloads/microsoft/application-request-routing.

Note: The latest version of Application Request Routing extension also includes the URL Rewrite extension.

URL Rewrite

URL rewrite is an IIS extension that enables you to create powerful inbound or outbound rules to implement URLs. This makes its ideal for a reverse proxy, if you only intend to forward the requests. For more information, refer to: https://www.iis.net/downloads/microsoft/url-rewrite.

Configuring Website on Application Request Routing Server

The **Application Request Routing Server** website is accessible to external users, and therefore must be secured appropriately with firewalls and other security measures.

- Create application request routing server website as follows:
- Setup an integrated application pool

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Create a website
- Setup website authentication by enabling only **Anonymous Authentication**

Configuring URL Rewrite

Add reverse proxy rules as follows:

- **Inbound Rules:** Enter IP address of the **Enterprise Console Web Server**
- **Outbound Rules:** Select **Rewrite the domain names of the links in HTTP responses**
- **From:** Enter the IP address along with the port (**8000**) of the **Enterprise Console Web Server**. For example, **10.012.123.45:800**0
- **To:** Enter the IP address or name of the **Enterprise Web Application Proxy Server** (**Application Request Routing Server**). For example, **ARRSERVER01**

Add outbound pre-conditions as follows:

- **Condition Input:** RESPONSE_CONTENT_TYPE
- **Check if input string:** Matches the Pattern
- **Pattern:** gzip

Configure Enterprise Console Web Server

Setup Website Authentication

By default, the **Enterprise Console Web Server** is set to **Windows Authentication**. Ensure the authentication has not changed.

Setup Service Principal Names

Setup two service principal names on **Enterprise Console Web Server**. The first one with the fully qualified system name and the second one with the system name. To setup **Service Principal Names**, run the **Command Prompt** as an **Administrator**, and then run the following commands:

First Service Principal Name

```
SETSPN -S HTTP/<ARR Server Name>.<Fully Qualified Domain Name> <Domain Name>\<User Name>
```

For example, if the ARR Server Name is **ARRSERVER01**, the Domain Name is **mydomain.com**, and the User Name is **myusername**, then the command is:

```
SETSPN -S HTTP/ARRSERVER01.mydomain.com mydomain\mysusername
```

Second Service Principal Name

```
SETSPN -S HTTP/<ARR Server Name> <Domain Name>\<User Name>
```

For example, if the ARR Server Name is **ARRSERVER01**, the Domain Name is **mydomain.com**, and the User Name is **myusername**, then the command is:

```
SETSPN -S HTTP/ARRSERVER01 mydomain\mysuserName
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Guidelines for Antivirus

Configure your antivirus software to prevent the files and directories listed below from being scanned. You can stop Windows Security from alerting you or blocking the program by adding the file to the exclusions list.

Note:
- Only do this for files or folders listed below. The paths listed here may be different based on your installation.
- Adding an exclusion for an unsafe program could expose your systems and data to increased risk.

ArchestrA and SMC Logger Storage folder exclusions:

- C:\ProgramData\ArchestrA\ and all subfolders
- C:\Program Files\Common Files\ArchestrA\
- C:\Program Files (x86)\Common Files\ArchestrA\
- C:\ProgramData\AVEVA Work Tasks\ and all subfolders

AVEVA Work Tasks, AVEVA License Manager and PCS folders exclusion:

- C:\Program Files\AVEVA\
- C:\Program Files (x86)\AVEVA\

Windows Temp folder exclusion:

Exclude the following folders

- C:\Windows\Temp\WorkTasksTemp

.Net Temp Folder (Could vary with .Net Version):

- C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Temporary ASP.NET Files\centralconfig
- C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Temporary ASP.NET Files\enterpriseconsole
- Custom Enterprise Console folders, if any

SQL Server Database exclusion:

Exclude the files in the SQL Server database location:

- C:\Program Files\Microsoft SQL Server\MSSQL.MSSQLSERVER\MSSQL\DATA\ (will vary by SQL Server version)

Exclude SQL Server database files within this directory with below listed types:

- .mdf
- .ldf
- .ndf

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Ports (Could vary based on configuration customization)

The following table lists the default ports used by various services while running AVEVA Work Tasks, exclude the ports as applicable:

Services	Port Type	Default Port number
Advance Server Services with Multicast IP address 225.0.0.10	multicast port	8850
	tcp port	8853
Workflow Engine	load balance server port	8853
	load balance client port	8859
	port	8855
SMTP services	smtp port	25
Communication services	listener port	8770
	pop-mail port	110
Task Scheduler services	load balance server port	8853
	load balance client port	8860
	port	8856
Service Monitor EXE	port	8866
License Server Port	port	55555
Client service	listener port	8863

Following ports used by AVEVA Work Tasks services, when the AVEVA Work Tasks Pro application is used outside Corporate Network, exclude the ports as applicable.

Services	Port Type	Default Port number
Workflow Engine	SB Over TCP	9350, 9351, 9352, 9353
Communication services	SB Over TCP	9350, 9351, 9352, 9353
Push Notification service	http port	80
	https port	443

Troubleshooting

For details on troubleshooting AVEVA Work Tasks, see the following sections:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Client Service Monitoring Errors
- Accessing Central Configuration and Enterprise Console

Prerequisites for Debugging Exceptions

There are various ways in which the user can view the details of the exceptions and warning messages that are encountered while using the product. AVEVA Work Tasks provides powerful debugging facilities which can be used to track configuration issues or exceptions during runtime.

Following are the components for debugging exceptions:

- Log Files
- Logger Console
- Event Viewer
- Email alert for Exceptions

Following topics provide extra information on settings for debugging:

- Understanding and Setting Debug Level
- Understanding log messages

Log Files

The log files are located in the <AVEVA Work Tasks installed path>\Logs (For example: *C:\Program files\AVEVA\Work Tasks\\Logs*) folder. Logs for a specific repository will be available inside the folder with repository name. Global exceptions will be logged directly under <AVEVA Work Tasks installed path>\Logs folder. The naming convention used for log files is “Current Year + Current Month + Current Day”.

Log files created under repository will contain repository name prefixed. For example, TelematicsRepository_20140213.log.

Logger Console

This is a debugging tool which can be used to watch the messages logged real time. The Logger Console can be used by developers to view the messages logged by AVEVA Work Tasks during development. Only one instance of the Logger Console can be opened at a time. The Logger Console can be opened only on AVEVA Work Tasks installed machines. We recommend to keep the Logger Console closed in production environment as the memory usage will be high. The log history for the Logger Console can be viewed from Log files.

To open the Logger Console,

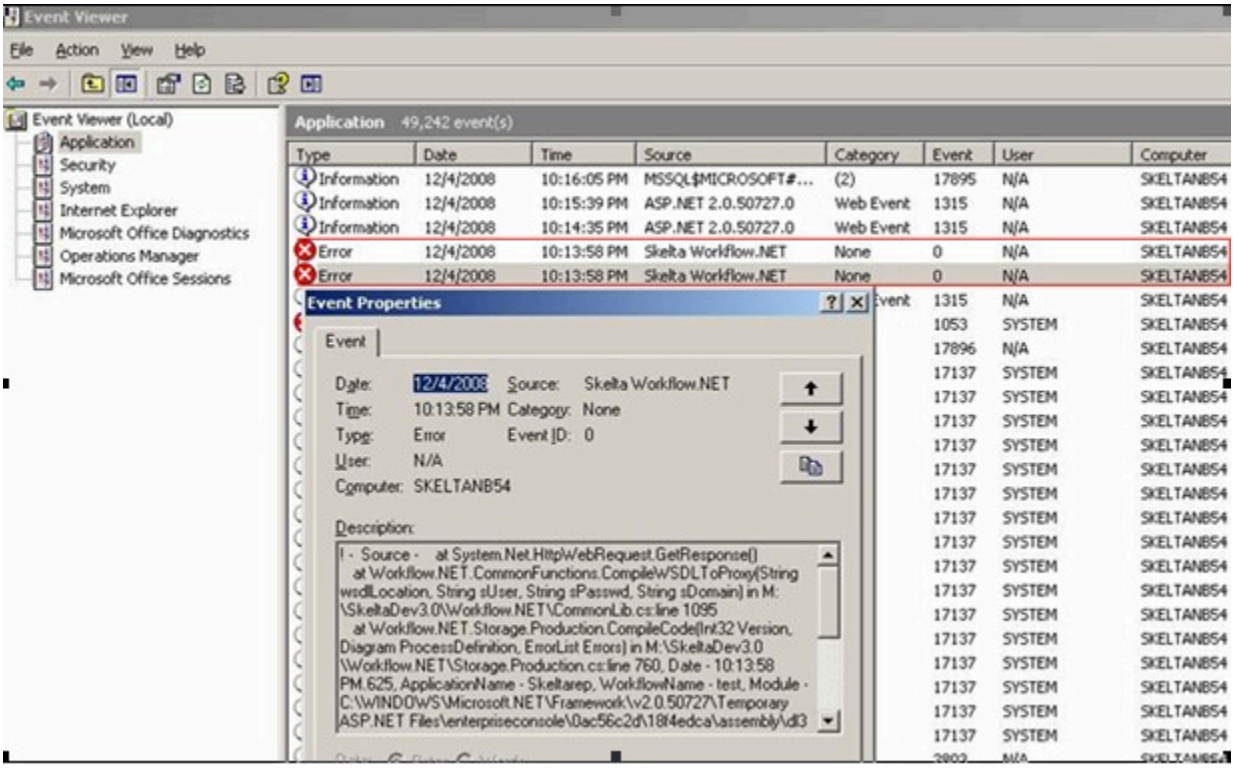
On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **WTLogger Console**.

Event Viewer

Exceptions and warnings are also logged in the Event Viewer. This helps users to check for exception or warning on the production environment. Users have an option to stop logging these exceptions. In addition, users can even avoid logging certain messages in the Event Viewer. Refer to the instructions given under the following headings -

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- To stop logging errors and warnings in the Event Viewer
- To stop logging specific warnings, messages, or errors in the Event Viewer



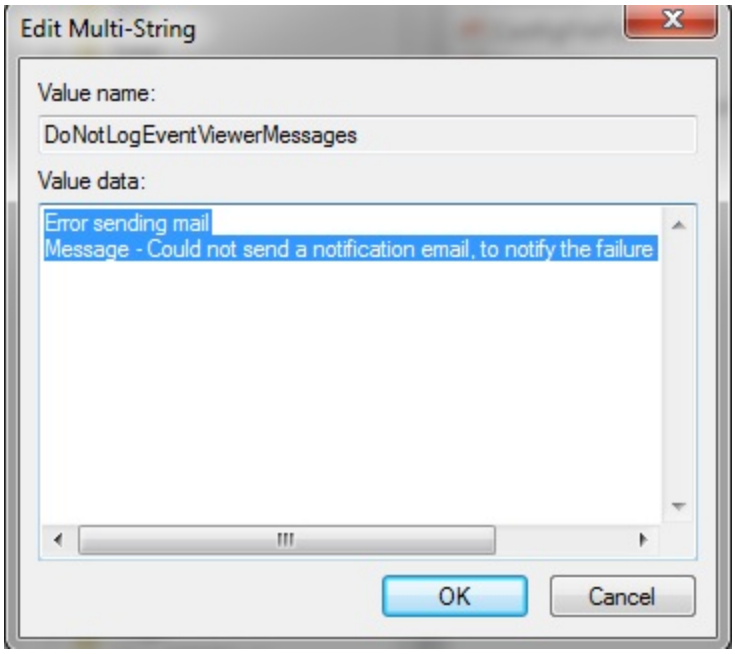
To stop logging errors and warnings in the Event Viewer, complete the following steps:

1. On the Windows **Start** menu, click **Run**.
2. Type **Regedit**.
3. Click **OK**.
4. The Registry Editor screen appears.
5. Create a Registry Key named "StopLoggingOnEventViewer" and set the value as Yes in AVEVA Work Tasks registry under "HKEY_LOCAL_MACHINE\SOFTWARE\AVEVA\WorkTasks".

To stop logging specific warnings, messages, or errors in the Event Viewer:

1. Create a new registry key named "DoNotLogEventViewerMessages" of type "Multi-String Value" in the path HKEY_LOCAL_MACHINE\SOFTWARE\AVEVA\WorkTasks.
2. Add the messages or a portion of a message that need to be omitted from being logged in the event viewer.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Restart the services.

   The error and warning messages specified in the key value will not be logged in the event viewer.

Email alert for Exceptions

Option to get email alerts for all exceptions logged from AVEVA Work Tasks. Mail settings and email id has to be configured for enabling this feature. After it is configured, the exceptions are sent as email message to the mail id specified in To Email field.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.
2. On the **Edit Settings** window, check **Service Settings**.
3. Click the icon next to **Global Settings**. Select the **Communication** tab, and set **From Email** and **To Email** fields.

Understanding and setting Debug Level

The number of messages getting displayed, varies based on the Debug Level which is set in Farm Configuration Wizard.

To change the Debug Level:

- Debug Level 1:

  Information related to starting and ending of a workflow instance and runtime exceptions occurred are logged. We recommends this Debug Level on production environment.
- Debug Level 2:

  Information related to starting and ending of a workflow instance, runtime exceptions occurred during workflow execution and step by step information of each activity execution (Example: Human activities log the number of resources available, Properties set for the activity, Task assigned information for each user) are logged. We recommends this Debug Level on developer environment.
- Debug Level 3:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Information related to starting and ending of a workflow instance, runtime exceptions occurred and Task (Human activity) assigned to user information are logged.

To set the Debug Level:

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**. The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** appears.
2. Click **Next**. The **Edit AVEVA Work Tasks Farm Settings** window appears.
3. Select **Service Settings**.
4. Click **Next**. The **Services/Global Settings** window appears.
5. Click against **Global Settings**. The **Global Settings** screen appears.
6. Click **Log/Transaction**.
7. Select the Debug Level from the drop down box.
8. Click **OK**.

Understanding log messages

The log messages are represented by the following symbols:

- **Information (i)**: This type of message is preceded by a lower case 'i'. It indicates messages containing general information about the events being logged. Each message of this type has four parts.
- **Warning (?)**: This type of message is preceded by a question mark '?'. These messages are warning notes to the users about the events being logged. The messages of this type have a structure similar to the 'Information' messages with the same four parts as mentioned earlier.
- **Error (!)**: This type of message is preceded by an exclamation mark '!'. It indicates error messages with details specific to the error. Messages of this type can be further classified into four parts similar to the other two messages. In addition, it has the 'Exception Stack Trace', which occurs before the machine name and date/time. It can be identified by the keyword 'Source' and it can be used to trace the source of the error.

Example

```
i - @#@Date - 10:48:52 PM.635@#@ApplicationName - LeaveApplication@#@WorkflowName -
EmployeeLeave@#@ExecutionId - 5@#@ExecutionDetailsId - 85@#@WorkItemId -
cd16d86c-5269-4f26-aa97-ad532bd65837@#@Module - C:\WINDOWS\assembly\GAC_MSIL\Skelta.HWS\
3.5.0.0__926ee05bf5158731\Skelta.HWS.dllC:\WINDOWS\assembly\GAC_MSIL\Skelta.HWS\
3.5.0.0__926ee05bf5158731\Skelta.HWS.dll , Assembly version = 3.5.9009@#@_MachineName -
skeltadtp22, Process - WFEngine.NET2@#@Message - Activity of type (Leave Form) added for
resource – Michelle Rose@#@
```

The following table explains the information available in the message:

Information	Description
i –	Type of log
Date - 10:48:52 PM.635	Log Date Time
ApplicationName – LeaveApplication	Repository Name
WorkflowName – EmployeeLeave	Workflow Name

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Information	Description
ExecutionId – 5	Workflow instance ID. Each workflow instance will have a unique ID.
ExecutionDetailsId – 85	Workflow Step ID.
WorkItemId - cd16d86c-5269-4f26-aa97-ad532bd65837	Work item or task ID.
Module -C:\WINDOWS\assembly\GAC_MSIL\ Skelta.HWS \3.5.0.0__926ee05bf5158731\ Skelta.HWS.dll C:\WINDOWS\assembly\GAC_MSIL \Skelta.HWS\3.5.0.0__926ee05bf5158731 \Skelta.HWS.dll , Assembly version = 3.5.9009	Assembly details to track message stack.
_MachineName - skeltadtp22	Machine Name
Process - WFEngine.NET2	Process from which the message was logged.
Message - Activity of type (Leave Form) added for resource – Michelle Rose	Custom Message

Since the message was logged as information, there is no stack information available.

For exceptions the first message will carry the stack trace. Check the following message.

```
! -  Source -    at System.Data.SqlClient.SqlConnection.GetOpenConnection(String
method)   at System.Data.SqlClient.SqlConnection.ValidateConnectionForExecute(String
method, SqlCommand command)   at System.Data.SqlClient.SqlCommand.ValidateCommand(String
method, Boolean async)   at
System.Data.SqlClient.SqlCommand.RunExecuteReader(CommandBehavior cmdBehavior,
RunBehavior runBehavior, Boolean returnStream, String method, DbAsyncResult result)   at
System.Data.SqlClient.SqlCommand.RunExecuteReader(CommandBehavior cmdBehavior,
RunBehavior runBehavior, Boolean returnStream, String method)   at
System.Data.SqlClient.SqlCommand .ExecuteScalar()   at
Workflow.NET.DataHandler.SQLDataHandler.ExecuteInsert (DataRow Row, Boolean InsertId) in
M:\SkeltaDev3.0\Workflow.NET\SQLDataHandler.cs:line 1152, Date - 4:08:16 PM.888, Module
- C:\WINDOWS\TEMP\Skelta2007Temp\ShadowTempDefault\assembly\dl3\057300c8\
f66dc290_1643c901\Workflow.NET.NET2.DLL@#@_MachineName - skeltanb30, Process -
WFEngine.NET2, Message - @@SQL -insert into SKWorkItemTransaction (Id,Application,
ChannelName,HWSActivityId,WorkItemId,VirtualActorId,QueueId,Type,ToVirtualActorId,Create
dDateTime,CommentsId,DataId,CurrentStatus,LastUpdatedDateTime) values (@p1,@p2,@p3,
@p4,@p5,@p6,@p7,@p8,@p9,@p10,@p11,@p12,@p13,@p14) ; select @@identity, - ExecuteScalar
requires an open and available Connection. The connection's current state is open.@#@
```

Database Connection Errors while Installing Farm

Before going through this section, see Prerequisites for Debugging Exceptions.

Troubleshooting issues encountered for Database Connections while installing the Farm Server or the Farm Client are listed below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. User receives 'Could not establish connection to the SQL Server data source, Login failed for user <UserName>' on the Create Farm Database page or on the Connect to existing database page.

Probable Cause

SQL Server Instance is not able to authenticate the user.

Solution

The user in the error message, needs to be added in the SQL Database Instance(entered in Server Name field). This user can be a Windows user or can be a SQL user.

- Add Windows user
- Add SQL user

Database User should have logon machine rights in the Database Server Instance and the Farm Server (Database User should be able to login in to these machines successfully).

Add Windows user

1. In SQL Server Management Studio, open Object Explorer and expand the folder of the server instance in which to create the new login.
2. Double click the Security folder, right-click Logins.
3. Click New Login.
4. On the General page, enter the name of a Windows user in the Login name box.
5. Select Windows Authentication.
6. Click OK. The user will get listed in the Logins folder.

Add SQL user

1. In SQL Server Management Studio, open Object Explorer and expand the folder of the server instance in which to create the new login.
2. Right-click the Security folder, point to New, and then click Logins.
3. Click New Login.
4. On the General page, enter a name for the new login in the Login name box.
5. Select SQL Server Authentication. Windows Authentication is the more secure option.
6. Enter a password for the login.
7. Select the password policy options that should be applied to the new login. In general, enforcing password policy is the more secure option.
8. Click OK.

Check *http://msdn.microsoft.com/en-us/library/aa337562.aspx* for more details.

2. Error: *Test Connection Succeeded. However the login - <UserName> does not have dbcreator role in <ServerInstanceName>.*

Probable Cause

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Farm Configuration Wizard is trying to create the Farm Database, entered in Database Title, using the <UserName> from the error message.

Solution

1. In SQL Server Management Studio, open Object Explorer and expand the folder of the server instance in which to create the new login.
2. Double click the Security folder, double click Logins.
3. Right-click on the <UserName>. Click Properties.
4. Click Server Roles. Check dbcreator.
5. Click OK.

3. The Create the Farm Database or Connect to existing database page does not respond while configuring the Farm.

Probable Cause

The local machine is not able to connect to the database server instance.

Solution

1. Connect to the database using an SQL Server Instance Client.
2. If Windows authentication was used while connecting, confirm that the database user has not changed the Active Directory password.
3. **Test Connection Tool by** AVEVA Work Tasks**:**

   This tool is used with the Load Balanced Server. Using the Test Connection tool, check if all the machines (Server and Client) are able to communicate with each other before installation. This test ensures that all the Server machines involved can actively support the Load Balancing and Failover support features of the Load Balanced Server. To check the communication between machines, the Test Connection tool must run simultaneously on all the machines.

   Messages received from all the machines are displayed by the Test Connection tool indicating that the machines are able to communicate with each other. Machines that are not able to communicate will not be able to receive or display messages from other machines.
4. **DTCTester Tool by Microsoft**:

   The MSDTC connection settings can also be checked using the DTCTester tool. Check the availability in Microsoft Download site.

   The DTCTester tool does the following:

   - Establishes a connection to SQL Server using a data source name (DSN), username and password. You can establish this connection via a command line using the default network library.
   - Creates a temporary table
   - Enlists the connection in a transaction
   - Does an insert on the temporary table
   - Commits the distributed transaction
   - Selects inserted value to make sure it was committed
   - Closes the connection

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To use the DTCTester Tool:

1. Create an ODBC data source for your SQL Server through the ODBC utility in Control Panel.
2. Execute the following from the command line: dtctester <dsn name><user name><password>.
3. Replace the values in brackets as appropriate for your environment.

The table below lists the ports to be opened for DTC:

Port Number	Purpose
135	RPC EPM (end point mapper)
1433	TDS SQL traffic when using TCP/IP
1434	SQL 2000 Integrated Security
5100 - 5200	MSDTC (Dynamically assigned a port by the EPM)

Check Enable and Verify MSDTC for more details.

4. Error logged in the Logger Console: RunScript failed, ExecuteNonQuery failed for Database <Database Name>

Probable Cause

If the local machine is 64-bit, two dlls of 'Microsoft.SqlServer.BatchParser' with Processor Architecture as x86 and AMD64 should exist in the assembly. If you have only one version of this dll present, refer to the following solution.

Solution

1. Go to the following website:
   http://www.microsoft.com/en-in/download/details.aspx?id=29065
2. Download and install the following packages:
   - X64 Package (SQLSysClrTypes.msi) from the Microsoft System CLR Types for Microsoft SQL Server 2016 section
   - X64 Package (SharedManagementObjects.msi) from the Microsoft SQL Server 2016 Shared Management Objects section.

Client Service Errors

Before going through this section, see Prerequisites for Debugging Exceptions.

Issue

User receives 'Unable to run AVEVA Work Tasks Client Service' message.

Probable Cause

When one machine is having multiple IP addresses or running on LAN or High-Speed Internet.

Solution

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- While Installing the Configuration Farm
- While Editing the Configuration Farm

While Installing the Configuration Farm

1. In your installation scenario, when you have reached to the 'Defining Services and Global Settings' step, perform till step 3 which are instructed in this section.
2. Select the **Machine Settings** tab, and set Machine Address field to local IP Address.
3. Click **OK**.

While Editing the Configuration Farm

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.
2. From **Edit Settings** page, check **Service Settings**.
3. Click the icon next to **Global Settings**. Select the **Machine Settings** tab, and set **Machine Address** field to local IP Address.
4. Click **OK**.

Windows Service Errors

While starting a AVEVA Work Tasks windows service, if the status is "starting" for a long time and does not change to "running", then there are chances that the Database server is down or the service is not able to connect to the Farm database. Ensure the Database server is up and able to connect to the Farm database.

Client Service Monitoring Errors

- Client service monitoring is not enabled after changes to the ServiceMonitorConfig.xml. We can ensure if client service is enabled by checking in the log file for this message:

  Info:- AVEVA Work Tasks Services Monitoring. Monitoring is enabled: "".

  This message is logged every 10 minutes. This could be due to an invalid syntax in ServiceMonitorConfig.xml. Open the file in XML editor and ensure all syntaxes are proper. Ensure that changes made are not commented.

- Client service configuration file changes are not getting reflected. After making changes to ServiceMonitorConfig.xml, ensure the Client service is restarted. After the client service is restarted, it re-caches the configuration changes and all the changes will be considered.

- Client service is not able to start a AVEVA Work Tasks service. In some situations, client service may not be able to start a service. It will continue to restart the service and send restart failure emails. One of the reasons could be that the database server is down or is not getting connected to the windows services.

- Client service is restarting the remaining AVEVA Work Tasks windows services continuously in regular interval and sending the emails. Ensure that all services are able to connect to the database and the database is up. Other reason could be due to thread hanging behavior of WFEngine. See Workflow Engine for more information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- In Service Monitoring, if the thread hanging or locking behavior is not getting monitored, then there are chances that cached variable value is zero. This can happen, if the database server is down while reading the data from the database. Restart the client service which will read data once again and recreate the cache.

Database Connection Errors

Before going through this section, see Prerequisites for Debugging Exceptions.

1. While opening the Central Configuration or while opening the Farm Configuration Wizard, the following database connection error message appears on Welcome to the AVEVA Work Tasks Configuration Wizard.

- *A valid connection to AVEVA Work Tasks database could not be established. Would you like to change the Database User Settings?*
- The following error message appears in the Central Configuration.

  Cannot proceed further. A valid connection to the AVEVA Work Tasks Farm could not be established. Check the Connection settings and also MSDTC settings.

  You will also get the message in the WTLogger Console:

  Probable Cause

Database User's credentials are changed:

- From Active Directory
- From SQL Database instance

Solution

From Active Directory

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **AVEVA Work Tasks Farm Configuration Wizard** appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm settings** screen appears.
3. Select **Database User Settings**.
4. Click **Next**.

   The **Database User Settings** screen appears.
5. Enter the updated User Name, Password, and Domain.
6. Click **OK**. The following confirmation message appears:

   Make sure that the login used for connecting to database has SELECT, INSERT, UPDATE, DELETE permissions on all tables and Execute permission on the Stored Procedures.
7. Click **OK**. A message indicating successful updates of database user settings appears.
8. Click **OK**.

From SQL Database instance

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.
   The **AVEVA Work Tasks Farm Configuration Wizard** appears.
2. Click **Next**.
   The **Edit AVEVA Work Tasks Farm settings** screen appears.
3. Click **Database Settings**.
   The **Connect to existing database** screen appears.
4. From the **Connection String** option, modify the User Id and Password for connecting to the database.
5. Click **Test Connection**. On success, the *Test Connection Succeeded* message appears.
6. Click **OK**.
7. Click **Next**. The user credentials are updated.

2. User is getting the following database connection error message while creating Repository Datasource or while opening an Enterprise Console.

Error while creating Repository Datasource in the Central Configuration:

Invalid connection string. Please check MSDTC settings and check if Database User Settings are valid, if you are using windows authentication. Check AVEVA Work Tasks Logger Console for more details.

Error while opening an Enterprise Console. This page gets hanged and displays the following message in the Logger Console.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Probable Cause

Database User Settings are not set and Windows authentication is used in the Repository Datasource.

Solution

- Either use Windows authentication for the Repository Datasource
- SQL authentication for the Repository Datasource

Steps for Windows authentication

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.
   The **AVEVA Work Tasks Farm Configuration Wizard** appears.
2. Click **Next**.
   The **Edit AVEVA Work Tasks Farm settings** screen appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Select **Database User Settings**.
4. Click **Next**.

   The **Database User Settings** screen appears.
5. Enter the updated User Name, Password, and Domain.
6. Enter Database User credentials.
7. Click **OK**.

Steps for SQL authentication

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **AVEVA Work Tasks Central Configuration**.

   The **Central Configuration** page appears.
2. Click **Datasource Settings**. The **Datasource List** appears.
3. Click required Datasource from Datasource column.
4. Make changes to user SQL authentication. Select **SQL** from **Authentication Type** drop down list.
5. Click **Test Connection**, and then click **OK**.

IIS Error while Configuring Farm Configuration

Before going through this section, see Prerequisites for Debugging Exceptions.

Issue

The following IIS error message appears while opening the Farm Configuration Wizard.

IIS is not installed. To proceed install IIS.

Probable Cause

IIS is not installed on your Server.

Solution

You must install a valid version of IIS before configuring the Farm Configuration.

ASP.NET Error while Configuring Farm Configuration

Before going through this section, see Prerequisites for Debugging Exceptions.

Issue

The following ASP.NET related error message appears while opening the Farm Configuration Wizard.

Failed to enable the IIS features automatically. Please enable them manually.

Probable Cause

IIS is installed on your Server, but ASP.NET is not installed on your system.

Solution

You must install ASP.NET before configuring the Farm Configuration.

For more information, see Configure Internet Information Services.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Windows Authentication or Anonymous Authentication Error

Before going through this section, see Prerequisites for Debugging Exceptions.

Issue

The following authentication error messages appears while opening the Farm Configuration Wizard.

Ensure that Windows Authentication is enabled to access the Central Configuration and Windows authentication or anonymous authentication is enabled to access the Enterprise Console.

Probable Cause

Windows Authentication or Anonymous Authentication is not enabled on your computer.

Solution

You must enable:

- Windows Authentication to access the Central Configuration
- Windows Authentication or Anonymous Authentication to access the Enterprise Console

For more information, see Windows Server 2008 or Windows Vista.

Non-Query Execution Failure

Before going through this section, see Prerequisites for Debugging Exceptions.

Issue:

Error on the Central Configuration: *ExecuteNonQuery failed for the Database <databaseName>*.

Probable Cause:

Workflow.NET.sql script located at <AVEVA Work Tasks installed path>\DatabaseScripts\SQLServer could not run on the database Instance.

Solution:

Run the Workflow.NET.sql file located at [AVEVA Work Tasks Installed Folder]\DatabaseScripts\SQLServer on the Database manually and in case of any issues while running the script, contact support.

Recommended collation for the SQL server are:

- SQL_Latin1_General_CP1_CI_AS
- SQL_Latin1_General_CP1_CS_AS

Note: This is valid only for SQL Server data source type.

Not Authorized to Access Site Error

Before going through this section, see Prerequisites for Debugging Exceptions.

Issue

While opening the Central Configuration, the following error message appears:

You are not authorized to access this site. You need to be a member of Group Administrators or

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

WorkflowManagementAdmins.

Probable Cause

The user who is accessing the Central Configuration is actually not a member of Administrators or AVEVA Work Tasks Admins group.

or

Anonymous Access to the Central Configuration is enabled.

Solution

The user accessing Central Configuration needs to be a member of the Administrators group for the machine.

Anonymous access rights for the CentralConfig site should be removed from IIS.

Steps

1. Click Start > Run.
2. Type inetmgr > Enter.
3. In the IIS, right click on the CentralConfig site > Directory Security > Edit. The Authentication Methods screen appears.
4. Uncheck Enable anonymous access. The site should use only Integrated Windows Authentication.
5. Click OK.

Enterprise Console Errors

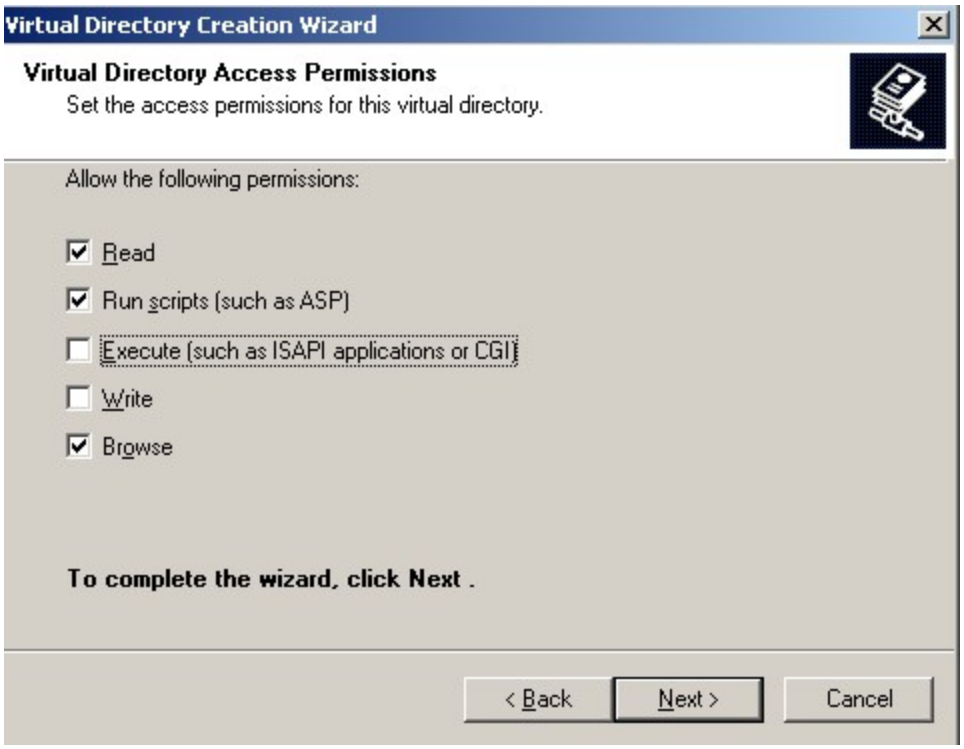
Before going through this section, see Prerequisites for Debugging Exceptions.

1. A message "The resource cannot be found" is displayed while opening the Enterprise Console.

Solution

1. Start > Run > inetmgr.
2. Expand Internet Information Services.
3. Expand AVEVAWorkTasksSites > EnterpriseConsole.
4. Right-click **EnterpriseConsole** > **New** > **Virtual Directory**.
5. Click **Virtual Directory**.
6. Click **Next**. Type BPMUITempaltes.
7. Click **Next**.
8. Select *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\BPMUITemplates*.
9. Click **Next**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



10. Click **Next**. Click **Finish**.
11. Right-click on newly created BPMUITemplates site. Click Properties.
12. Click **Remove**.
13. Click **OK**.

2. The following error message appears while opening the Enterprise Console:

Repositories are not mapped for the current site.

Resolution

- Check if any Repository is mapped to the Enterprise Console.
- Check status of the Enterprise Console in the Farm Database.

Steps to check Repository mapping to the Enterprise Console

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**. The **Central Configuration Web** page appears.
2. Click **Repository Settings**. The **Repository List** page appears.
3. From the Repository Name column, click on a **Repository**.
4. Map the Repository to the Enterprise Console.
5. Click **OK**.
6. Refresh Enterprise Console from Repository List page.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to check the status of the Enterprise Console in the Farm Database

- Check in the Farm database, table SKWebSite, if the related Enterprise Console has the "IsActive" flag set to False, change it to True.

3. Login fails in case you create a new Enterprise Console and configure the repository with AVEVA Identity Manager (AIM)

Resolution

- Do the IIS reset.
- Reconfigure AIM from Central Configuration.

4. Application error occurs when you try to login to the Enterprise Console after a long time

Application error occurs when you have two repositories configured (say, AIM repository and AVEVA List repository) and you perform the actions below:

- You logged into one of the repositories (say, AIM repository) and after a long time you browse the Enterprise Console URL and try to login with different repository (say, AVEVA List repository).
- You logged out from one of the repositories (say, AIM repository) and try to login with different repository (say, AVEVA List repository).

Resolution

Clear the cookies and login again.

Authentication and Personalization

Issue

The user is able to view all the web parts only in the Browse mode and is unable to add, edit, or delete the web parts.

Probable Cause

Anonymous Authentication has been Enabled.

Resolution

The user has to check the IIS 7 settings and make the following changes:

Steps

1. Click Start.
2. Point to Administrative Tools > Internet Information Services (IIS) Manager.

   The Internet Information Services (IIS) Manager screen appears.
3. Expand Sites, select Enterprise Console, and then select Authentication from right pane.
4. Ensure that the authentication settings are as follows:
   - Anonymous Authentication: Disabled

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- ASP.NET Impersonation: Disabled
- Basic Authentication: Disabled
- Forms Authentication: Disabled
- Windows Authentication:Enabled

  If the status of the Anonymous Authentication is Enabled, then disable the Windows Authentication and enable Forms Authentication.

5. Right-click the Windows Authentication Mode and select Disabled.
6. Right-click the Forms Authentication Mode and select Enabled.

This always works in the above mentioned combination. If the Forms Authentication is enabled then the Windows Authentication must be disabled and the Anonymous Authentication must be enabled in the IIS Manager.

If in the above scenario, the Anonymous Authentication is not enabled, then the Auto-redirection for Active Directory will skip the login page and load the web parts only in Browse mode. In order to be able to add, edit, or delete the web parts, the user has to enable the Anonymous Authentication.

If the user changes the authentication mode to Forms, then the code given below should be added to the web.config file:

```
<authentication mode="Forms">
<forms timeout="500000" loginUrl="~/BPMUITemplates/Default/Repository/Site/Login.aspx">
</forms>
</authentication>
```

Performance Issue While Accessing Enterprise Console

Issue:

Performance issue while accessing different pages of Enterprise Console.

Probable Cause

SQL Server instance connectivity issues between the AVEVA Work Tasks server and the AVEVA Work Tasks database server.

Resolution

To identify possible cause of delayed response of Enterprise Console:

1. Open the SQL Server instance on AVEVA Work Tasks server.
2. Connect to an instance of the database server.
3. Execute a query which fetches large number of records. For example, Select * From SWExecute
4. Verify whether the query takes a long time to execute or throws any exception.

If the query takes a long time to execute or throws any exception, validate the SQL Server instance connectivity between the server and the database server.

This approach is applicable only when the engine and the database server are on two separate computers.

Also, verify that the connectivity between the Active Directory and AVEVA Work Tasks server is proper using the AVEVA Work Tasks Active Directory (ADGC) Tool. This approach is valid even when the server and the database

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

server are on the same computer.

Central Configuration is not Loading

Before going through this section, see Prerequisites for Debugging Exceptions.

Issue

Central Configuration is not loading. The following error message appears:

Unable to locate string resources under directory (C:\Program files\AVEVA\Work Tasks\locales\)

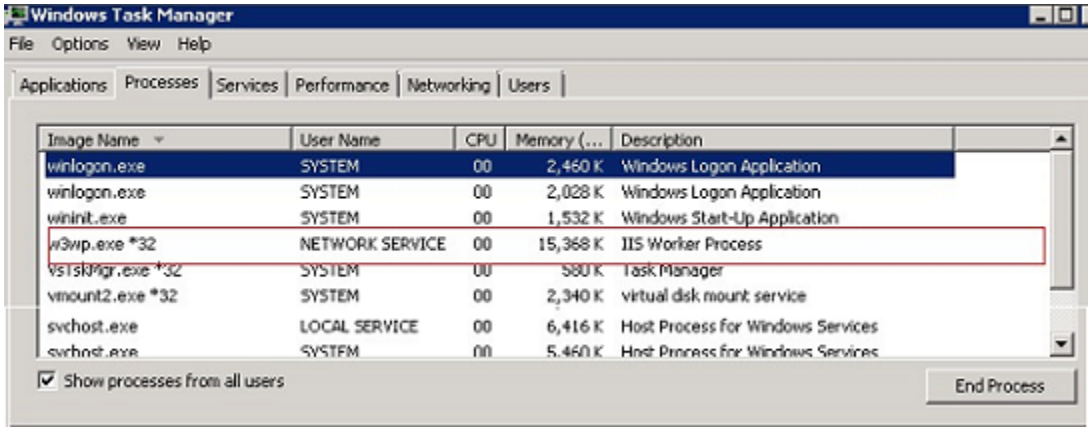
Probable Cause

This error occurs if the user specifies different installation path other than the default path (*C:\Program files\AVEVA\Work Tasks*) while installing AVEVA Work Tasks on a 64-bit machine with Windows Server 2008 and IIS7.

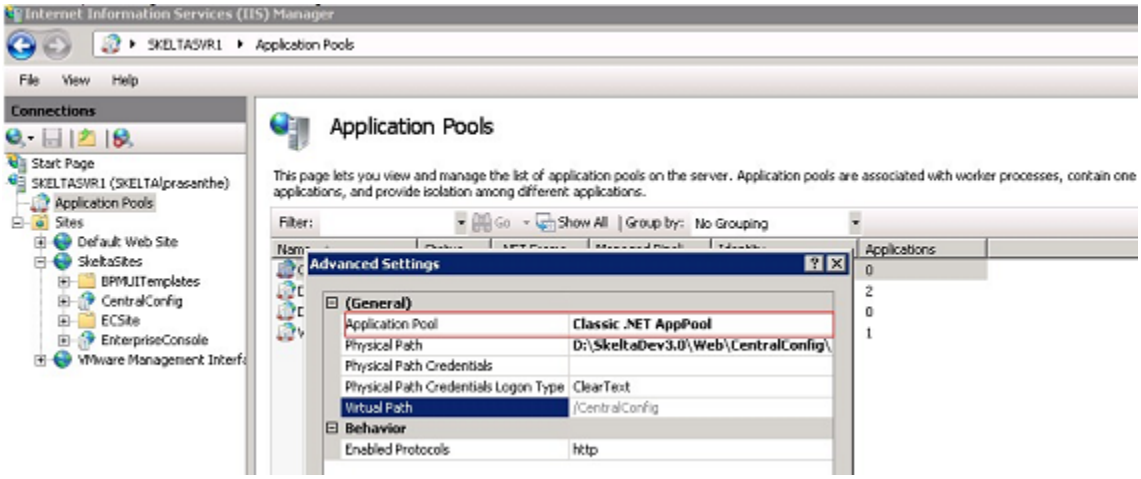
Resolution

To avoid the error, follow the configuration steps given below to run a 32-bit in 64-bit mode.

1. Go to Task Manager.
2. Click the Processes tab, and locate the w3wp process. The following figure shows the w3wp process is running on 32-bit machine.

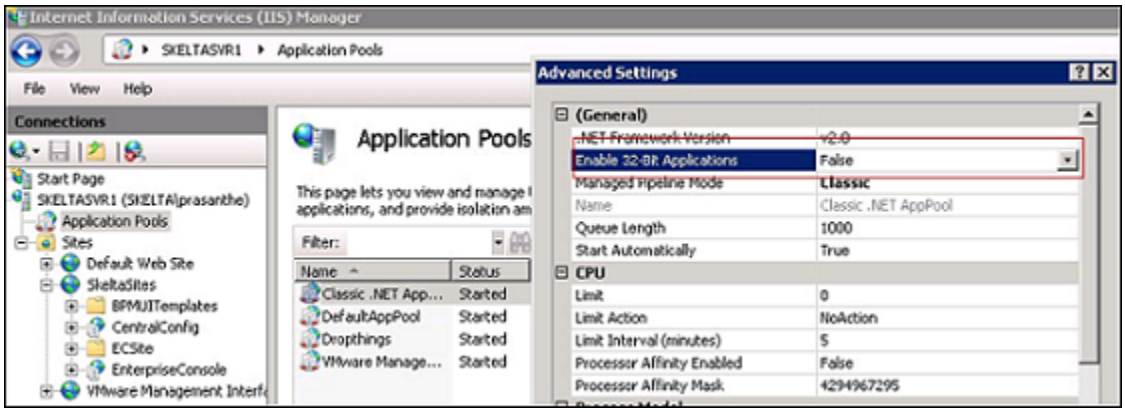


3. Go to the Internet Information Services (IIS) Manager, and check the pool configured for Central Config. The following figure shows that the pool is configured for 32-bit machine.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. To edit the pool settings to run in 64-bit mode, complete the following steps:
   a. Right-click on the Application Pool and select Advanced Settings or, select the same from the Actions pane after selecting Application Pool.
   b. In the Advanced Settings window, change Enable 32-bit Applications to True, if you want the Application Pool to spawn in a 32-bit mode.
   c. Click OK.



Communication Service Errors

Before going through this section, see Prerequisites for Debugging Exceptions.

1. You get the following error while setting Mail Settings in the Farm Configuration Wizard.

AVEVA Work Tasks Communication Service test failed. Service test failed

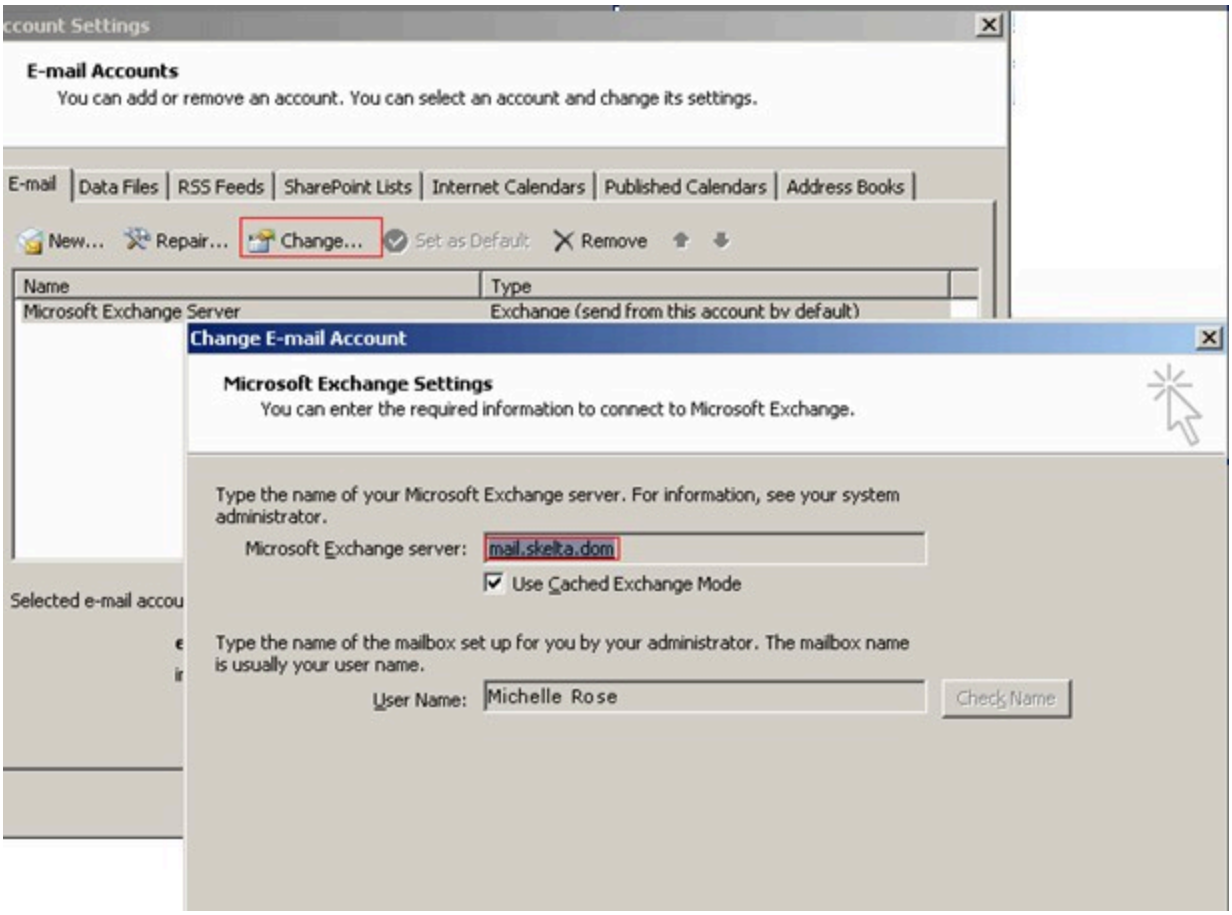
Following is the message displayed in the Logger Console:

```
! - Test SMTP failed, - Failure sending mail., Inner Exception :Unable to connect to the
remote server, Module - C:\Program Files\AVEVA\AVEVA Work Tasks\Bin\
AVEVAWorkTasksFarmConfigMgr.exe ,Assembly version = 6.0.1027.0, Source - at
System.Net.Mail.SmtpClient.Send(MailMessage message) at
Workflow.NET.EmailHandler.SendMail(String smtpServer, Int32 smtpPort, Boolean isssL, String
authenticationUserName, String authenticationuserPassword, Boolean ispasswordEncrypted,
String fromAddress, String toaddress, String subject,String body) at AVEVA Work
Tasks_Console_Setup.GlobalSettings.linkTestSMTP_LinkClicked(Object sender,
LinkLabelLinkClickedEventArgs e), BANVM-RND6, Process - AVEVAWorkTasksFarmConfigMgr
```

Solution

Check Host and Port settings for **Email Settings**. Host refers to Mail server and the port refers to the port in which mail server is listening for mails. (Example: Host value can be set as Machine name/IP of Microsoft exchange server. The default port for Microsoft Exchange server is 25). Microsoft exchange server name can be found in **Microsoft Office Outlook > Tools > Account Settings** option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



With reference to above image "mail.skelta.com" will be set as Host and Port will remain as 25. After settings Host & Port, Set values for "From Email" and "To Email" fields to test SMTP.

Exception logged in logger console:

```
! - Test SMTP failed, - Mailbox unavailable. The server response was: 6.7.1 Unable to relay
for michellerose@gmail.com, Module - C:\Program Files\AVEVA\AVEVA Work Tasks\Bin\
AVEVAWorkTasksFarmConfigMgr.exeC:\Program Files\AVEVA\AVEVA Work Tasks\Bin\
AVEVAWorkTasksFarmConfigMgr.exe , Assembly version = 3.0.*, Source - at
System.Net.Mail.SmtpTransport.SendMail(MailAddress sender, MailAddressCollection
recipients, String deliveryNotify, SmtpFailedRecipientException& exception) at
System.Net.Mail.SmtpClient.Send(MailMessage message)
```

Solution

Mail server configured does not support sending mail to an external mail id other than corporate account. Change the "To Email" to a proper corporate account and try.

2. Mail Channel or Wait-For-Email activity not working. Following error may pop up on start of the Communication Service.

```
? - Could not listen on port 8770. Please make sure that this port is available. Will try
to listen on this port after some time., Module - M:\SkeltaDev3.0\Bin\
Workflow.NET.Communication.NET2.dllM:\SkeltaDev3.0\Bin\Workflow.NET.Commuication.NET2.dll ,
Assembly version = 3.0.*.
```

Probable Cause

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

May be Port 8770 used for AVEVA Work Tasks - Communication Service may be used by some other application. This port is used for communication between the workflow engine and the communication engine.

Resolution

Change the Communication Service port number.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.

   The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** appears.
2. Click **Next**.

   The **Edit AVEVA Work Tasks Farm Settings** window is displayed.
3. Select **Service Settings**.
4. Click **Next**. The **Services/Global Settings** page appears.

   The **Enterprise Server Settings** can be selected if the Farm Server Installation is using Enterprise Server Installation.
5. Click against AVEVA Work Tasks - Communication Service to modify the Communication Service settings
6. Change the Port in Communication Server tab.

   Listener IP Address is AVEVA Work Tasks installed machine's IP Address.

Issue

The Communication Service no longer processes Mail Channel mails from the POP3 account after the From Email property has been changed for a repository.

Probable Cause

The POP3 account for the Communication Service has not been configured with the new Email ID that was configured for the From Email property of the repository.

Resolution

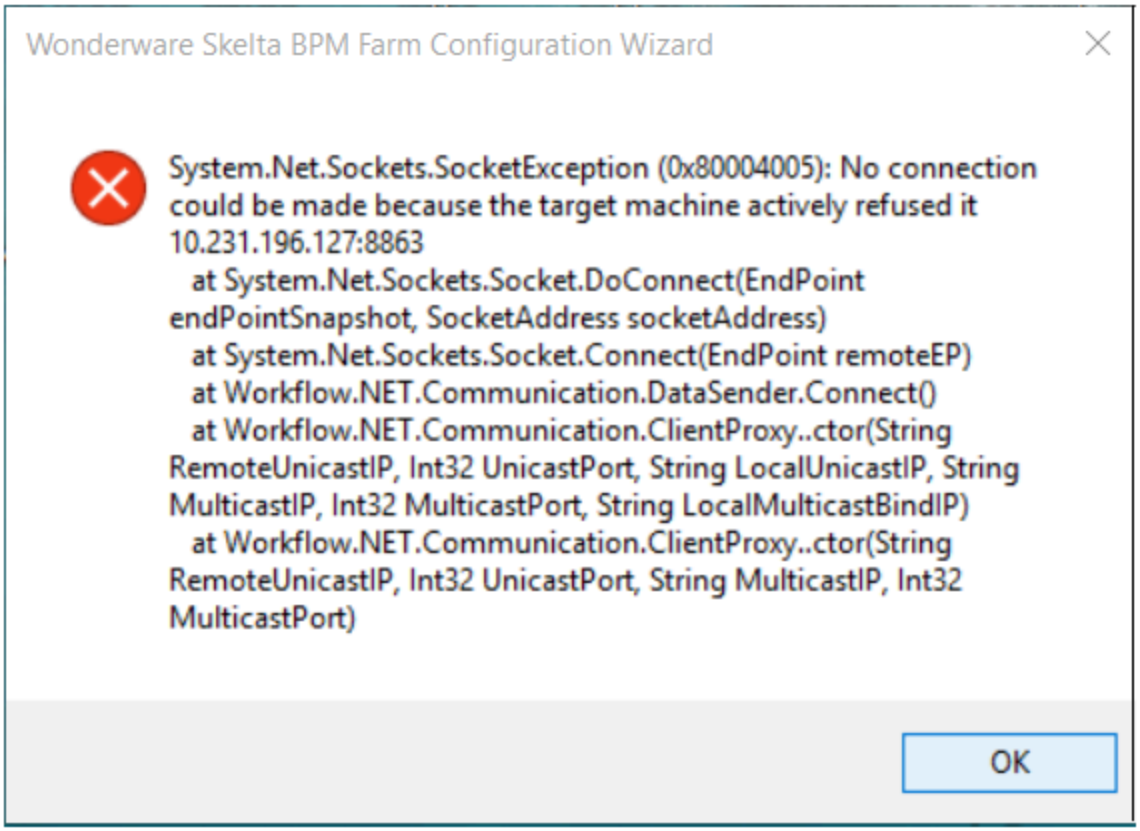
Add a new POP3 user or update the existing POP3 user with the email account that was configured for From Email property of the Repository, in the Communication Service settings of the Farm Configuration Wizard and restart the service.

Farm Configuration Error

1. Issue:

When you run the Farm Configuration Wizard, the following error message appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Probable Cause

This error occurs if there is a change in the IP address of the machine, where AVEVA Work Tasks is configured.

Resolution

Restart all the AVEVA Work Tasks Services.

2. Continuous Error Logs in WTLogger Console

The following errors are logged continuously in the WTLogger Console.

```
Error: Cannot add duplicate collection entry of type 'virtualDirectory' with uni
que key attribute 'path' set to '/BPMUITemplates'

).Skelta.Core.DS.SingleTableObjectDataRowException: Could not load data for tab
le (SKRepository), No matching record found for column: @Name, value: Filename:

Error: Cannot add duplicate collection entry of type 'virtualDirectory' with uni
que key attribute 'path' set to '/BPMUITemplates'

, Repository:$CentralConfig$, severityLevel: Fatal
   at Skelta.Core.DS.SingleTableObject.LoadRowData()
   at Skelta.Core.DS.SingleTableObject.EnsureRecordsInTableRow()
   at Skelta.Core.DS.SingleTableObjectWithName.Initialize(String name)
   at Skelta.FarmManager.Repository..ctor(String name)
   at Workflow.NET.Config.GetDataSourceType(String Application), Module - C:\Win
dows\Microsoft.Net\assembly\GAC_MSIL\Workflow.NET.NET2\v4.0_5.0.0.0__c932f724c98
4e915\Workflow.NET.NET2.dll ,Assembly version = 5.0.1001.0, ThreadID - 3, BPM201
7SP16, Process - SKClientSrv, 15:07:23.444
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
? - Error while getting the data source type for application ( Filename:
Error: Cannot add duplicate collection entry of type 'virtualDirectory' with uni
que key attribute 'path' set to '/BPMUITemplates'

).Skelta.Core.DS.SingleTableObjectDataRowException: Could not load data for tab
le (SKRepository), No matching record found for column: @Name, value: Filename:
 Error: Cannot add duplicate collection entry of type 'virtualDirectory' with uni
que key attribute 'path' set to '/BPMUITemplates'

, Repository:$CentralConfig$, severityLevel: Fatal
   at Skelta.Core.DS.SingleTableObject.LoadRowData()
   at Skelta.Core.DS.SingleTableObject.EnsureRecordsInTableRow()
   at Skelta.Core.DS.SingleTableObjectWithName.Initialize(String name)
   at Skelta.FarmManager.Repository..ctor(String name)
   at Workflow.NET.Config.GetDataSourceType(String Application), Module - C:\Windows\
Microsoft.Net\assembly\GAC_MSIL\Workflow.NET.NET2\v4.0_5.0.0.0__c932f724c98
4e915\Workflow.NET.NET2.dll ,Assembly version = 5.0.1001.0, ThreadID - 3, BPM201
7SP16, Process - SKClientSrv, 15:07:24.242

? - Error while getting the data source type for application ( Filename:
Error: Cannot add duplicate collection entry of type 'virtualDirectory' with unique key
attribute 'path' set to '/BPMUITemplates'
```

Probable Cause

While uninstalling AVEVA Work Tasks from a machine, all the files might not be completely removed, due to some issues. When you install AVEVA Work Tasks on this machine, the above issue might occur.

Resolution

To resolve this issue, perform the following:

1. Browse to the Installed folder of AVEVA Work Tasks.
2. Remove the folder.
3. Restart the machine.
4. Install the latest version of AVEVA Work Tasks.

3. The Farm Configuration Wizard crashes while configuring License Server 3.6 version

Probable Cause:

The port number goes as blank while configuring License server.

Resolution:

Provide the port number as 55555 to continue with Farm configuration.

Workflow Test Run Errors

Before going through this section, see Prerequisites for Debugging Exceptions.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Error: '*No published workflow '<WorkflowName>' found to execute.*

Probable Cause

The workflow is not published.

Solution

1. Go to Workflow List.
2. Right-click the workflow.
3. Publish the workflow.

2. While opening the Test Run window, error "*Workflow Engine not running*" is thrown.

Probable Cause

The Workflow Engine service is stopped.

Solution

Steps

1. Go to Run, and open services.msc. The Windows Services window appears.
2. Right-click Workflow Engine.
3. Click Start.

Move or Copy Item Activity Errors

Before going through this section, see Prerequisites for Debugging Exceptions.

Issue

Workflow ends with "executed with error". In Process Execution view, the last activity which was executed is Move Copy Item activity.

Probable Cause

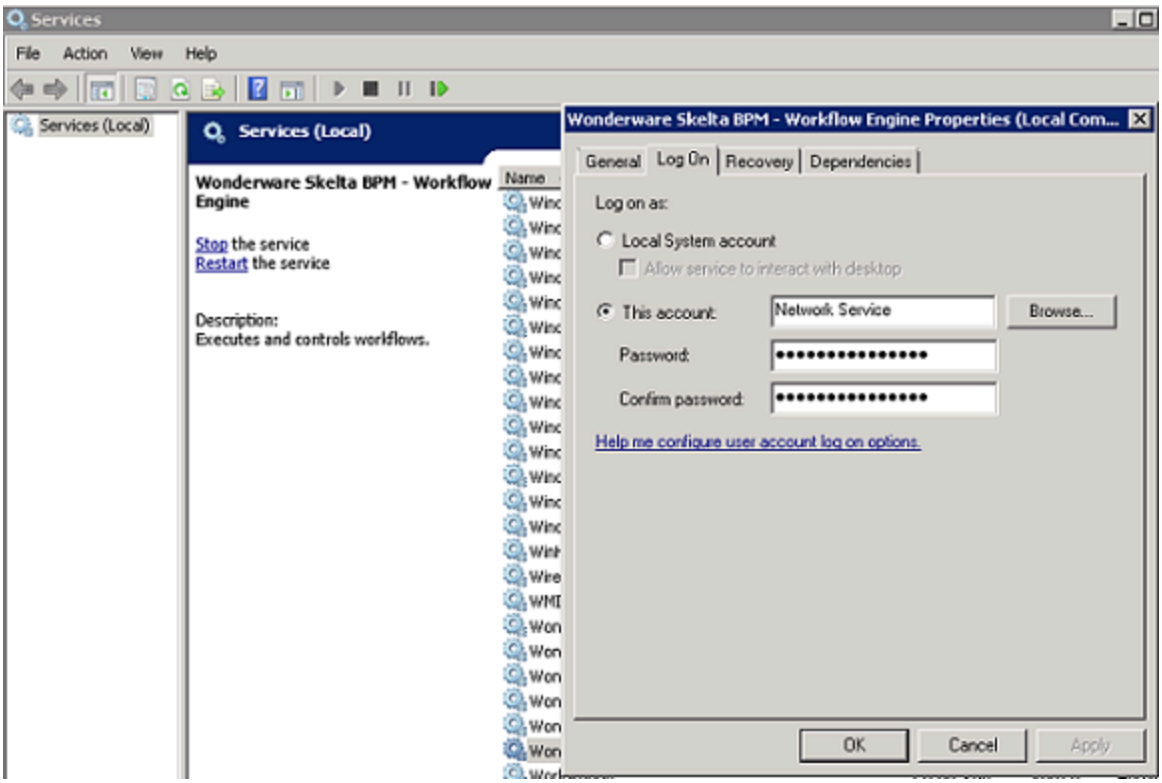
Using the Move Copy Item activity, the user may get an exception as "Access to the path <path> is denied" and "Source Path Not Found" in the Log files or Logger console or Event Viewer.

Solution

Ensure that all the services are running with the specified user account and that user should have read and write access on the physical or network folder.

Ensure that the password will not expire or change for this account.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



If it is Enterprise Server scenario, then specify a network location (\\<computer-name>\Source) as Source and Destination property for Move Copy Item activity. Also ensure that the services running on AVEVA Work Tasks Servers are running with the specified user account having rights on the physical or network folder specified in the source and destination property.

Changing Resource Account Credentials

Before going through this section, see Prerequisites for Debugging Exceptions.

This topic provides information about the settings to be used when the authenticated user credentials are changed.

Settings to be used after changing the resource account credentials to log on to AVEVA Work Tasks sites.

Settings to connect to the Enterprise Console:

- Changing Database User Settings for Windows Authentication
- Changing SQL User Credentials for SQL Authentication

Settings to connect to the Central Configuration:

- Changing Password of SQL User Credentials used to Connect to Repository Data Source

  If the password of impersonation user that is used for AVEVA Work Tasks SharePoint services authentication is changed, follow the steps for resetting the password in this link.
- Changing a Web Service Authentication User

  If the password is modified by the user then the same should be modified using Enterprise Console, Central

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration and SharePoint. If this is not done the user will not be able to login to any of the AVEVA Work Tasks sites.

Changing Database User Settings for Windows Authentication

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard.**
   The **Welcome to the AVEVA Work Tasks Farm Configuration Wizard** appears.
2. Click **Next**.
   The **Edit AVEVA Work Tasks Farm Settings** screen appears.
3. Select **Database User Settings**.
4. Click **Next**.
   The Database User Settings screen is displayed.
5. Enter the updated User Name, Password and Domain.
6. Click **OK**. A message to ensure that the login has required permissions is displayed.
7. Click **OK**. A message indicating Database User settings edited successfully is displayed.
8. Click **OK**.

Changing SQL User Credentials for SQL Authentication

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Farm Configuration Wizard**.
   The **Welcome to the AVEVA Work Tasks Configuration Wizard** appears.
2. Click **Next**.
   The **Edit AVEVA Work Tasks Farm Settings** screen appears.
3. Click **Database Settings**.
   The **Connect to existing database** screen appears.
4. From the Connection String option, modify the User Id and Password for connecting to the database.
5. Click **Test Connection**.
6. Click **OK**.
7. Click **Next**. The user credentials are updated.

Changing Password of SQL User Credentials used to Connect to Repository Data Source

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.
   The **AVEVA Work Tasks Central Configuration** page appears.
2. Click **Datasource Settings**. The **Datasource List** appears.
3. From the **Datasource** column, click the required data source to be modified. The **Manage Datasource** page appears.
4. From the **Connection String** option, modify the User ID and Password field.
5. Click **OK**. This will change the password of SQL user credentials.

Changing a Web Service Authentication User

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. From the Resource Provider drop down box, select the authentication provider for the AVEVA Work Tasks Repository selected from the previous step.
2. Enter the Login Name and Password.
3. Click Authenticate. After successful authentication, the screen to Re-Authenticate users appears.

You can Re-Authenticate with a different user credential if the authenticated user details does not have the required permissions.

You can either select, create or edit an impersonation user from the Select User drop down box, Create New User or Edit User option respectively. Follow the steps below to:

- Create/add a new impersonation user
- Edit an impersonation user

Adding the Impersonation User

The following steps will assist you to add a new impersonation user.

1. Click **Create New user**.
2. Enter **Display Name**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Enter **Login Name** and **Password**.
4. Click **Save**.

A windows authenticated login name and Password should be provided. The login name must be preceded with the domain name.

Editing the Impersonation User

Using Enterprise Console

The following steps can be used to create or modify an impersonation user using the Enterprise Console.

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **Central Configuration**.

   The **AVEVA Work Tasks Central Configuration** page appears.
2. Click **Repository Settings**. The **Repository List** page appears.
3. From the Site Path column, click on the required Repository.
4. Log on to the Enterprise Console.
5. On the Enterprise Console menu, point to Settings, and then click Impersonation User.
6. Click **New** to create a new user. The **New User** screen appears.

   Alternatively, to edit an existing user select the required user and click **Edit**, or alternatively select **Edit User** from the context menu.
7. Enter the **Display Name**, **Account Name**, and **Password**.
8. Click **Save**. A message indicating Impersonation user saved successfully appears.

   If the Password is modified in the Active Directory then the same should be modified using the Enterprise Console or SharePoint. If this is not done the workflows will fail. The changes will take affect only after 3

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

minutes in the Workflow Engine.

9. Repeat Step 2 to Step 10 for all repositories where the impersonation user is used.
10. Reset IIS to clear the cache.

Using Microsoft SharePoint Central Administration

1. Select the required user to be edited from the **Select User** drop down box.
2. Click **Edit User**.
3. Modify the password by entering the new password.
4. Click **Save**. The changes are updated in the Database.

Users can also create or edit impersonation user through the Enterprise Console.

Error while Installing Configuration Farm or Web Sites

Before going through this section, see Prerequisites for Debugging Exceptions.

1. During install, you get the following error message:

Error: Cannot read configuration file

Probable Cause

This error occurs if the AVEVA Work Tasks sites, namely the Central Configuration and the Enterprise Console, exists in Internet Information Services (IIS) Manager.

Resolution

1. Open Internet Information Services (IIS) Manager.
2. Expand Web Sites.
3. Expand AVEVAWorkTasksSites.
4. Delete AVEVA Work Tasks sites, namely the Central Configuration and the Enterprise Console.
5. Install AVEVA Work Tasks based on your installation scenario again.

2. Websites created by the Farm Configuration wizard are not getting deleted while uninstalling Farm Configuration Wizard.

Probable Cause

This occurs if the the user creates a website with the same name as that of the parent website which was created by the Farm Configuration Wizard using the New Website feature in IIS Structure.

Example:

1. If Farm Configuration Wizard had a site which was created using the New Website feature in IIS with structure the name Telematics.
2. Later the user creates a Enterprise Console or Central Configuration or Virtual Directory with the name Telematics under the parent website Telematics (created in step 1).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Resolution

Ensure that the parent site is removed from the IIS.

Error while reinstalling AVEVA Work Tasks Client

Issue 1:

The AVEVA Work Tasks Server/Client configuration fails while using Farm Configuration Wizard, displaying an error message: "*The given key was not present in the dictionary.*"

Cause:

This issue occurs if **AVEVAWorkTasksSites** is already present in IIS before configuring the farm. In this scenario, an entry for AVEVAWorkTasksSites is not created in the SKWebSite table during the configuration process.

Solution:

To resolve this issue, follow these steps:

1. Open **IIS Manager**.
2. Locate and delete the **AVEVAWorkTasksSites** site from IIS.
3. Use the **Farm Configuration Wizard** to reconfigure the farm. Ensure that a new Farm Database is used during the configuration process.

Issue 2:

When clicking the **Edit** icon in Enterprise Console under Farm Configuration, the following error is displayed: *Index was out of range. Must be non-negative and less than the size of the collection. Parameter name: index*

Cause:

This issue occurs if **AVEVAWorkTasksSites** is already present in IIS before configuring the farm. In this scenario, an entry for AVEVAWorkTasksSites is not created in the SKWebSite table during the configuration process.

Solution:

To resolve this issue, follow these steps:

1. Open **IIS Manager**.
2. Locate and delete the **AVEVAWorkTasksSites** site from IIS.
3. Use the **Farm Configuration Wizard** to reconfigure the farm. Ensure that a new Farm Database is used during the configuration process.

Unable to Log on to Enterprise Console

Issue

When you log on to Enterprise Console, a message Unable to fetch provider appears and you will not be able to log on to Enterprise Console.

Probable Cause

The computer on which AVEVA Work Tasks is installed has a underscore (_) in the name. When underscore is used in the machine name, the cookies are not generated.

Resolution

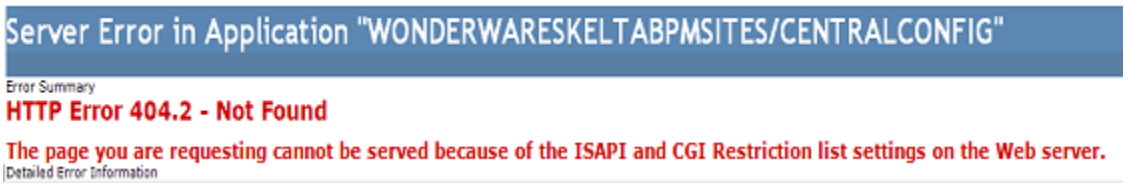
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Use the IP Address of the system on which AVEVA Work Tasks is installed to access Enterprise Console and Microsoft SharePoint. Do not use a computer name as it has an underscore (_).

Server Error in Application

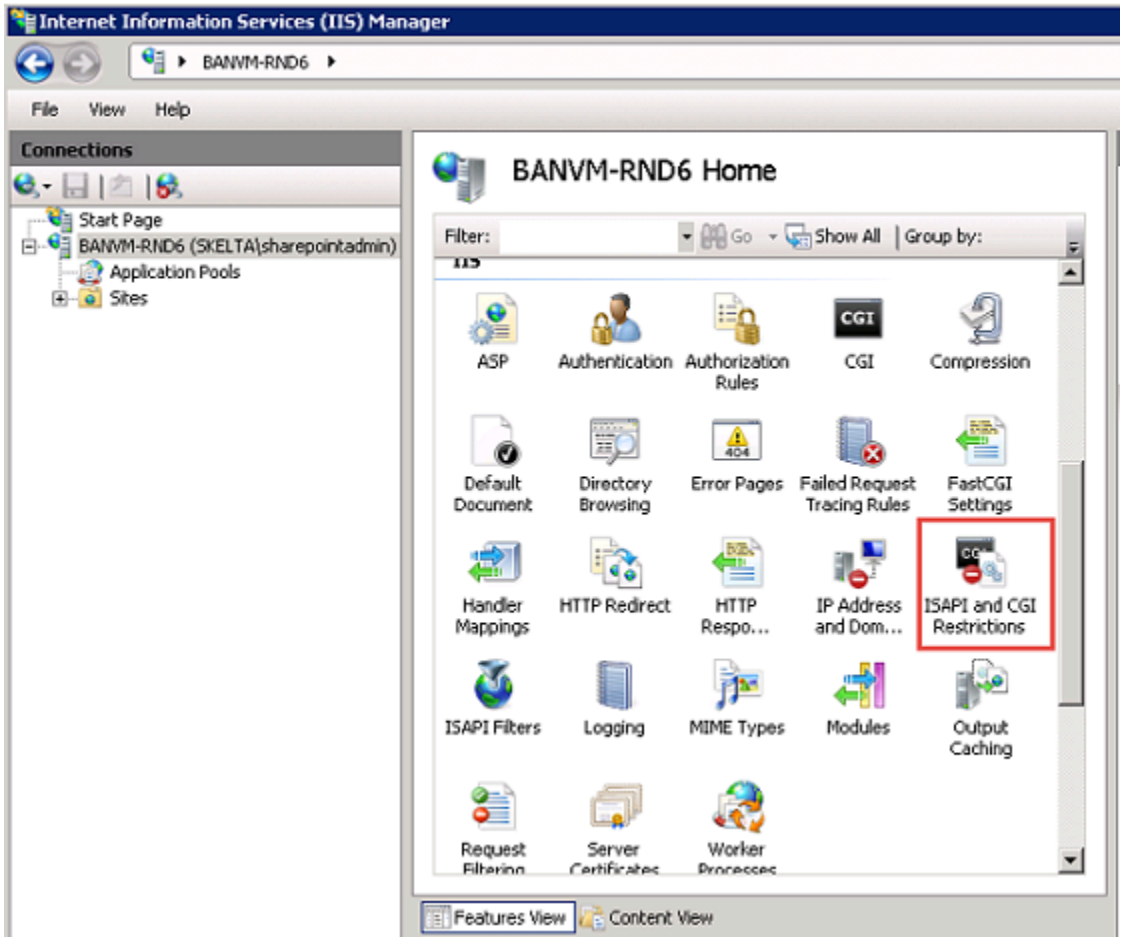
Issue

In a fresh installation environment, after configuring the AVEVA Work Tasks sites, when you try to open Central Configuration, you get the following error message:



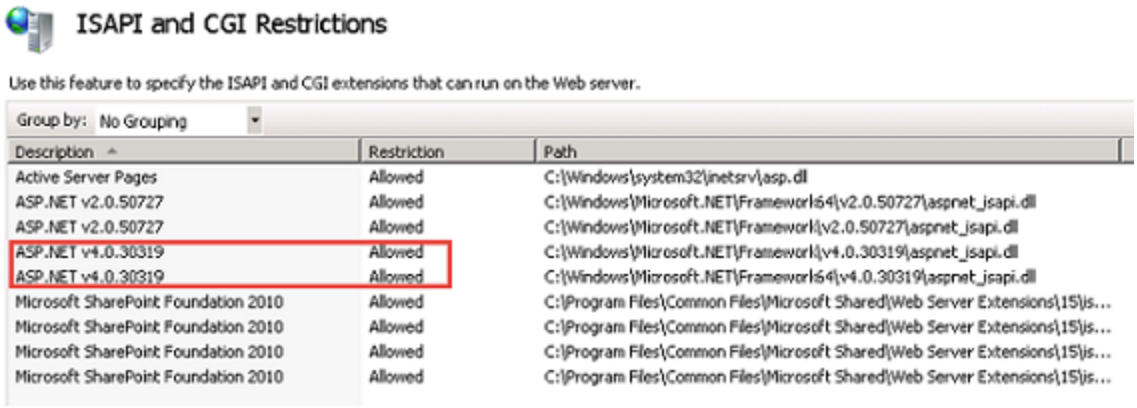
Resolution

1. Open the IIS Manager, and then go to server level.



2. In the Features View, click ISAPI and CGI Restrictions. The ISAPI and CGI Restrictions window opens.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Change the Restriction of ASP.Net v4.0.30319 to Allowed.

Troubleshooting AVEVA Work Tasks Upgrade

Issue:

Could not load file or assembly 'Microsoft.SqlServer.BatchParser

While creating a DataSource, the following error message appears:

Could not load file or assembly 'Microsoft.SqlServer.BatchParser, Version=12.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91' or one of its dependencies. The system cannot find the file specified.

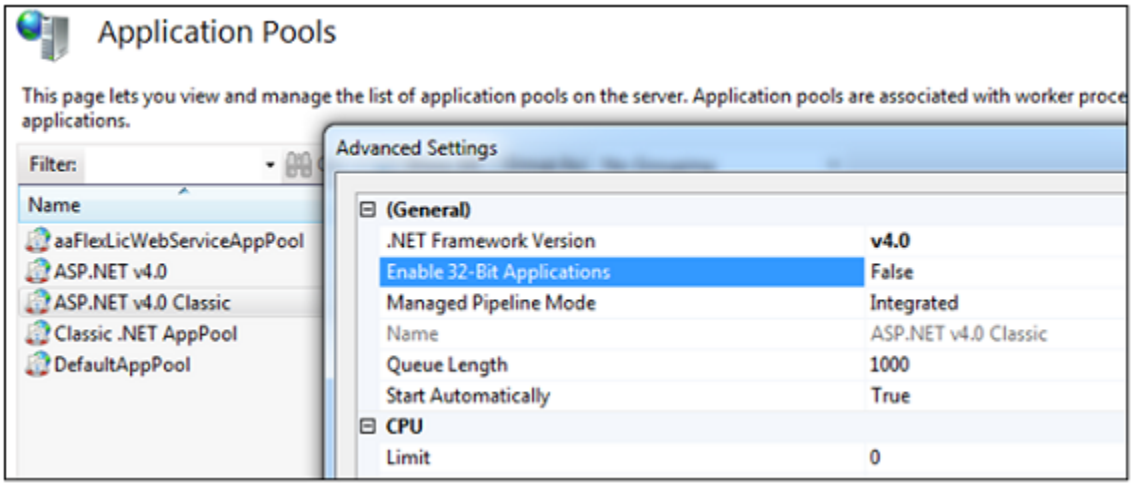
Probable Cause

The Application runs under different processor operating mode than the installed operating system. For example, on a 64Bit OS application runs in 32 bit mode.

Corrupt or wrong SMO installation.

Resolution

- Ensure that the correct SMO and CLR Type are installed in the system. For example, for 64-bit OS, install 64-bit SMO and 64-bit CLR Types.
- If the OS is 64-bit, set Enable 32-Bit Application flag to False.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

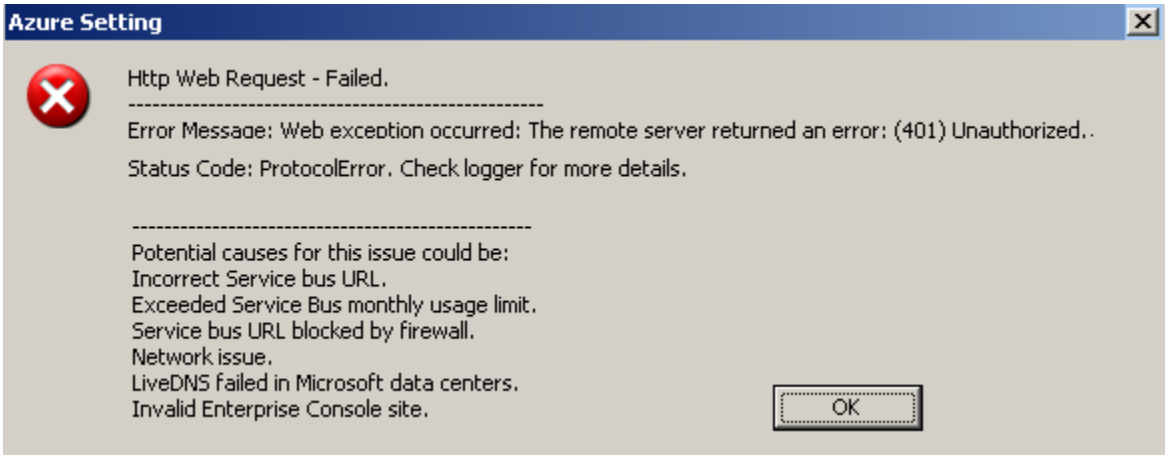
Service Bus Listener Error

1.The exception *Could not connect to workflow engine* appears on click of Test.

Resolution:

- In a Standalone environment, ensure the Workflow engine is running.
- In an Enterprise server environment, ensure the Advance Server Service is running in the server where the test is performed and ensure that the Workflow engine is running in at least one of the servers in the environment.

2. Web Exception occurred: The remote server returned an error: (401) Unauthorized..Status Code: ProtocolError. Check logger for more details.



Probable Cause:

Anonymous access is not enabled for the Enterprise Console in IIS 6.

Resolution:

1. Access IIS Manager.
2. Click **AVEVAWorkTasksSites**, and then click **Enterprise Console**.
3. Right-click **Enterprise Console** and select **Properties**. The **Properties** dialog box appears.
4. Click the **Directory Security** tab, and then click **Edit** in **Authentication and Access Control**. The **Authentication Mode** appears.
5. Select **Enable Anonymous access**, and enter the credentials of the administrator.
6. In **Authenticated Access**, select **Basic authentication** (password is sent in clear text).
7. Click **OK**.
8. Test the Service Bus Listener.

Mobile Notification Service Errors

Issue:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Notifications not received on the mobile device.

Resolution:

1. Verify if Push Notifications were allowed at the time of installation.
   - If Push Notifications were not allowed during installation, then reinstall AVEVA Work Tasks Pro.
   - If Push Notifications were allowed, then go to Notification Settings in your mobile device, and then enable Sound and Badge notification.

   Verify if notifications are received. If notification is still not received follow the next set of steps.
2. Verify the AVEVA Work Tasks log files, verify all the exceptions that start with the text 'AMNS_' and all exceptions form the process ArchestrAMobileNotificationService.exe.
   - Contact the support team with the exception. For more information about the logger console, see *Logger Console* section in the *Developer Guide*.
   - If exceptions are related to connectivity follow the next set of steps.
     To verify the connectivity between AVEVA Work Tasks server and Mobile Notification Service on the cloud, complete the following steps:
3. Open Farm Configuration Wizard.
4. Click **Next**.
5. Select **Service Settings** to open the **Services/Global Settings** screen.
6. Click **Next**.
7. Click against Mobile Notification Service to open the Mobile Notification Service Settings screen.
8. Go to the **Register** tab.
9. Click **Test Connectivity** in the **Register** section.
   - If connectivity test fails, check the network connection of the Server or contact your network administrator.
   - If connectivity passes, send a test notification to verify if notifications are successfully sent.
10. To send test notifications,
    a. In the **Registration Details** section, click **Test Notification**.

       The **Test Notification** dialog appears.
    b. Select a mobile device and click **Send Notification**.

    If the test notifications are not sent, view the Logger console for specific logs from the process ArchestrAMobileNotificationService.exe. Contact the support team with the exception.

Error: Could not load file or assembly or Error while creating instance

Issue:

After upgrading to AVEVA Work Tasks, you might get the following error:

Could not load file or assembly

or

Error while creating instance

Resolution:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If custom assemblies are used for any custom activities, SOA assembles, custom event handler, custom resource provider, or any custom components, you must upgrade the custom assemblies to .NET framework supported version to work with the latest version of product.

Rebuild the custom assemblies with the latest version of product and with .NET framework supported version. Ensure to update the assembly in the appropriate path. If custom assemblies are in GAC, you must update the assembly in the GAC too as GAC for different versions of framework are different.

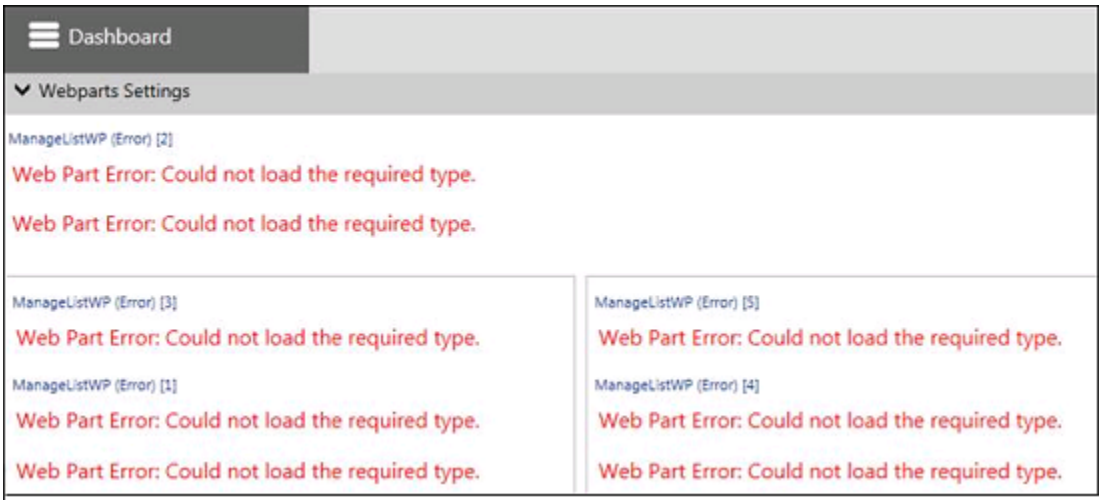
Web Part Error: Could not load the required type

Issue:

In **Wonderware Skelta BPM 2014** release onwards, when we create a new **Enterprise Console** and map an upgraded repository (upgraded from **Skelta BPM.NET 2011 SP4**) to the new site, then the following error appears while logging in to the repository:

Web Part Error: Could not load the required type

The following image shows the error snapshot:



Resolution:

Adding the following code to the **web.config** file of the newly created **Enterprise Console**:

```
<runtime>
<assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
<dependentAssembly>
<assemblyIdentity name="Skelta.WebParts" culture="neutral"
publicKeyToken="bc10944fa2f62aa6" />
<bindingRedirect oldVersion="5.0.0.0" newVersion="6.0.0.0" />
</dependentAssembly>
</assemblyBinding>
</runtime>
```

File Watcher Events Error

Issue:

The **File Watcher Workflows** configured using the existing **File Watcher Folder** are not triggered (even if the folder is pointing to the correct physical location) in the following scenarios:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- When you back up and restore the Farm database and the Data Source
- When you create a Repository using the **Attach Data Source** option

Resolution:

1. Edit and save the **File Watcher Folder**.
2. Restart all the AVEVA Work Tasks Services.

Access Denied to AVEVA Work Tasks Temp Folder along with System IO Errors

Issue

User receives an error while accessing **Central Configuration** or **Enterprise Console** and **Logger Console** logs System IO errors accessing specific file in **AVEVA Work Tasks Temp** folder (**C:\Windows\Temp\AVEVAWorkTasksTemp**).

Probable Cause

Access Control List (ACL) updated by Windows updates or policies.

Resolution

Provide access to **AVEVA Work Tasks Temp** Folder (**C:\Windows\Temp\WorkTasksTemp**):

- Delete and recreate **AVEVA Work Tasks Temp** folder.

  OR
- Provide access to account **ApplicationPoolIdentity**.

  OR
- Provide access to group **Everyone**.

  Note: This option is not recommended as it will leave your system open to security vulnerabilities.

References

For more information on:

- Application Pool Identities, see https://docs.microsoft.com/en-us/iis/manage/configuring-security/application-pool-identities
- Access Controls, see https://technet.microsoft.com/en-us/library/dd277411.aspx

Cache Refresh not Reflected after the Product Upgrade

The changes in the Enterprise Console like the js, css, images, etc. files are not getting reflected after the product

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

is upgraded or a package(which includes overwriting existing files) is installed.

Probable Cause:

- You have not cleared the browser cache after the product upgrade.
- The time (in seconds) set for `cacheControlMaxAge` attribute in the Web.Config file located in <product installation path>\Web\BPMUITemplates\Default\ of the Enterprise Console is not reached.

Resolution:

- After the product upgrade, delete the browser cache manually on the client browser. You need to clear the browser cache only once.
- If you change the .js, .css, or image files, the changes would reflect after the time specified for `cacheControlMaxAge` attribute.

For more information, see Cache Refresh Settings.

Unable to Re-install AVEVA Work Tasks

Issue

Unable to re-install AVEVA Work Tasks 2020 on a server after uninstalling it from the server.

Resolution

- Check if the Platform Common Services (PCS) is installed. You may check this in Add/Remove Programs.
- Restart ArchestrA Data Store service from Services Applet. It has dependent services, and those also need to be restarted.

Unexpected Error when Authenticating against Azure Active Directory

Issue

AVEVA Work Tasks does not work as expected if TLS 1.2 is not used to communicate or authenticate against Azure Active Directory.

Probable Cause

Microsoft Azure Active Directory does not support Transport Layer Security (TLS) protocols, TLS 1.1 and TLS 1.0.

Resolution

You need to enable support for TLS 1.2 in your environment. For more information, refer microsoft documentation on enabling support for TLS 1.2.

Swapping Repository Provider

You can swap the primary provider records to the secondary provider in SkVirtualActor table when the primary provider is not accessible or damaged.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Swapping is possible only if the repository you are accessing has multiple providers configured.

Note: Before you swap the provider, take the backup of SkVirtualActor table.

Issue

The repository provider you are accessing is damaged and not accessible.

Solution

Swap the damaged repository provider with the other provider available for that repository.

To swap the repository provider

1. Access **Central Configuration** site and navigate to **Repository Provider List**. For more information, see Repository Provider List.
2. Right-click the repository provider that is damaged and select the option **Swap Provider**.

   Note: The **Swap Provider** option is available only if the repository has more than one provider instances.

   The values get swapped from the primary provider to the secondary provider. The secondary provider is the one which is at the top of the provider list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Connectors and Extensions User Guide

This section is for the system integrator responsible for installing the product and has knowledge of operating systems, relational database concepts, and the database engines in your environment. It assumes that you are familiar with the interface requirements for your supporting applications.

Key Features

The following section highlights the key features of the AVEVA Work Tasks connectors and extensions.

AVEVA Work Tasks Connector for System Platform

- Provides an IDE Extension to connect to the AVEVA Work Tasks repository.
- Provides an AppObject named WorkflowGateway for Connectivity configuration, and to route Outbound and Inbound connections.
- Provides complete process management inside IDE.
- Provides option to capture workflow events and bind the events to workflows.
- Provides option to interact (Read/Write attributes) with Galaxy through the workflow process.
- Provides ArchestrA Symbols for InTouch HMI.
- Provides the Historian EventScript library. This script library can be used to monitor triggered Historian event tags.

AVEVA Work Tasks Extension for System Platform

- Installs System Platform Plug-in on AVEVA Work Tasks server. The plug-in is required for Galaxy to connect to the AVEVA Work Tasks workflow repository.
- Modifies the user interface files for rendering the InTouch controls and to render the Enterprise Console functionality in IDE.

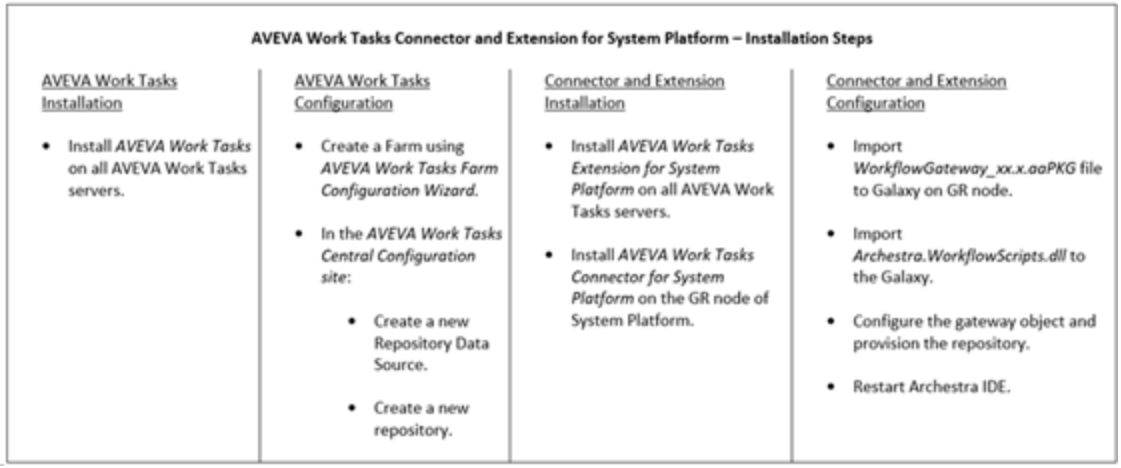
AVEVA Work Tasks Extension for OPC UA

- You use the AVEVA Work Tasks Extension for OPC UA to read or write tag values from the OPC UA server.
- Connect AVEVA Work Tasks workflow server to one or more existing OPC UA servers on the network.
- Read or Write Tag values from the OPC UA server which is connected to the AVEVA Work Tasks server.

Overview

The following image shows the installation sequence for AVEVA Work Tasks Connector and Extension for System Platform:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Step 1: AVEVA Work Tasks Installation

- Install AVEVA Work Tasks on all the servers.

Step 2: AVEVA Work Tasks Configuration

- Configure Workflow in AVEVA Work Tasks. For more information about configuring workflow, see *Administrator Guide*.

Step 3: AVEVA Work Tasks Extension and Connector for System Platform Installation

1. Install the AVEVA Work Tasks Extension for System Platform on all AVEVA Work Tasks installed computers (servers or clients), and verify the installation.
2. Install the AVEVA Work Tasks Connector for System Platform on the GR Node of System Platform, and verify the installation.

Alternatively,

AVEVA Work Tasks Connector and Extension for Information Server Installation

1. Install the AVEVA Work Tasks Extension for Information Server on all AVEVA Work Tasks Servers, and verify the installation.
2. Install the AVEVA Work Tasks Connector for Information Server on the Information Server, and verify the installation.

Step 4: Connector Configuration

1. Import *WorkflowGateway_xx.xx.aa.PKG* file in Galaxy on GR Node.
2. Import ArchestrA.WorkflowScripts.dll in the Galaxy.
3. Configure WorkflowGateway object in the Galaxy.
4. Provision a repository in AVEVA Work Tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

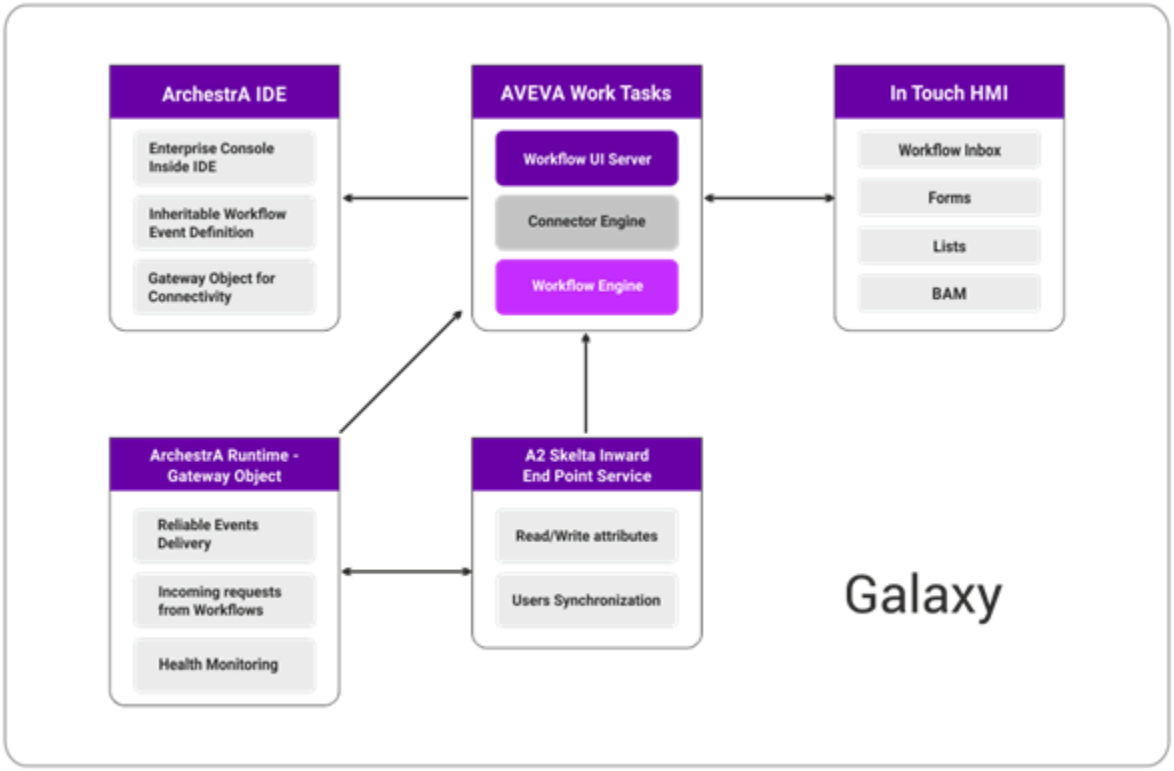
Connector for System Platform

The System Platform contains an integrated set of services and an extensible data model to manage plant control and information management systems. The System Platform supports both the supervisory control layer and the manufacturing execution system (MES) layer, presenting them as a single information source.

Modular applications sit on top of the System Platform. AVEVA Work Tasks and other third-party integrators are delivering a growing inventory of application components that use the System Platform services.

Architecture

The following image shows the architecture of AVEVA Work Tasks Connector for System Platform:



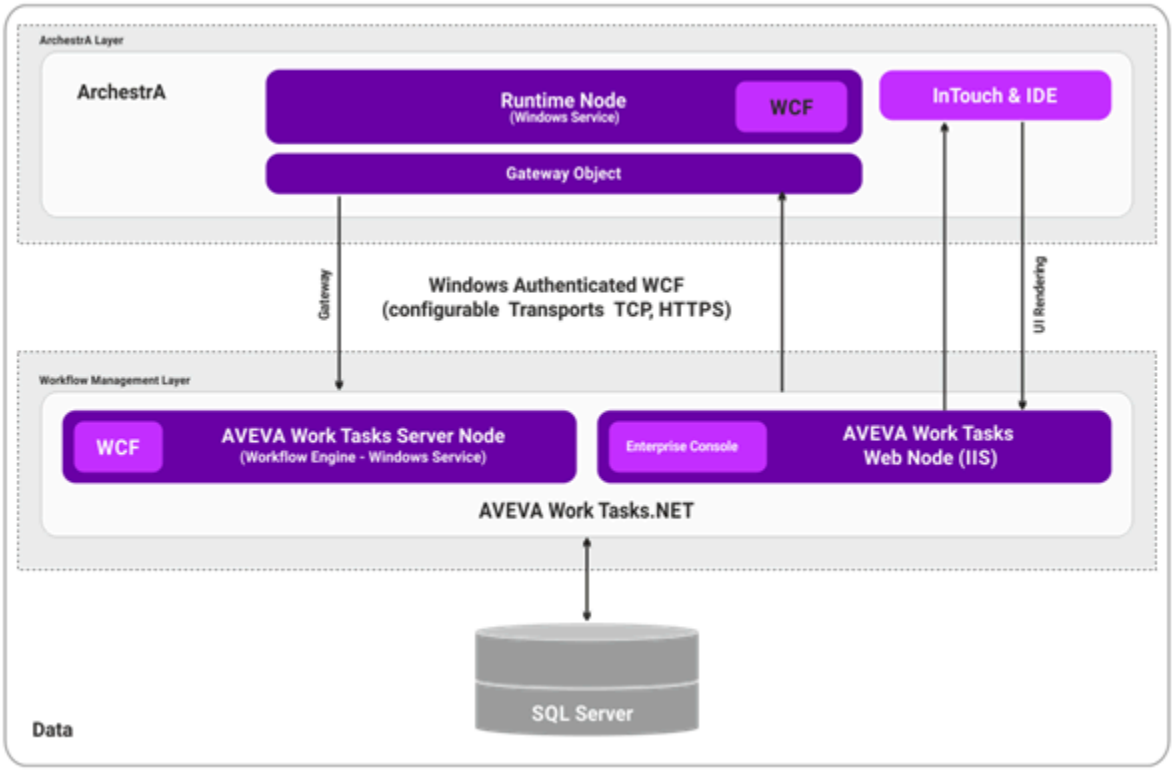
Principles of Integration with System Platform

- ArchestrA is used as IDE for contextual development of Workflows and Composite applications
- Work with Object-oriented concepts of System Platform
- Allows scalable deployments
- Allows redundancy for reliable operations
- Provides automated deployment across platforms
- ArchestrA graphics symbol for bringing process-specific UI inside HMI
- Get Events from any object
- Provides a scalable, easy to use and configurable, and inheritable model

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Enables existing users of IDE/HMI to work with workflows

High Level View of AVEVA Work Tasks Connector for System Platform Architecture



AVEVA Work Tasks and System Platform Integration

A WCF (Windows Communication Foundation) layer integrates the System Platform with AVEVA Work Tasks. An AppObject named WorkflowGateway, placed in the ArchestrA environment, manages the Outbound and Inbound communication between these layers.

- **Outbound connection** contains information related to the WCF service hosted inside the workflow engine (windows service). Outbound connection is used to trigger workflow during events in ArchestrA. The same connection is also used to get information from AVEVA Work Tasks layer.
- **Inbound connection** contains information related to the WCF service hosted inside the Runtime node as a windows service. Inbound connection is used to perform read/write attribute operation and also to synchronize user information from ArchestrA layer to AVEVA Work Tasks layer.

InTouch and System Platform IDE renders the Enterprise Console functionality.

Installation Scenarios

The following are the ways you can install connectors for System Platform:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Single Computer
- GR and System Platform IDE on Computer 1, InTouch and Runtime on Computer 2, and AVEVA Work Tasks on Computer 3
- GR and System Platform IDE on Computer 1, System Platform IDE on Computer 2, InTouch and Runtime on Computer 3, and AVEVA Work Tasks on Computer 4

Single Computer

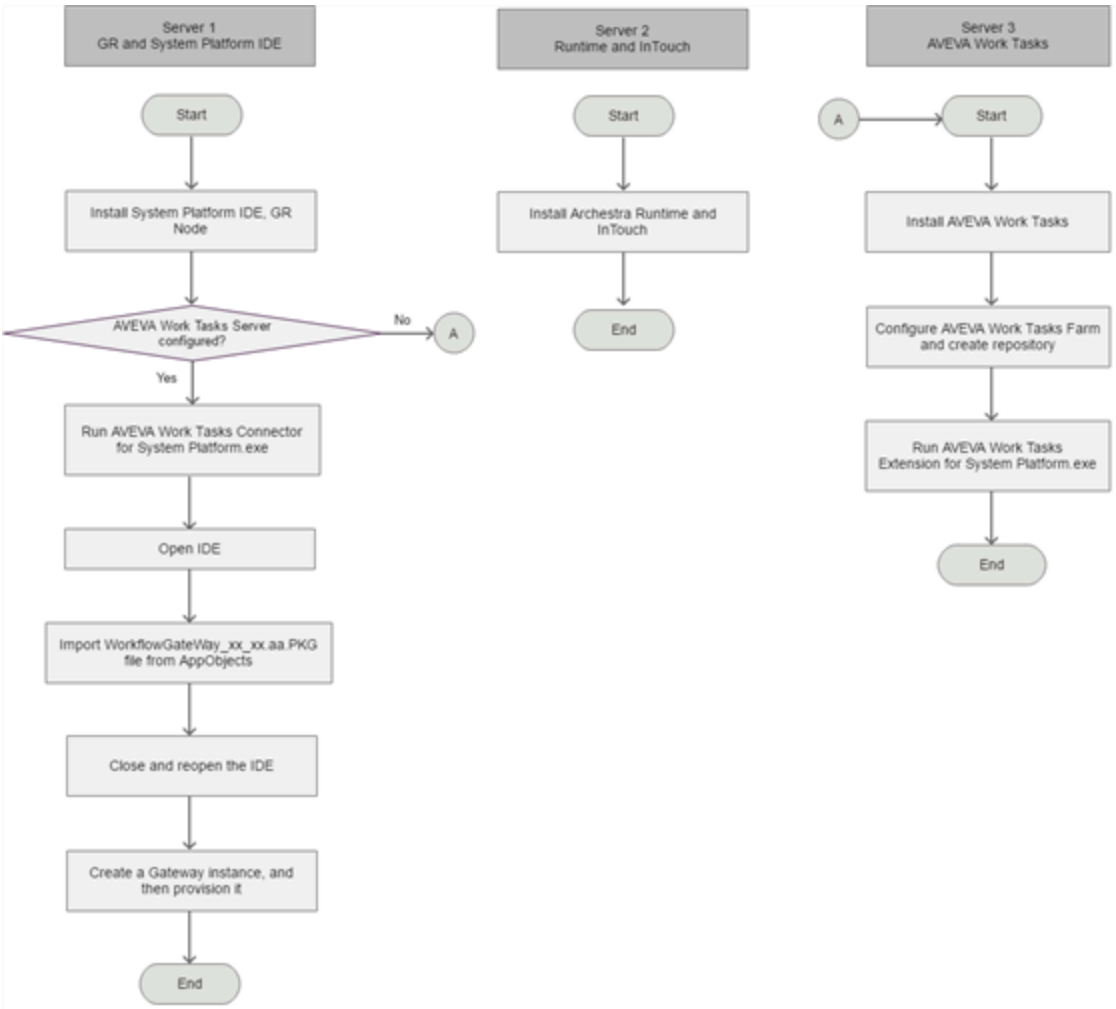
For complete details about installation steps, see the following topics:

1. Install Extension for System Platform
2. Install Extension for System Platform
3. Install Connector for System Platform

GR and System Platform IDE on Computer 1, InTouch and Runtime on Computer 2, and AVEVA Work Tasks on Computer 3

The following flowchart shows the steps to install AVEVA Work Tasks Connector for System Platform when GR and System Platform IDE are on Computer 1, InTouch and Runtime are on Computer 2, and AVEVA Work Tasks is on Computer 3.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



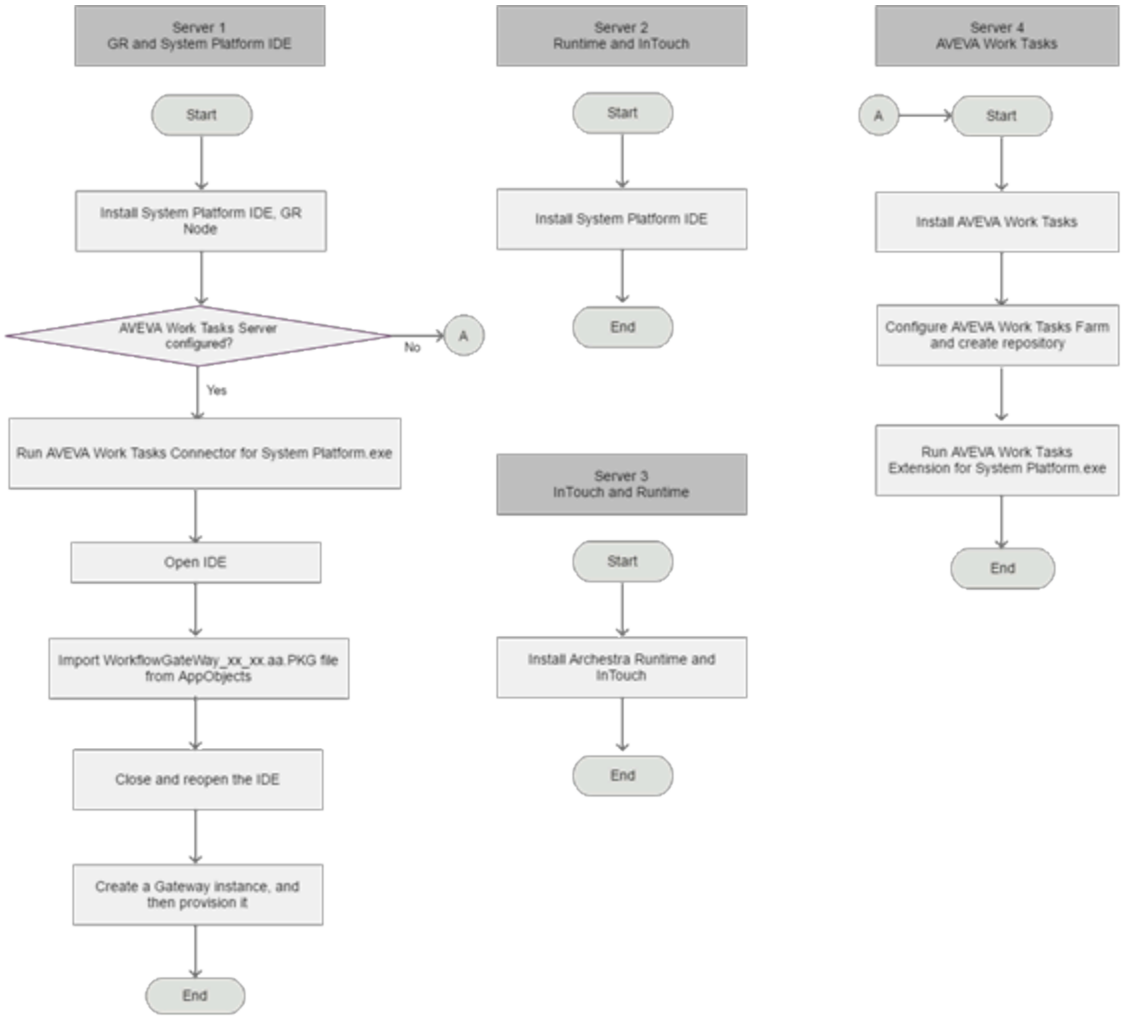
For complete details about installation steps, see the following topics:

1. Install Extension for System Platform
2. Install Extension for System Platform
3. Install Connector for System Platform

GR and System Platform IDE on Computer 1, System Platform IDE on Computer 2, InTouch and Runtime on Computer 3, and AVEVA Work Tasks on Computer 4

The following flowchart shows the steps to install AVEVA Work Tasks Connector for System Platform when GR and System Platform IDE on Computer 1, System Platform IDE on Computer 2, InTouch and Runtime on Computer 3, and AVEVA Work Tasks on Computer 4.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



For complete details about installation steps, see the following topics:

1. Install Extension for System Platform
2. Install Extension for System Platform
3. Install Connector for System Platform

Installation

AVEVA Work Tasks Connector for System Platform has a single installer for 64-bit computers. This installer should be installed on the GR Node of System Platform. It creates a Workflow folder under the ArchestrA installed path and stores the required dll files.

Note: AVEVA Work Tasks 2023 SP1 Connector for System Platform requires .Net framework 4.8 as a pre-requisite component.

The AVEVA Work Tasks Connector for System Platform plugs inside the System Platform server. It adds the following major components in System Platform IDE:

- WorkflowGateway AppObject
- AVEVA Work Tasks Enterprise Console menus and Content pages

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Adds a tab called Workflow for each and every Application Object. It is used to create events and associate workflow to those events.

In case of fresh installation, AVEVA Work Tasks 2023 SP1 Connector for System Platform is compatible with the following:

- AVEVA System Platform 2023 SP1

See the Product Compatibility for the System Platform compatibility information.

Install Connector for System Platform

To install AVEVA Work Tasks Connector for System Platform

1. Run the *AVEVA Work Tasks 2023 SP1 Connector for System Platform.exe* installer.

   The installer splash screen, and then the **AVEVA Work Tasks 2023 SP1 Connector for System Platform** screen appears.
2. Click **Next**.

   The License Agreement screen appears.
3. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.

   The Customer Information screen appears.
4. Enter the **User Name** and **Organization** name, and then click **Next**.
5. Click **Install** to begin the installation.
6. Click **Finish** to complete the installation.
7. Restart the System Platform IDE.

Installing the Connector for System Platform updates the following folders:

- The Object file **aaPKG** or **aaPDF** is saved in ArchestrA Installed Path > Workflow folder > AppObjects folder. This object needs to be imported to System Platform IDE.
- The **IDE extension DLLs** are stored in the IDE extension folder, **InTouch DLLs** in the InTouch folder, and **Script DLLs** in the Script Library folder.

Import files to System Platform IDE

After installing the Connector for System Platform, you have to import the following files:

- The WorkflowGateway object file, aaPKG or aaPDF
- ArchestrA.InTouch.SkeltaControls.dll to enable workflow features in InTouch HMI
- Script function library (ArchestWorkflowScripts.dll)
- AVEVA Work Tasks controls for InTouch OMI

To import WorkflowGateway Object file

The aaPKG or aaPDF file saved in the AppObjects folder has to be imported to System Platform IDE. Complete the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

following steps to import the files:

1. Open **System Platform IDE**.
2. From the **Galaxy** menu, select **Import** > **Objects** > **From Package**.

   The Import Objects from package window appears.
3. Browse to the path <**ArchestrA Installation Path**>**ArchestrA\Workflow\AppObjects**, select **WorkflowGateway_XX.X.aaPKG**, and then select **Open**.

   The Import Preferences window appears.
4. Click **OK**.

   The status of importing the object appears in the Import Automation Object(s) window.
5. After importing the object completes, click **Close**.
6. Close the System Platform IDE, and then open it again.

On importing the object to System Platform IDE

- The following dll files listed are stored in ArchestrA Installation folder > Framework > FileRepository > CustomCategoryUIFiles folder.
  - ArchestrA.Workflow.IDEExtension.dll
  - Skelta.ArchestrA.WinForms.dll
  - Skelta.WinForms.dll
  - Skelta.WinForms.Security.dll

    If these dll files are not added automatically, you must copy the files manually from **C:\Program Files (x86)\ArchestrA\Workflow\IDEExtension** to:
  - C:\Program Files (x86)\ArchestrA\Framework\FileRepository\CustomCategoryUIFiles
  - C:\Program Files (x86)\ArchestrA\Framework\Bin\IDEExtensions

- Verify whether the files stored in CustomCategoryUIFiles folder are copied to ArchestrA installed path > Framework > Bin >IDEExtensions folder. If not, they can be copied manually. The Workflow tab in System Platform IDE > Workflow - Toolbox tab will not be visible without these DLL files.

- Import ArchestrA.InTouch.SkeltaControls.dll to use additional controls that enables you to use workflow features in InTouch. For information about InTouch Controls, see Workflow Status View Inside InTouch.

To import Script Function Library (ArchestrAWorkflowScirpts.dll)

1. Open **System Platform IDE**.
2. From the **Galaxy** menu, select **Import** > Script Library.

   The Import Script Library window appears.
3. Select the **<Installation Path>\ArchestrA\Workflow\Scripts** folder.
4. Select **ArchestrAWorkflowScripts.dll** file, and then click **Open**.

To import AVEVA Work Tasks controls for InTouch OMI

1. Open **System Platform IDE**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. From the **Galaxy** menu, select **Import** > **Visualization** > **Client controls**.
   The **Import Client Controls** dialog box appears.
3. Select the **<Installation Path>\ArchestrA\Workflow\ClientControls** folder.
   The imported controls appear in the **Graphic Toolbox**.

To import ArchestrA App

1. Open **System Platform IDE**.
2. From the **Galaxy** menu, select **Import** > **Visualization** > **OMI App**.
   The **Import ArchestrA App** dialog box appears.
3. Select the **<Installation Path>\ArchestrA\Workflow\InTouchOMI\AVEVAWorkTaskApp** folder.
   The imported ArchestrA App appears as **AVEVAWorkTasksAppXXX** in the **Graphic Toolbox**.

Note:
- Import the ArchestrA App only once as System Platform IDE allows ArchestrA App imports multiple times by creating multiple objects with an increasing sequence number like AVEVAWorkTasksAppXXX. This may create discrepancy.

Install Connector for System Platform in Silent Mode

You can install the Connector for System Platform in silent mode using the **AVEVA Work Tasks** Connector for System Platform installer with switches for silent mode.

The installation of the Connector for System Platform in silent mode considers user inputs from command-line parameters.

The full syntax for running the AVEVA Work Tasks Connector for System Platform installer from the command line is as follows:

<Installer EXE Name>.exe /S /v/qn

For example: "AVEVA Work Tasks 2023 SP1 Connector for System Platform.exe" /S /v/qn

where,

- /S denotes silent install
- /v denotes pass arguments to Msiexec (Basic MSI and InstallScript MSI projects). The /v option is used to pass command line options and values of public properties through to Msiexec.exe.
- /qn denotes an msiexec switch indicating to run with no user interface (UI).

This command string takes the default values for parameters, such as user name, installation location, and company name.

Silent installation with logging

If you want to install in silent mode with logging, then enter the syntax for silent installation command as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

<Installer EXE Name>.exe /S /v/qn /v"/L*v <Log File Path and Name>"

For example: "AVEVA Work Tasks 2023 SP1 Connector for System Platform.exe" /S /v/qn /v"/L*v c:\logfileName.log"

Return Results

The return result is stored in the **%ERRORLEVEL%** variable. Its value is **0** if the installation is successful and other than zero if the installation is not successful.

To get the return result, create and run a batch file with the following commands:

```
"AVEVA Work Tasks 2023 SP1 Connector for System Platform.exe" /S /v/qn
IF %ERRORLEVEL% EQU 0 (
echo Connector silent installation completed successfully.
) ELSE (
echo Connector silent installation not completed successfully.
)
```

Note: The installer will install connector files in the ArchestrA installation path automatically. After you run the silent installer, wait for a minimum of 30 minutes until the installation is complete. Check the installation folder to ensure that all the required files and folders are copied to the installation folder.

Verify Installation

The verification of the Connector for System Platform installation has been broadly divided into the following three steps.

Verify basic AVEVA Work Tasks Connector for System Platform installation

A folder "Workflow" gets created in the ArchestrA installed root folder. The user should be able to perform the following tasks successfully:

1. Import the **WorkflowGateway (WorkflowGateway_20.1.aaPKG)** from the **<ArchestrA installed root >\workflow\AppObjects** folder.
2. Create an instance of WorkflowGateway.
3. Open the WorkflowGateway and provide the AVEVA Work Tasks server information and authentication credentials.

Verify configuration of WorkflowGateway AppObject

1. Ensure that the Windows authentication credentials provided for AVEVA Work Tasks Server Connectivity (Outbound Connection) are valid.

   Click on the **Test Connection** button in WorkflowGateway to populate the workflow repositories configured in the AVEVA Work Tasks server.
2. Ensure that Workflow Repository authentication information is valid.

   Click on the **Authorize and Provision** button for provisioning the repository.
3. Ensure that *WorkflowGateway* AppObject has been provisioned successfully.

   Select a *WorkflowGateway* instance from the Workflow tool box drop-down list on the left upper corner

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

of IDE to list different menu options in the drop-down list, based on the logged-in user rights.

Verify runtime functionality of WorkflowGateway AppObject

- Verify if WorkflowGateway is deployed successfully on the RunTime node.

The WorkflowGateway attribute *GatewayObjectDeployErrMsg* should have the value *NoError* and the attributes *InvalidInboundConnAuthInfo* and *InvalidInboundConnPortInfo* should have the value as false after deployment.

Upgrade Connector for System Platform

You can upgrade to **AVEVA Work Tasks 2023 SP1 Connector for System Platform** from the following version:

- AVEVA Work Tasks 2023 Connector for System Platform

Prerequisites

- In the AVEVA Work Tasks server, upgrade **AVEVA Work Tasks 2023** to **AVEVA Work Tasks 2023 SP1**.
- Undeploy the Platform and Workflow Gateway Object.
- Upgrade the Connector to the latest version.

Steps to Upgrade

Importing files to System Platform IDE

After installing the Connector for System Platform, you have to import the following files:

- The WorkflowGateway object file, aaPKG or aaPDF
- ArchestrA.InTouch.SkeltaControls.dll to enable workflow features in InTouch HMI
- Script function library (ArchestWorkflowScripts.dll)
- AVEVA Work Tasks controls for InTouch OMI

The aaPKG or aaPDF file saved in the AppObjects folder has to be imported to System Platform IDE. Complete the following steps to import the files:

1. Open **System Platform IDE**.
2. Select **Galaxy** menu, select **Import**, and then select **Object(s)**.

   The Import Automation Object(s) window appears.
3. Go to ArchestrA Installed Path, select **Workflow**, select **AppObjects**, and then select the **WorkflowGateway_XX.X.aaPKG** file.
4. Click **Open**.

   The Import Preferences window appears.
5. Click **OK**.

   The status of importing the object appears in the Import Automation Object(s) window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. After importing the object completes, click **Close**.
7. Close the System Platform IDE, and then open it again.

On importing the object to System Platform IDE:

- The following dll files listed are stored in ArchestrA Installation folder > Framework > FileRepository > CustomCategoryUIFiles folder.
  - ArchestrA.Workflow.IDEExtension.dll
  - Skelta.ArchestrA.WinForms.dll
  - Skelta.WinForms.dll
  - Skelta.WinForms.Security.dll

    If these dll files are not added automatically, you must copy the files manually from **C:\Program Files (x86)\ArchestrA\Workflow\IDEExtension** to:
  - C:\Program Files (x86)\ArchestrA\Framework\FileRepository\CustomCategoryUIFiles
  - C:\Program Files (x86)\ArchestrA\Framework\Bin\IDEExtensions

- Verify whether the files stored in CustomCategoryUIFiles folder are copied to ArchestrA installed path > Framework > Bin >IDEExtensions folder. If not, they can be copied manually. The Workflow tab in System Platform IDE > Workflow - Toolbox tab will not be visible without these DLL files.

- Import ArchestrA.InTouch.SkeltaControls.dll to use additional controls that enables you to use workflow features in InTouch. For information about InTouch Controls, see Workflow Status View Inside InTouch.

- Import Script Function Library (ArchestrAWorkflowScirpts.dll)
  1. Open **System Platform IDE**.
  2. Select **Galaxy** menu, select **Import**, and then select **Script Function Library**.

     The Import Script Function Library window appears.
  3. Go to ArchestrA Installed Path, select **Workflow**, and then select **Scripts**.
  4. Select **ArchestrAWorkflowScripts.dll** file, and then click **Open**.

To import ArchestrA App:

1. Open **System Platform IDE**.
2. Select **Galaxy** menu, select **Import**, and then select **ArchestrA Apps**.

   The **Import ArchestrA App** dialog box appears.
3. Select the **<Installation Path>\ArchestrA\Workflow\InTouchOMI** folder.

   The imported ArchestrA App appears as **AVEVAWorkTasksAppXXX** in the **Graphic Toolbox**.

Note:
- Import the ArchestrA App only once as System Platform IDE allows ArchestrA App imports multiple times by creating multiple objects with an increasing sequence number like AVEVAWorkTasksAppXXX. This may create discrepancy.

- When you upgrade the product, it will not overwrite/upgrade the existing AvevaWorkTasksApp and it may

---

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

create issues related to old and new versions. Hence it is recommended to delete the existing AvevaWorkTasksApp and then import the new one.
Also, make sure that the older/existing AvevaWorkTasksApp is not being used in any of the ViewApp/Layout, etc. otherwise, it will cause some warnings in the logger.

Uninstall Connector for System Platform

To uninstall AVEVA Work Tasks Connector for System Platform

1. Delete all the *WorkflowGateway* instances and templates.
2. Close all the IDEs.
3. Uninstall the Connector for System Platform.

Impact of AVEVA Work Tasks Connector for System Platform uninstallation

Following is the list of files which will be deleted from the respective folders:

- ArchestrA > Workflow > IDEExtention folder
  - ArchestrA.Workfow.IDEExtention.dll
  - ArchestrA.WorkflowScripts.dll
  - Skelta.ArchestrA.Winforms.dll
  - Skelta.Winforms.Security.dll
- ArchestrA > Workflow > AppObjects folder
  - WorkflowGateway_20.1.aaPKG
- ArchestrA > Workflow > InTouch folder
  - ArchestrA.InTouch.SkeltaControls.dll
  - Skelta.Winforms.dll
  - Skelta.Winform.Security.dll
- ArchestrA > Workflow > ScriptLibrary folder
  - ArchestrA.WorkflowSripts.dll
  - Historian.EventScripts.dll
- ArchestrA install path
  - Workflow folder is deleted.
- ArchestrA > Framework > FileRepository > CustomCategoryUIFiles and ArchestrA > Framework > Bin > IDEExtensions folders
  - ArchestrA.Workfow.IDEExtention.dll
  - Skelta.ArchestrA.Winform.dll
  - Skelta.Winforms.dll

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Skelta.Winform.Security.dll
- ArchestrA.Workflowscripts.dll

Limitations

- The Workflow-Toolbox provides a list of the WorkflowGateway instances configured in Galaxy. Selecting a WorkflowGateway instance will list the Enterprise Console menu for the mapped repository in the Tree view. Clicking on the link will open the respective tab window on the right side of the IDE. Using the Workflow-Toolbox users can work with multiple repositories. However, there is no support for working with multiple repositories (WorkflowGateway instance) at the same time. Close all the open windows for a repository, before working on another repository. Switching over to one repository while having one or more windows open for a different repository could lead to termination of session. Multiple presses on the Back button on these windows is not supported.

- Authentication or the Verifier user credential used for the 'Write' Attribute should be user-created before deploying WorkflowGateway. Any user used for authentication or verification after WorkflowGateway deployment may not work and require a redeploy of WorkflowGateway.

Extension for System Platform

AVEVA Work Tasks Extension for System Platform is installed in all the servers and client machines. It will add the required assemblies and files under the AVEVA Work Tasks installation folder.

Install Extension for System Platform

Install **AVEVA Work Tasks**, before installing Extension for System Platform.

In case of fresh installation, AVEVA Work Tasks 2023 SP1 Extension for System Platform is compatible with the following:

- AVEVA System Platform 2023 SP1

Install Extension for System Platform

Steps to install AVEVA Work Tasks Extension for System Platform

If you are installing AVEVA Work Tasks in the Enterprise Server mode, install the AVEVA Work Tasks Extension for System Platform on all the servers.

1. Stop all the AVEVA Work Tasks services.
2. Run the **AVEVA Work Tasks 2023 SP1 Extension for System Platform.exe** installer.

   The installer splash screen, and then the **AVEVA Work Tasks 2023 SP1 Extension for System Platform** screen appears.
3. Click **Next**.

   The License Agreement screen appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.

   The Customer Information screen appears.
5. Enter the **User Name** and **Organization** name.
6. Under **Install this application for**, select one of the following:
   - **Anyone who uses this computer (all users):** If the application can be accessed by all the users using this computer.
   - **Only for me (localadm):** If the application can be accessed by only the local administrator of the computer.
7. Click **Next**.

   The Click Install to begin the installation screen appears.
8. Click **Install** to begin the installation.

   After the installation is complete, the InstallShield Wizard Completed screen appears.
9. Click **Finish** to complete the installation.

Files Installed by AVEVA Work Tasks Extension for System Platform:

The AVEVA Work Tasks Extension for System Platform installation copies the following files:

- DLL files to the Bin folder
- Localization files to the Locals folder
- BPMUITempates files under the Web folder
- Workflow elements files under the WorkflowElements folder

The AVEVA Work Tasks Extension for System Platform does the following:

- Creates an entry in the **SKAddInProviders** table with the name field *'ArchestrA'*.
- Creates a new record in the table **SKReleaseVersionStamp** which stores the current product release version and product name. If the table is not present, the installer will create a new table. This table will be stored in Farm Database.
- The following installed files are copied from the **Bin** directory to all the **Enterprise Console site Bin** folders:
  - ArchestrA.Workflow.EventHandler.dll
  - Skelta.WinForms.Security.dll
  - ArchestrA.Workflow.Activity.Runtime.dll
  - Skelta.Connector.CommonLibrary.dll
  - Skelta.Connector.Proxy.dll
  - Skelta.ArchestrA.Connector.Client.Proxy.dll
  - ArchestrA.Workflow.Codebehind.InTouch.dll
  - ArchestrA.Workflow.Codebehind.IDE.dll
  - Skelta.ArchestrA.Connector.Client.dll

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Install Extension for System Platform in Silent Mode

You can install AVEVA Work Tasks Extension for System Platform in silent mode using the AVEVA Work Tasks Extension for System Platform installer with switches for silent mode.

The installation of the Extension for System Platform in silent mode considers user inputs from command-line parameters.

The full syntax for running the AVEVA Work Tasks Extension for System Platform installer from the command line is as follows:

<Installer EXE Name>.exe /S /v/qn

For example: "AVEVA Work Tasks 2023 SP1 Extension for System Platform.exe" /S /v/qn

where,

- /S denotes silent install
- /v denotes pass arguments to Msiexec (Basic MSI and InstallScript MSI projects). The /v option is used to pass command line options and values of public properties through to Msiexec.exe.
- /qn denotes an msiexec switch indicating to run with no user interface (UI).

This command string takes the default values for parameters, such as user name, installation location, and company name.

Silent installation with logging

If you want to install in silent mode with logging, then enter the syntax for silent installation command as follows:

<Installer EXE Name>.exe /S /v/qn /v"/L*v <Log File Path and Name>"

For example: "AVEVA Work Tasks 2023 SP1 Extension for System Platform.exe" /S /v/qn /v"/L*v c:\logfileName.log"

Return Results

The return result is stored in the **%ERRORLEVEL%** variable. Its value is **0** if the installation is successful and other than zero if the installation is not successful.

To get the return result, create and run a batch file with the following commands:

```
"AVEVA Work Tasks 2023 SP1 Extension for System Platform.exe" /S /v/qn
IF %ERRORLEVEL% EQU 0 (
echo Extension silent installation completed successfully.
) ELSE (
echo Extension silent installation not completed successfully.
)
```

Note: The installer will install extension files in the installation path automatically. After you run the silent installer, wait for a minimum of 30 minutes until the installation is complete. Check the installation folder to ensure that all the required files and folders are copied to the installation folder.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Verify Installation

You can verify the installation of the AVEVA Work Tasks Extension for System Platform from the AVEVA Work Tasks Connector for System Platform installation.

You can check the **SKProduct** table in the Farm database for the Product name along with the version number and the Build Version. The Build Version information will be in the 6.X.XXXX format.

Uninstall Extension for System Platform

To uninstall AVEVA Work Tasks Extension for System Platform:

1. Delete all the WorkflowGateway instances and templates.
2. Close all the IDEs.
3. Uninstall AVEVA Work Tasks Connector for System Platform.
4. Uninstall AVEVA Work Tasks Extension for System Platform.
5. All the files installed inside AVEVA Work Tasks installation folder are removed.

   Note: Also, ensure that the AVEVA Work Tasks Extension for System Platform is uninstalled before uninstalling AVEVA Work Tasks.

After uninstalling the Extension for System Platform, the following files are deleted from the respective folders:

- *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder
  - ArchestrA.Connector.Plugin.dll
  - ArchestrA.Workflow.Activity.Runtime.dll
  - ArchestrA.Workflow.Codebehind.IDE.dll
  - ArchestrA.Workflow.Codebehind.InTouch.dll
  - ArchestrA.Workflow.EventHandler.dll
  - ArchestrA.Workflow.Events.dll
  - ArchestrA.Workflow.GalaxySecurity.dll
  - ArchestrA.Workflow.GalaxySecurityProvider.dll
  - Skelta.ArchestrA.Connector.Client.dll
  - Skelta.ArchestrA.Connector.Client.Proxy.dll
  - Skelta.Connector.CommonLibrary.dll
  - Skelta.Connector.Proxy.dll
  - Skelta.WinForms.Security.dll
- *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\ECSite\Bin* and *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\EnterpriseConsole\Bin* folder
  - ArchestrA.Workflow.EventHandler.dll
  - Skelta.WinForms.Security.dll
  - ArchestrA.Workflow.Activity.Runtime.dll

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Skelta.Connector.CommonLibrary.dll
- Skelta.Connector.Proxy.dll
- Skelta.ArchestrA.Connector.Client.Proxy.dll
- ArchestrA.Workflow.Codebehind.InTouch.dll
- ArchestrA.Workflow.Codebehind.IDE.dll

- *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\IntegrationScripts* folder
  - All the files and folders will get deleted.

- *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Locales\en-US* folder
  - AAWaitForEvent.txt
  - AAWFAssoication.txt
  - InTouch.txt

- *Web\BPMUITemplates\Default\ ActivityImages*, *Web\BPMUITemplates\Default\ ArchestrA*, *Web\ BPMUITemplates\Default\ Help*, *Web\BPMUITemplates\Default\ InTouch*, *WorkflowElements\ Default* folders
  - Some AVEVA Work Tasks Extension for System Platform related files will get deleted.

Upgrade Extension for System Platform

You can upgrade to **AVEVA Work Tasks 2023 SP1 Extension for System Platform** from the following version:

- AVEVA Work Tasks 2023 Extension for System Platform

Prerequisites

- In the AVEVA Work Tasks server, upgrade **AVEVA Work Tasks 2023** to **AVEVA Work Tasks 2023 SP1** version.

Steps to Upgrade

If you are installing AVEVA Work Tasks 2023 SP1 in the Enterprise Server mode, install the AVEVA Work Tasks 2023 SP1 Extension for System Platform on all the servers.

1. Stop all the AVEVA Work Tasks services in the AVEVA Work Tasks server.
2. Run the **AVEVA Work Tasks 2023 SP1 Extension for System Platform.exe** installer.

   The installer splash screen, and then the **AVEVA Work Tasks 2023 SP1 Extension for System Platform** screen appears.
3. Click **Next**.

   The License Agreement screen appears.
4. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.

   The Customer Information screen appears.
5. Enter the **User Name** and **Organization** name.
6. Under **Install this application for**, select one of the following:
   - **Anyone who uses this computer (all users):** If the application can be accessed by all the users using this

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

computer.

- **Only for me (localadm):** If the application can be accessed by only the local administrator of the computer.

7. Click **Next**.

   The Click Install to begin the installation screen appears.

8. Click **Install** to begin the installation.

   After the installation is complete, the InstallShield Wizard Completed screen appears.

9. Click **Finish** to complete the installation.
10. Restart all the AVEVA Work Tasks services.

Extension for OPC UA

The OPC Unified Architecture (OPC UA) is a set of specifications that are not based on Microsoft COM. You can use the AVEVA Work Tasks Extension for OPC UA to read or write tag values from the OPC UA server.

Capabilities:

- Connect AVEVA Work Tasks to one or more existing OPC UA servers on the network.
- Read or write tag values from the OPC UA server which is connected to the workflow server.

The following are the **prerequisites** to access the application server:

- To access AVEVA Work Tasks server, Windows Authentication User information is required.
- The user must be a part of Roles Administrators, SkeltaAdmins, or the SkeltaRemoteUsers.
- The user should have administrative rights to the workflow repository.

Install Extension for OPC UA

Install the **AVEVA Work Tasks Extension for OPC UA** with **AVEVA Work Tasks** installation on all AVEVA Work Tasks client and servers.

Run the *AVEVA Work Tasks 2023 SP1 Extension for OPC UA.exe* installers on a 64-Bit computers to install Extension for OPC UA.

Note: If the Work Tasks farm is configured with **Windows** authentication, the user attempting the installation must have the **sysadmin** role on the database server.

Installation Sequence

The following image shows the installation sequence for **AVEVA Work Tasks Extension for OPC UA**:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. AVEVA Work Tasks Installation

   Install AVEVA Work Tasks on all the AVEVA Work Tasks servers.
2. AVEVA Work Tasks Configuration

   Configure Workflow in AVEVA Work Tasks. For more information about configuring workflow, see *Administrator Guide*.
3. AVEVA Work Tasks Extension for OPC UA Installation and Configuration
   a. Install AVEVA Work Tasks Extension for OPC UA on all the AVEVA Work Tasks servers.
   b. Provision a repository for OPC UA using the AVEVA Work Tasks OPC UA Repository Provisioning Tool.
   c. Verify the AVEVA Work Tasks Extension for OPC UA installation.

Install AVEVA Work Tasks

To know about installing **AVEVA Work Tasks**, see *Installing AVEVA Work Tasks* section in the *Administrator Guide*.

Install Extension for OPC UA

Complete the following steps to install the **AVEVA Work Tasks Extension for OPC UA**.

1. Stop all the AVEVA Work Tasks services.
2. Run the *AVEVA Work Tasks 2023 SP1 Extension for OPC UA.exe* on the 64-bit computer. You must run the installer with the *Run as administrator* option.

   The installer splash screen, and then the **AVEVA Work Tasks 2023 SP1** Extension for OPC UA dialog box appears.
3. Click **Next**.

   The License Agreement screen appears.
4. Read the terms, select the **I accept the terms in the license agreement** option, and then click **Next**.

   The Customer Information screen appears.
5. Enter the **User Name** and **Organization** name.
6. Click **Next**.

   The Click Install to begin the installation screen appears.
7. Click **Install** to begin the installation.

   After the installation is complete, the InstallShield Wizard Completed screen appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

8. Click **Finish** to complete the installation.

Provision Repository Using OPC UA Repository Provisioning Tool

The **AVEVA Work Tasks OPC UA Repository Provisioning Tool** is used to provision a workflow repository for OPC UA. Only the provisioned repositories can be connected to OPC UA. Workflow repository provisioning for OPC UA should be done only after installing AVEVA Work Tasks.

The **AVEVA Work Tasks OPC UA Repository Provisioning Tool** gets installed during the installation of AVEVA Work Tasks Extension for OPC UA.

To Provision a Repository for OPC UA:

1. Run the **AVEVA Work Tasks OPC UA Repository Provisioning Tool**.

   On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then click **AVEVA Work Tasks OPC UA Repository Provisioning Tool**. The **AVEVA Work Tasks** Extension for OPC UA Configuration window appears which displays all the repositories from the computer.
2. Right-click on any cell column of the particular repository, and then select the **Provision**.

   Alternatively, click the row arrow corresponding to the particular repository. When the repository row gets selected, click **Provision**.

   The following warning message appears:

   Provisioning will modify the repository database by adding new tables and adding or updating of records in the existing tables. It is highly recommended to take a backup of the repository database before provisioning, as there is no option to revert the provisioned changes. To revert back the repository database to its previous state, use the Database Restore option. Do you want to Continue?
3. Select **Yes** to provision the workflow repository.

   Repository is successfully provisioned.
4. Click **Close**.

Note: To unprovision an already provisioned repository, right-click on any cell column of the particular repository, and then select **Unprovision**.

Upgrade Extension for OPC UA

You can upgrade to **AVEVA Work Tasks 2023 SP1 Extension for OPC UA** from the following version:

- AVEVA Work Tasks 2023 Extension for OPC UA

Prerequisites

- In the AVEVA Work Tasks server, upgrade **AVEVA Work Tasks 2023** to **AVEVA Work Tasks 2023 SP1** version.

Steps to Upgrade

1. Stop all the AVEVA Work Tasks services in the AVEVA Work Tasks server.
2. Install **AVEVA Work Tasks 2023 SP1 Extension for OPC UA** in the server.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Restart all the AVEVA Work Tasks services.

Work with Connectors for System Platform

The System Platform is an industrial software platform for Supervisory HMI, SCADA, and Production and Performance Management. System Platform contains an integrated set of services and an extensible data model to manage plant control and information management systems. System Platform supports both the supervisory control layer and the manufacturing execution system (MES) layer, presenting them as a single information source. Modular applications sit on top of the System Platform. Wonderware and other third-party integrators are delivering a growing inventory of application components that use System Platform services.

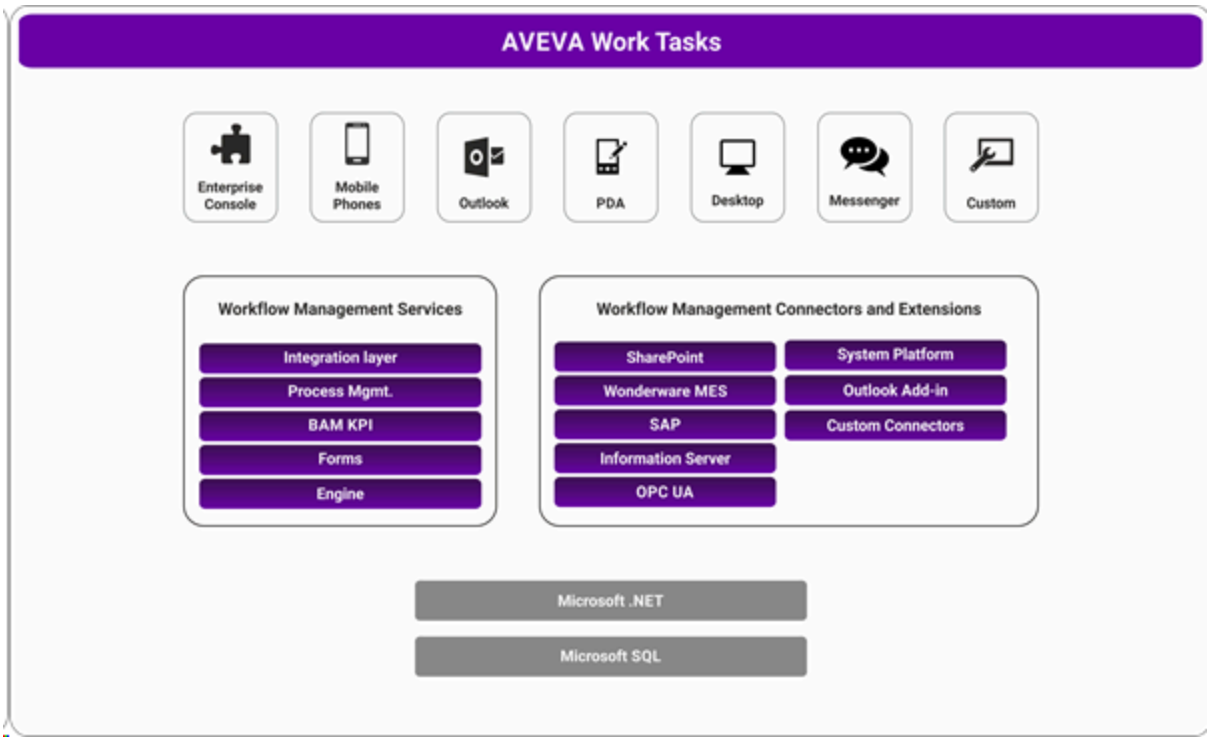
The AVEVA Work Tasks Connector for System Platform provides the following features:

- Configurable read/write and wait-for-event workflow activities for Application Server integration
- Access to Enterprise Console from Information Server
- Extends standard objects in System Platform IDE with a workflow tab to configure workflow events
- Enable Historian events to invoke workflows
- ArchestrA graphic templates for task delivery and response
- View and respond to workflow tasks in InTouch for System Platform

Overview

AVEVA Work Tasks provides a real time, end-to-end institutionalization of processes across hierarchies in a manufacturing organization. Productivity, both on the plant floor and the top floor is a function of robust and complex processes. The ability to follow such processes, especially in the backdrop of disruptive events on the plant floor is critical. End-to-end process management and procedural data management across hierarchies has always been an area of concern. AVEVA Work Tasks addresses this concern by enabling comprehensive, end-to-end process institutionalization across a manufacturing enterprise.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Features of AVEVA Work Tasks:

- Workflow Capability: Collaboration and workflow can be implemented at all levels, within and across levels with AVEVA Work Tasks Workflow.
- Collaboration: Drive higher levels of people-centric and system-centric collaboration.
- Inter-operability: Drive higher levels of inter-operability beyond technical integration.
- Procedural Key Performance Indicators (KPI): Provide procedural KPIs for expanded operations.
- Procedural data management: Centralized management of processes for enterprises.
- Connectors: Availability of out-of-the-box productized connectors.
- Holistic business processes: Workflows should be able to span across levels. An event occurring at Level 1 (controls) requires interaction at Level 2 (supervisory) and spawns actions at Level 3 (advanced applications).
- BPM + Composite Applications: BPM helps organizations to improve business processes. Composite applications are built by combining multiple existing functions into a new application, thus addressing extensibility.
- Composite Workflow Execution Framework: Move beyond technical integration to provide framework to orchestrate manufacturing/infrastructure services and plant-level business processes across disparate domains (people, processes and systems).
- Embed workflow capability: Embed workflow capability into applications at all levels that translates real-time business events into actionable information.

Workflow Features Inside System Platform IDE and Runtime

This topic provides information on how to use the Workflow Features inside System Platform IDE and during

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

runtime.

Overview of a Gateway Object

A Workflow Gateway object allows you to connect to multiple AVEVA Work Tasks servers from within a Galaxy. It also facilitates connection back from the server.

Multiple Gateway Objects can be present in each Galaxy. Each of these Gateway objects can connect to either the same or different workflow repositories. The mapping between workflow repositories and Galaxies can be many-to-many.

The Gateway Object handles connectivity from:

- The System Platform IDE and ArchestrA runtime node to the server
- The server to the ArchestrA runtime node

Connectivity from the System Platform to the server is called **Outbound Connection.** Connectivity from the server to the System Platform is called **Inbound Connection.**

The Gateway Object provides connectivity to the server during the configuration and deployment of the Gateway Object. The connectivity is provided to the server to:

- Fetch workflow repository information
- Provision the workflow repository
- Synchronize user information

During deployment, the Gateway Object connects to the server to provide information of the runtime node. The runtime node contains the **WCF** client service needed to perform read/write on attributes.

The communication between AVEVA Work Tasks and System Platform is handled using the Windows communication service. These connections are secured using Windows authentication. To do a call on the WCF hosted in the server,

- The Client (System Platform - Runtime node or System Platform IDE Node) requires valid Windows authentication user information of the server.
- Valid Windows authentication user information to the WCF hosted in the runtime node is also required.

  The WCF hosted in runtime node is primarily used to read or write attribute values.There is a second level of authorization to access the workflow repository available in the server under a farm.
- Only a user who has administrative rights in the workflow repository can make a connection to the WCF (Windows Communication Foundation) service and access workflow repository-specific information.

Initial Gateway Setup

Complete the following steps to setup the Initial Gateway:

Notes: The screen captures shown here are from System Platform 2023 IDE.

1. Open **System Platform IDE**.
2. On the **Galaxy** menu, click **Configure**, and then click **Security**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Security** dialog box appears.

3. On the **Authentication Mode** tab, change the authentication from **None** to **Galaxy**.

Note: Select **OS User based** or **OS Group based** to use system users or roles respectively. In such cases, configure users or roles under **Users** or **Roles** tab respectively.

4. On the **Users** tab, click the **Add** button, enter a username, and then click **Change Password**.

   The user is added, and the **Change Password** dialog box appears.

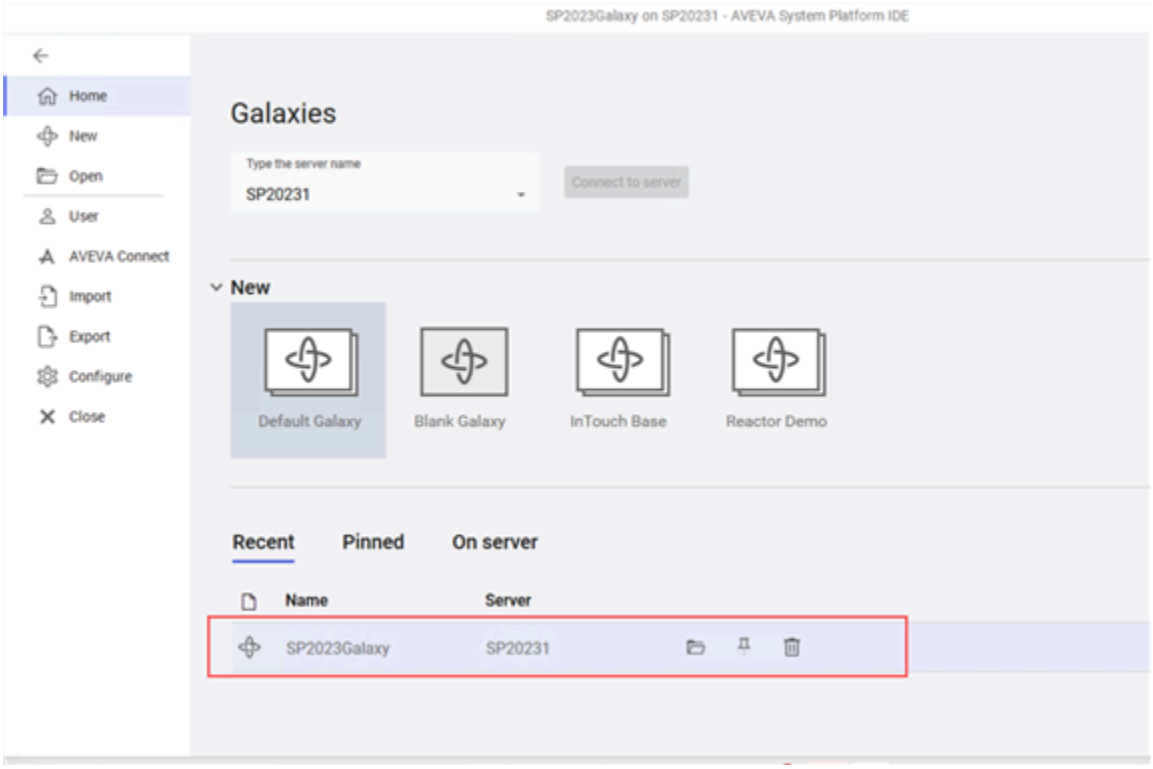
5. In the **New Password** and **Confirm New Password** boxes enter a password for the user, and then click **OK**.

   The password is saved for the user.

6. On the **Roles** tab, Select **Administrator**, and then click **OK**.
7. In the **System Platform IDE**, on the **Galaxy** menu, click **Change User**.

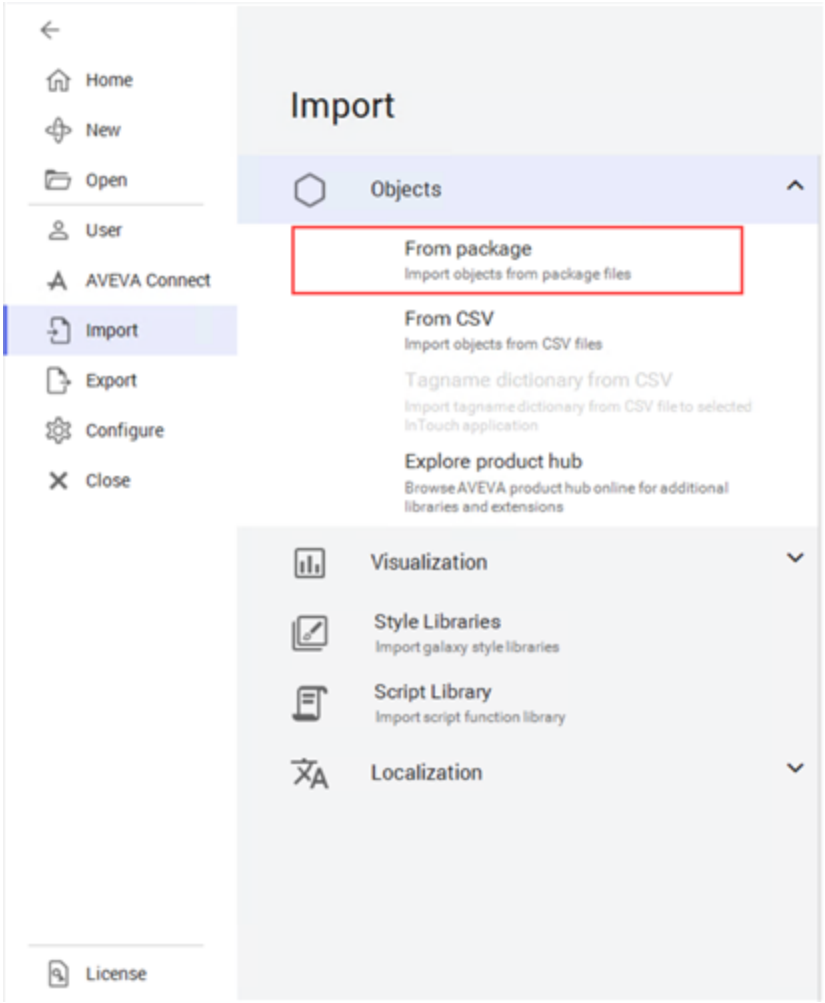
   The **Change User** dialog box appears to log on with new user credentials

8. To connect to a Galaxy, select the desired <Galaxy name> from the list. For instance, in this example, we are connecting to **SP2023Galaxy**.



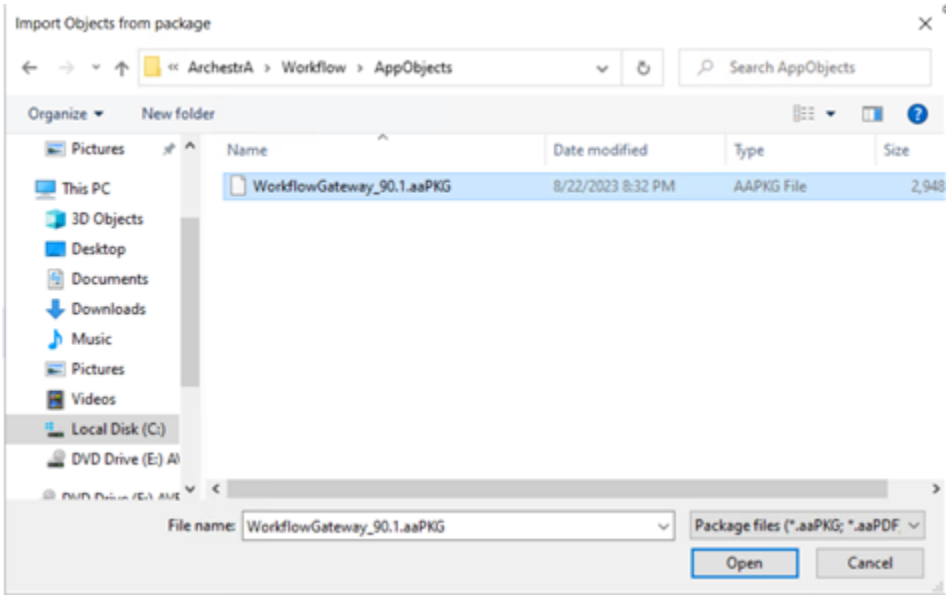
9. To import the objects, point to **Import** on the **Galaxy** menu, select **Objects** and then select **From package**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The Import Objects from package dialog box appears.

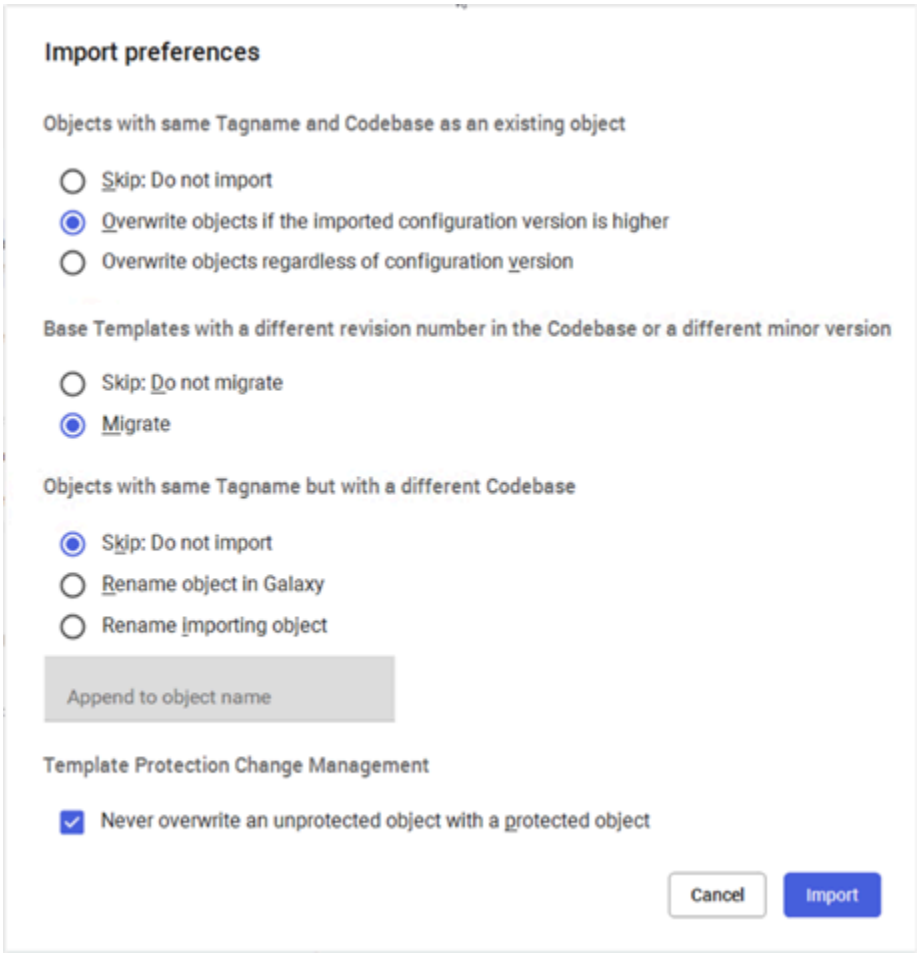
10. Open the *AppObjects* folder from the **..\<Archestra Installation Path>\Archestra\Workflow** path.



11. Select the **WorkflowGateway** file (for example, **WorkflowGateway_90.1.aaPKG**), and then click **Open**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

12. In the Import Preferences dialog box, select the appropriate options.



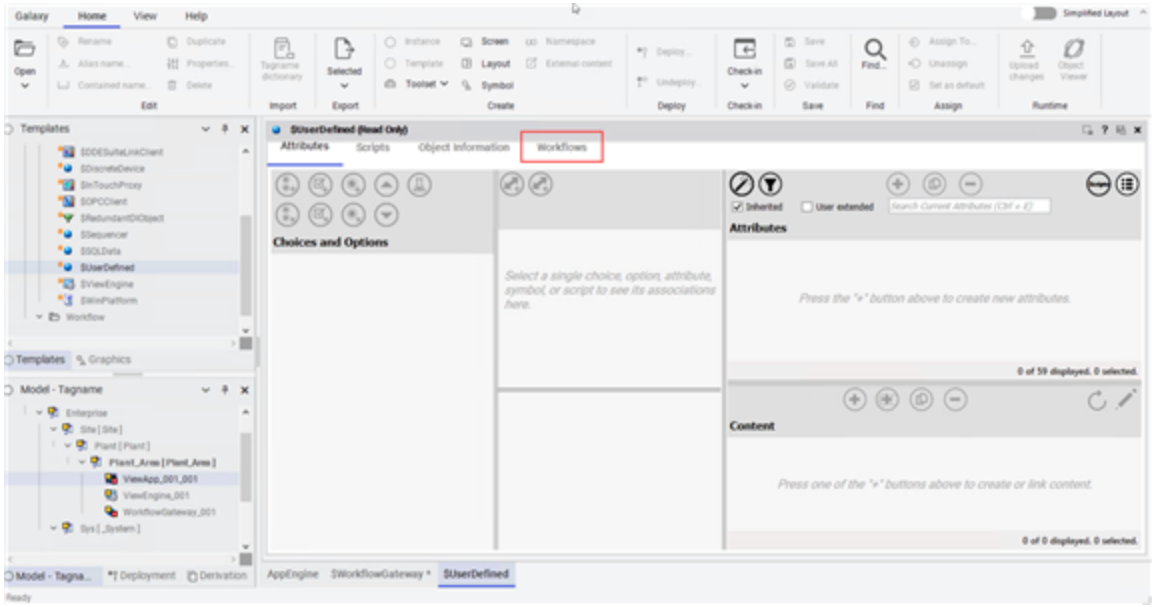
13. Click **OK**.

    The Import Completed dialog box appears after the import completes.

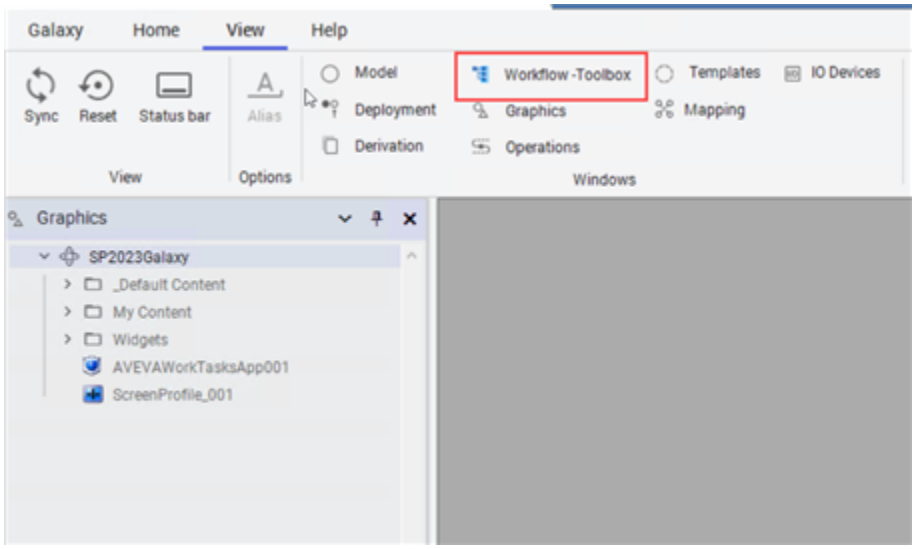
14. After the import is complete, close and re-start the IDE to ensure that the IDE extension appears.

    To confirm that the IDE extension is enabled, verify that the **Workflows** tab for an object is visible. You will also see a **Workflow-Toolbox** tab in the **Templates** section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



If the **Workflow-Toolbox** is not visible, navigate to **View** menu and select **Workflow-Toolbox**.

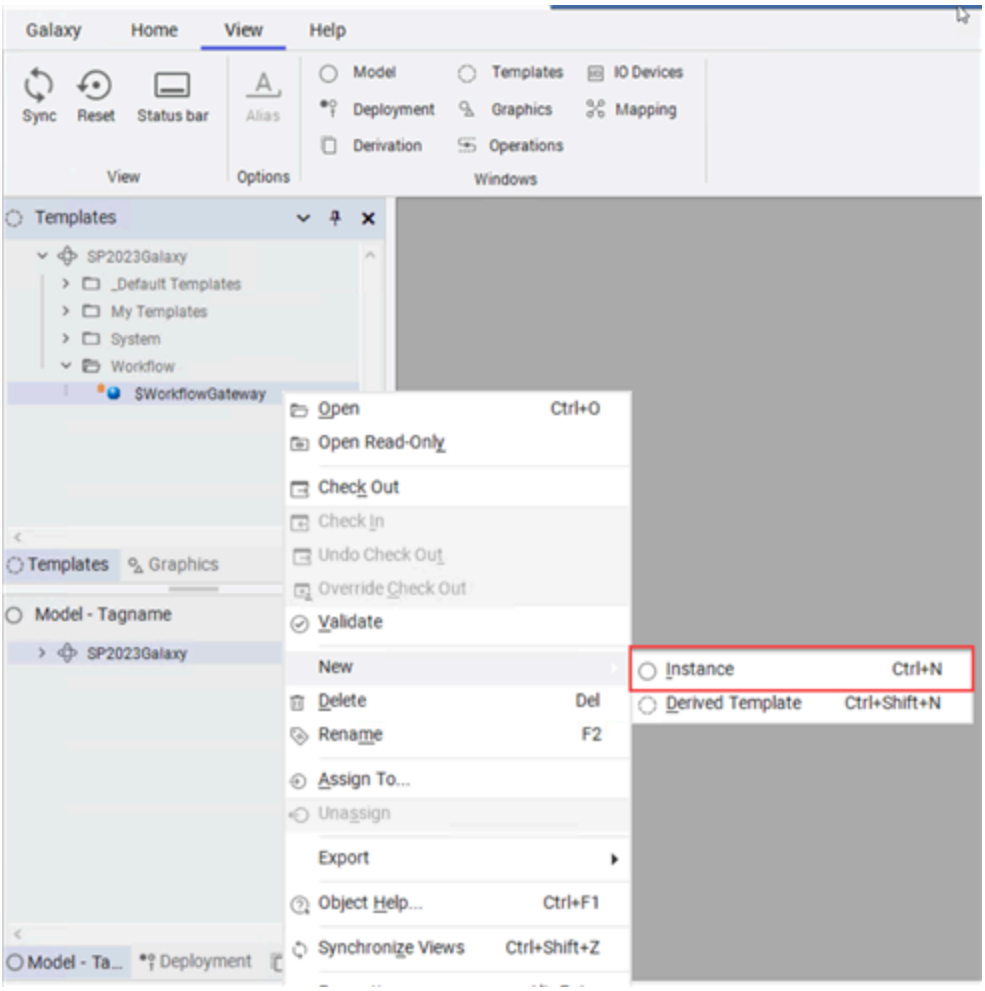


If the **Workflow-Toolbox** or the **Workflows** tab does not appear, copy the DLLs manually from ..\<Archestra Installation Path>\Archestra\Workflow\IDE Extension to **..\<Archestra Installation Path>\Archestra\Framework\BIN\IDE Extensions**.

After you copy the DLLs, restart the System Platform IDE to ensure that the **Workflow-Toolbox** and the **Workflows** tabs appear in the IDE.

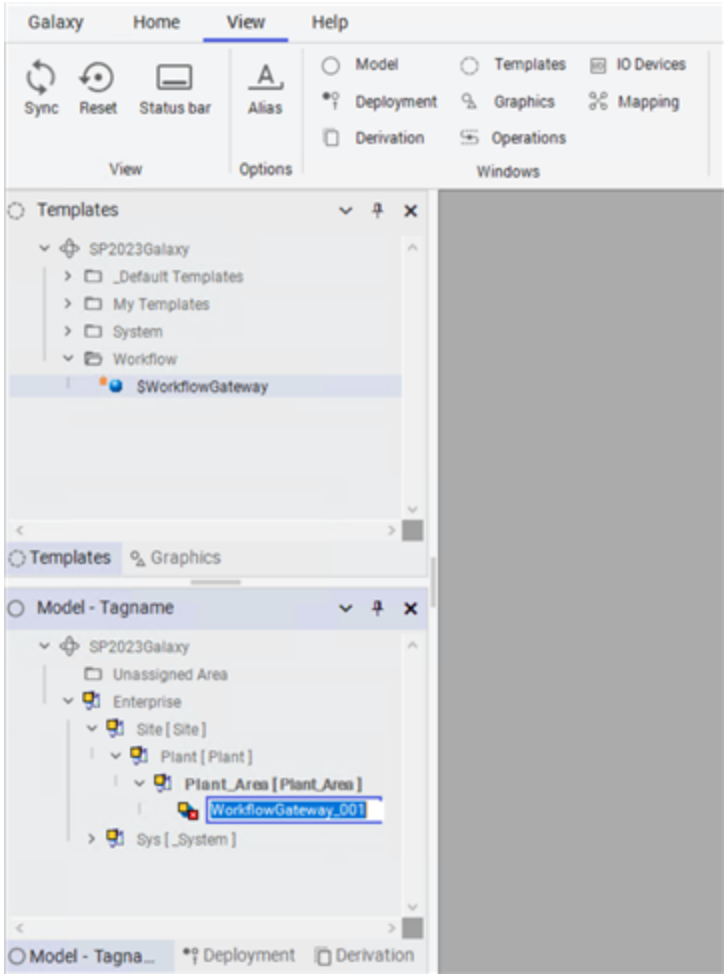
15. In the **Galaxy** menu, click **Import** to import the script library (ArchestrA.WorkflowScripts.dll) from the following path:
    - 64-bit machine: C:\Program Files (x86)\ArchestrA\Workflow\Script Library
16. Right-click **$WorkflowGateway**, point to **New**, and then select **Instance** to create an instance of a workflow gateway.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The *WorkflowGateway_001* instance is created under **Unassigned Area** in the **Model Tagname** section as shown in the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



If the WorkflowGateway object is created in the **Unassigned Area**, the user must drag it to the desired container.

AIM Authentication for accessing Enterprise Console in System Platform IDE

You can use AVEVA Identity Manager (AIM) authentication for AVEVA Work Tasks Enterprise Console (EC) website in **System Platform IDE**. To enable AIM authentication, ensure that in **Central Configuration** when the **Resource Provider** is set to **Active Directory**, the **Use Token Service** is set to the appropriate server.

When you access the EC website for the first time, you will be navigated to the AIM authentication page. After authentication, you can access all the EC pages without any prompt for AIM authentication again for each page.

To configure System Platform IDE Security

1. In the **System Platform IDE**, on the **Galaxy** menu, click **Configure**, and then click **Security**.

   The **Security** dialog box appears,
2. On the **Authentication Mode** tab, select **OS User based** or **OS Group based**.
3. On the **Users** tab, under **Authorized Users Available**, add users, and then under **Associated Roles for Administrator**, map **Administrator** role.
4. Click **OK**.

   The **System Platform IDE** will close and reopen.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Sometimes, you may face issues related to Anti-forgery. These issues might be because of cache. To resolve this issue, follow the below steps:

1. Close System Platform IDE.
2. Open browser window.
3. Clear browser cache.
4. Close all browser windows.
5. Open System Platform IDE and try accessing EC pages.

Configure Gateway

After you set up the Initial Gateway, you must configure it. This includes adding credentials for access to Workflow Server which is the first level of authentication.

Prerequisites:

- Before provisioning a repository, make sure you configure Galaxy security to any of the valid choices. Do not change the Galaxy security after provisioning.
- After you configure the Gateway Object and use it for creating events, do not change its name.

To configure the Gateway

1. Double-click the instance that you created while setting up the initial gateway.

   For example, double-click *WorkflowGateway_001* in the Deployment section.
2. In the Gateway Configuration tab that opens, enter the values for the following fields:
   - **Unique Environment Name**: The server is capable of communicating with multiple Galaxies in a single process. In a multi-Galaxy environment, you cannot ensure the uniqueness of the Galaxy name. An **Unique Environment Name** is used as work around for this.

     In an environment with a single Galaxy, or multiple Galaxies with unique names, the Unique Environment Name can be the same as the Galaxy name.

     In an environment with multiple Galaxies where different Galaxy environments share the same name, each separate Galaxy environment needs to be provided with an unique name while connecting to the server. An unique name is required to avoid naming conflicts.

     By default, in the **Unique Environment Name** field, the Galaxy name is populated. You can edit this once at the beginning. However, once configured and saved, the Unique Environment Name field becomes read-only.
   - **Passphrase**: The passphrase to configure Cryptography Settings and use the encryption algorithm in the product. Click **Import** to import the saved cryptography settings.

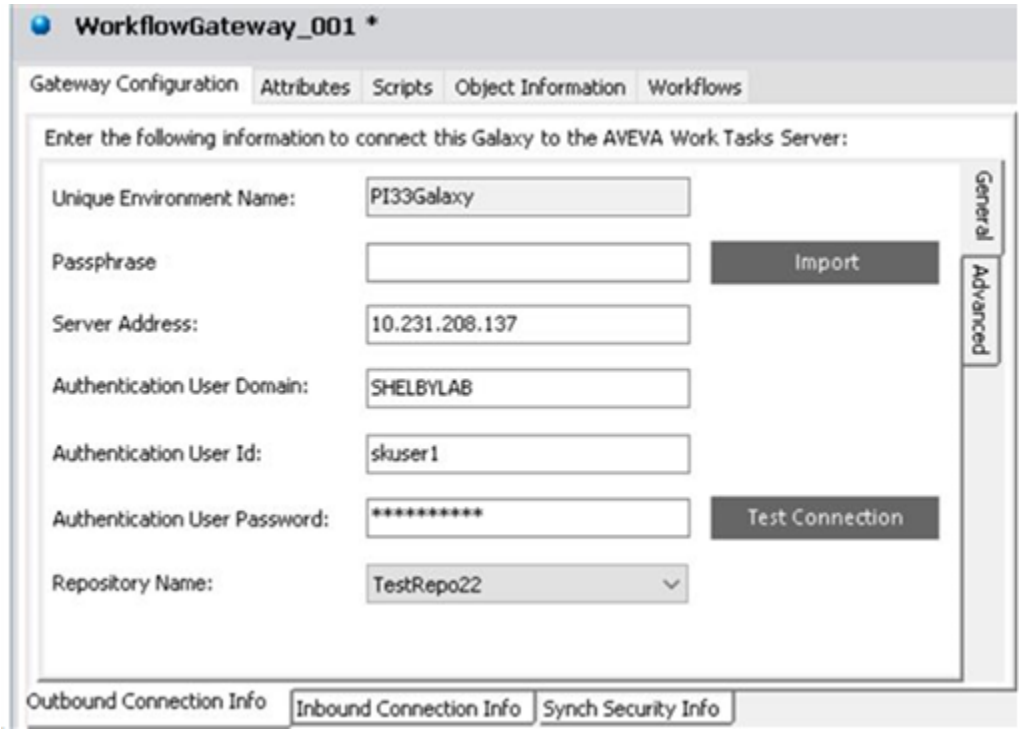
     Note:
     - When you provision the Gateway object to multiple AVEVA Work Task Servers, provide the same Passphrase settings on all the AVEVA Work Tasks Servers.
     - The passphrase value should have characters between 8 and 20, consisting of lowercase, uppercase,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

number, and special characters.

- **Server Address**: Address of the server.
- **Authentication User Domain**: User domain of the server. The domain name is case sensitive.
- **Authentication User Id**: User ID of the server.

  Authentication User Password: Password for the user name of the server.



3. Click **Test Connection** to verify the authentication.
   - If **Test Connection** is successful, the **Repository Name** field is populated with the available repositories.
   - If authentication is not successful, verify the credentials. If authentication is not successful consistently, contact the network administrator to ensure that the user credentials provided are part of the Administrators, SkeltaAdmins, or the SkeltaRemoteUsers roles, and there are no communication issues between the machines. Also, the SkeltaAdmins or SkeltaRemoteUsers roles need to be manually created.
4. In the **Repository Nam**e field, select the workflow repository to connect to. The Provision window appears.
5. Add credentials to access the workflow repository which is the second level of authentication. To authorize and provision access of and mapping to the workflow repository, perform the following steps:

   Workflow Repository Authentication Info

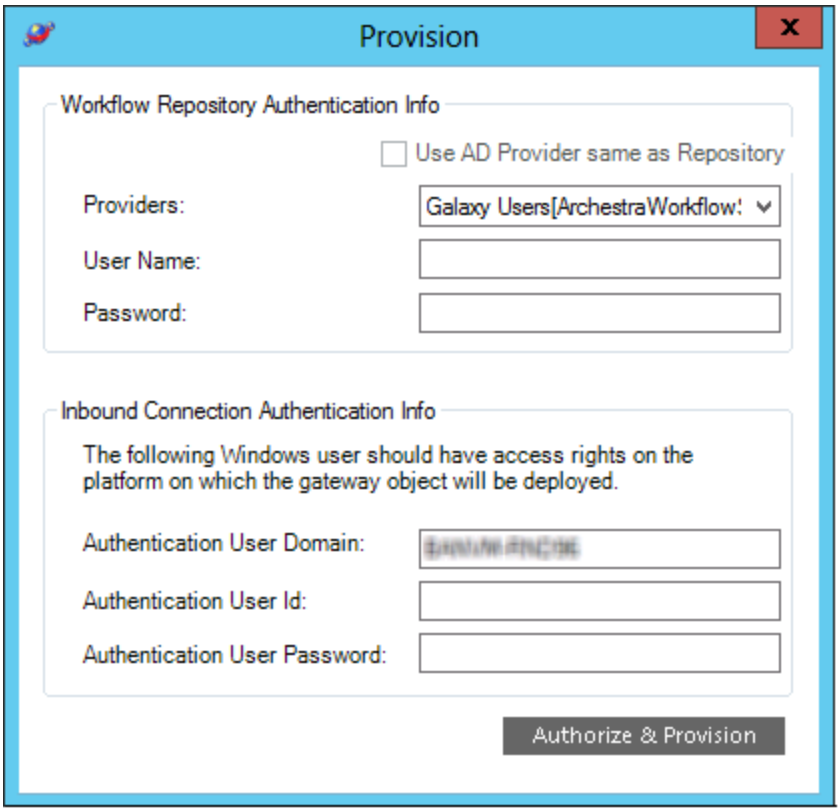
   a. In the **Providers** field, select Active Directory or AVEVA Work Tasks List based on the provider for which the workflow repository is created.

   b. Enter the user name and password details based on the user provider. For the default user name and password, contact the technical support team or refer the *Get Started Guide* in the ISO file of AVEVA Work Tasks.

      The user must have administrator rights on the workflow repository.

   c. Select **Use AD Provider same as Repository** to leverage Active Directory users instead of Galaxy users.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



6. For **Inbound Connection Authentication Info**, enter the credentials of GR Node in the **Authentication User Domain**, **Authentication User Id**, and **Authentication User Password** fields.

   Note: The **Authentication User Domain** is case sensitive and must be provided in proper case.

7. Click **Authorize & Provision**.

   The repository is now provisioned to be connected to the Galaxy. The repository and galaxy are now connected.

The **Use Active Directory Service Provision** check box appears only if all the following conditions are satisfied:

1. The Galaxy User security is Operating System user or Operating System group.
2. The repository to provision is created using the Active Directory authentication.
3. The Galaxy user and repository user domain are same.

On provisioning, the users in Galaxy are available to the workflow after creating a user provider for the Galaxy.

Galaxy users can log on to the **Enterprise Console** as *ArchestrA Users* after provisioning and deploying the gateway object to a workflow repository. If the users log on before provisioning and deploying the gateway object, the following message appears:

Deploy the workflow gateway object to login. Users can login only after the gateway object associated to the repository <repository name> is deployed from the System Platform IDE.

After deploying the gateway object, if the security mode in the galaxy has been changed, then the workflow gateway object needs to redeployed.

The following activities take place during provisioning:

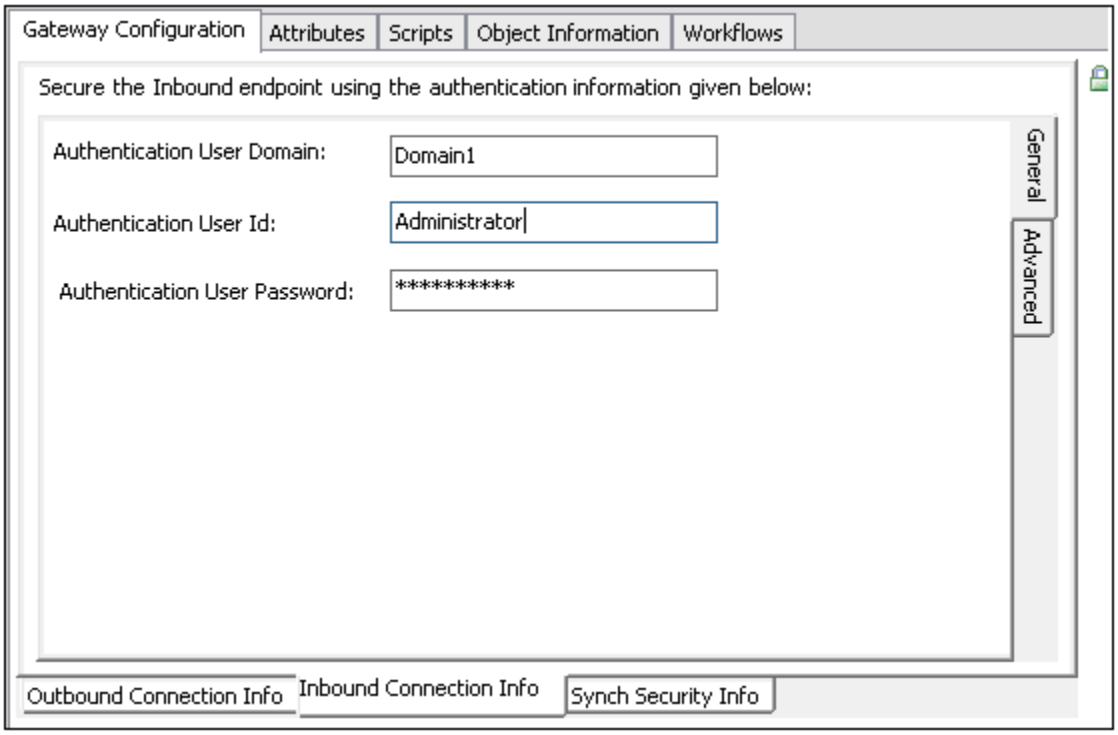
- The logged in user is added as administrator in the workflow repository based on the user authentication

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

provided during provisioning. This is done only if the user is an administrator in the workflow repository.

- Galaxy information is stored in AVEVA Work Tasks using tables and lists. The Galaxy users are synchronized to the workflow repository.
- Enabling a new user provider to map the Galaxy users to the repository allows the Galaxy users to be a part of the AVEVA Work Tasks workflow activities.
- The Galaxy users can also access the AVEVA Work Tasks functionalities, such as Inbox, Forms, Business Activity Monitoring (BAM), Process Designer.

The details that you enter in the **Inbound Connection Info** tab is the same as the **General** inbound tab as shown in the following image:



Unprovisioning a Provisioned Workflow Repository

You can unprovision a provisioned workflow repository. Unprovisioning is an option of the workflow gateway object to roll back the changes done to the workflow repository. While unprovisioning, the workflow and associations created during provisioning the workflow repository, are not deleted.

While moving the workflow repository database from one environment to another environment, you must unprovision the workflow repository.

To unprovision a provisioned workflow repository, complete the following steps:

1. Open the derived template or instance of the workflow gateway object.
2. Click **Test Connection** to validate the connection.
3. Select the provisioned repository from list of available repositories in the **Repository Name** field.

   The Provision dialog box appears.
4. Click **Un Provision**.

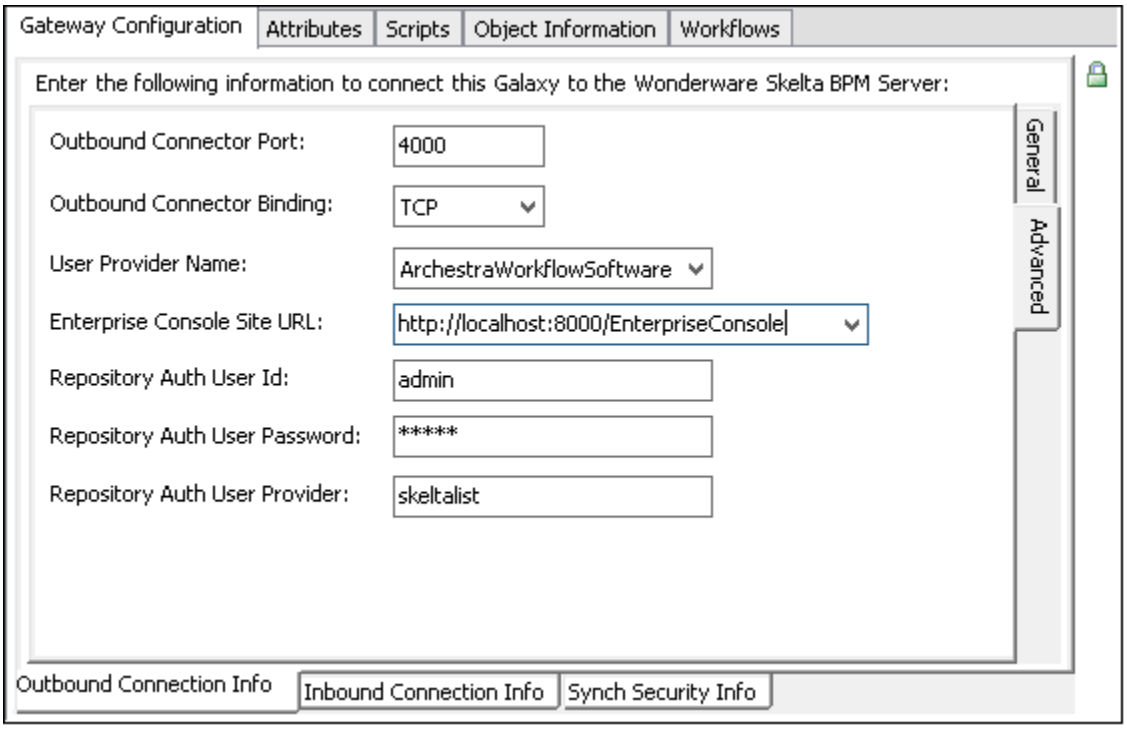
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using the Advanced Tabs

Advanced tabs are available for Inbound and Outbound Connection settings, for more control on the Gateway configuration.

Outbound Connection Info: Advanced Options

The following image shows the fields of the **Outbound Connection Info** tab.



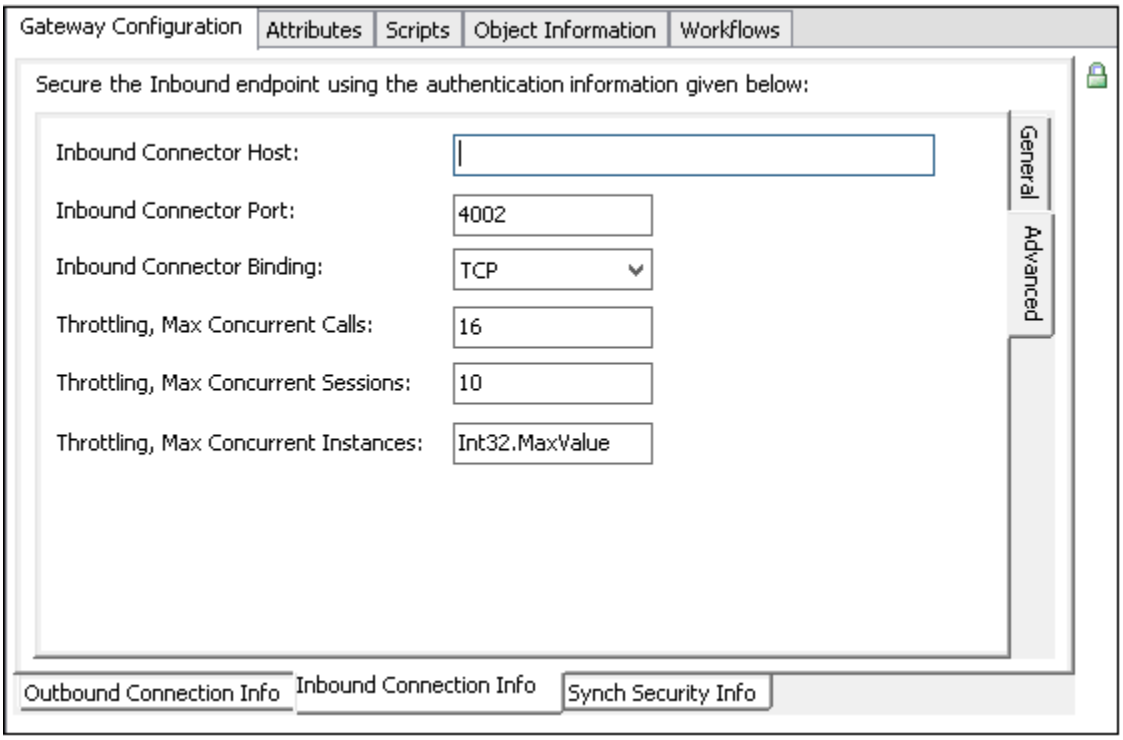
The **Outbound Connection Info** tab contains the following fields:

- **Outbound Connector Port**: This field can be edited after you provision the workflow repository for the Galaxy.
- **Outbound Connector Binding**: Binding protocol for the connection, preferably used when Workflow Server is within the same network. Choose from the following options:
  - **TCP**: Default binding protocol for the connection, preferably used when Workflow Server is within the same network.
  - **HTTPS:** Secure connectivity over the Internet.
- **Operation Timeout**: Time period after which the request made to the Workflow server does end if not connected.
- **Connection Life Time**: The duration for which the of the connection between the Galaxy and the Workflow Server lasts.
- **User Provider Name**: Repository user provider, either Active Directory or Skelta List.
- **Theme**: Theme of the repository.
- **Enterprise Console Site URL**: The URL used to access the workflow repository from the Workflow Server.
- **Repository Auth User Id**: Captured during and displayed during provisioning.
- **Repository Auth User Password**: Captured during provisioning.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Inbound Connection Info: Advanced Options

The following image shows the fields of the **Inbound Connection Info** tab.



The **Inbound Connection Info** tab contains the following fields:

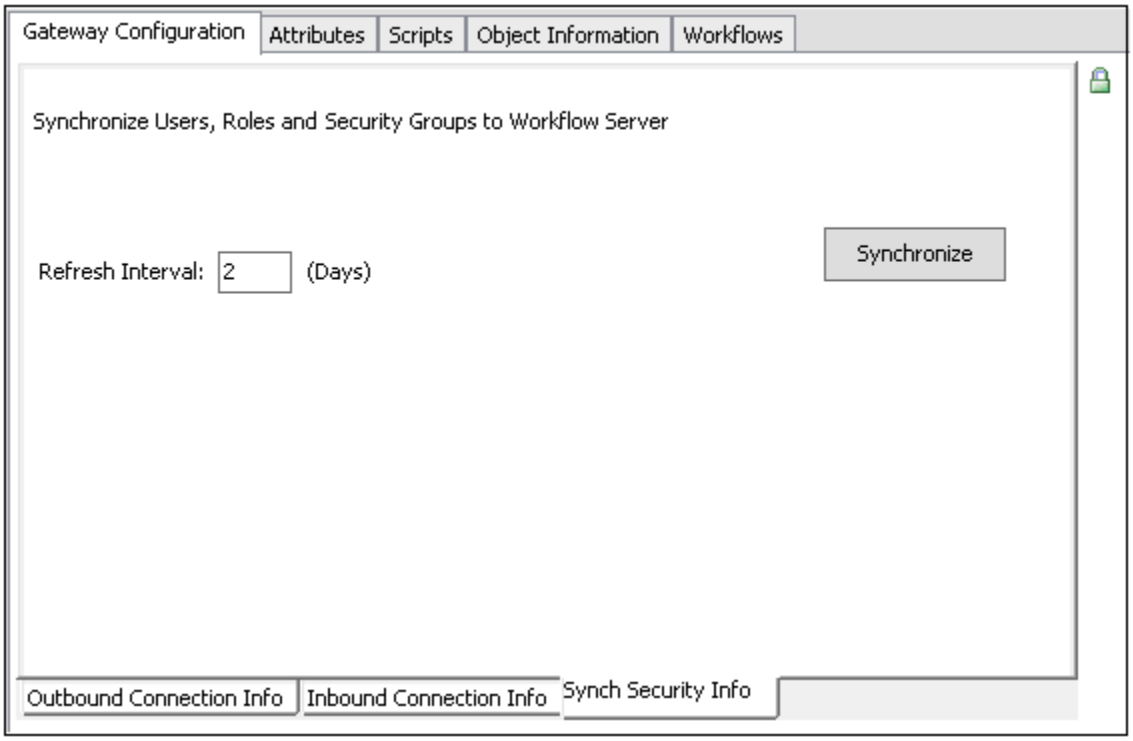
- **Inbound Connector Host**: Default runtime node where the Gateway object is deployed.
- **Inbound Connector Port**: Connection Port by AVEVA Work Tasks to fetch any data from Galaxy.
- **Inbound Connector Binding**: Binding protocol for the connection, preferably used when Workflow Server is within the same network. Choose from the following options:
  - **TCP**: Default binding protocol for the connection, preferably used when Workflow Server is within the same network.
  - **HTTPS**. Secure connectivity over the Internet.
- **Operation Timeout**: Time period after which the request made from the Workflow server to the Galaxy will end, if not connected.
- **Connection Life Time**: Time period that the connection will last between the Workflow Server and the Galaxy.
- **Throttling, Max Concurrent Calls**: Connector service properties, the maximum number of messages that can actively be processed. Each client channel can have one pending message that does not count against this total until the service begins to process it. Increase this value if you want your service to process a larger message load.
- **Throttling, Max Concurrent Instances**: Connector service properties, the maximum number of InstanceContext objects in a service that can be executed in a single instance.
- **Throttling, Max Concurrent Sessions**: The maximum number of sessions that a service can accept at one time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Sync Security Info: Advanced Options

Use this tab to synchronize users, roles, and security groups to the Workflow Server. Set the refresh interval for automatic synchronization in a specific time-frame. To synchronize, click **Synchronize**.

After changing the Advanced options, you save the object. You can now access the Enterprise Console functionality from the Workflow Toolbox.



Note:
- In a scenario where a derived template of workflow gateway object is provisioned with a workflow repository, one instance of the Gateway Object is deployed in the GR Platform and another instance deployed in the Boot Platform. If the inbound connector host name is changed the ArchestrA Read/Write activity will not work.

- The user, which is set as **Change Network Account** needs to be mapped to the Administrators group in order to avoid SkeltaConnectorClientArchestrA service installation/uninstallation during deploy/undeploy of Workflow Gateway Object.

- To set up the zoom level, refer to Setting Zoom for the Web Browser Control in System Platform and InTouch.

- To lookup/search the user from Galaxy User's provider, search with **Galaxy User Name** search filter, instead of **Galaxy User Full Name** search filter.

- When the user is searched based on the **All Providers** search filter, the user name from different providers is listed in a single line separated by comma, and the user from all providers may not be assigned the work item. Hence, it is good to use separate providers to search the user and assign the work item. .

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Define Workflow Events in AppObject

The Gateway object should be configured before associating an event for the AppObject. In the example below, you will associate an event to the *Drum_Conveyor* object.

The *Drum_Conveyor* object has an attribute named *Cycle*, which changes when the drum is moved in the conveyor. This value varies from one to six, based on the different stages covered by the drum.

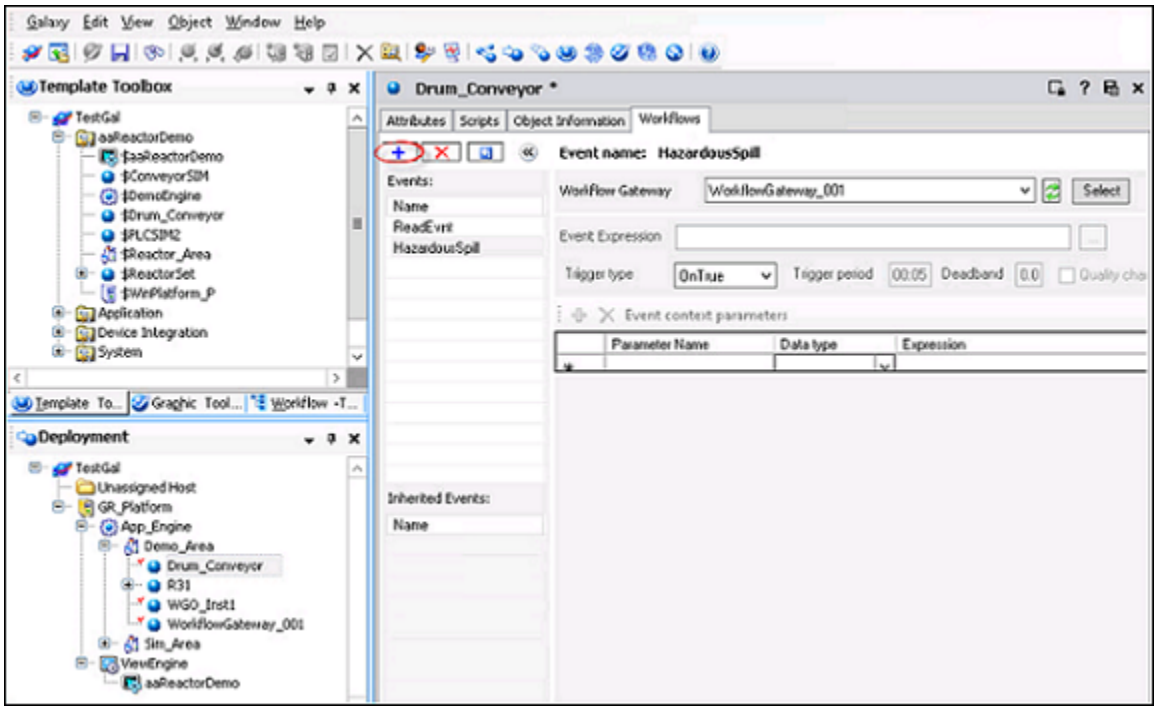
For example, the Starting of the Drum Movement in conveyor is marked as 1 and the Cycle Attribute and Falling of the Drum due to any issue in the conveyor is marked as 6. You will define an event to trigger a workflow when the attribute value changes to '6'.

To define workflow events in AppObject

1. Open the Drum_Conveyor object and select the **Workflows** tab.
2. Click the **Add Event** button to add a new event.

   For example, *HazardousSpill*.

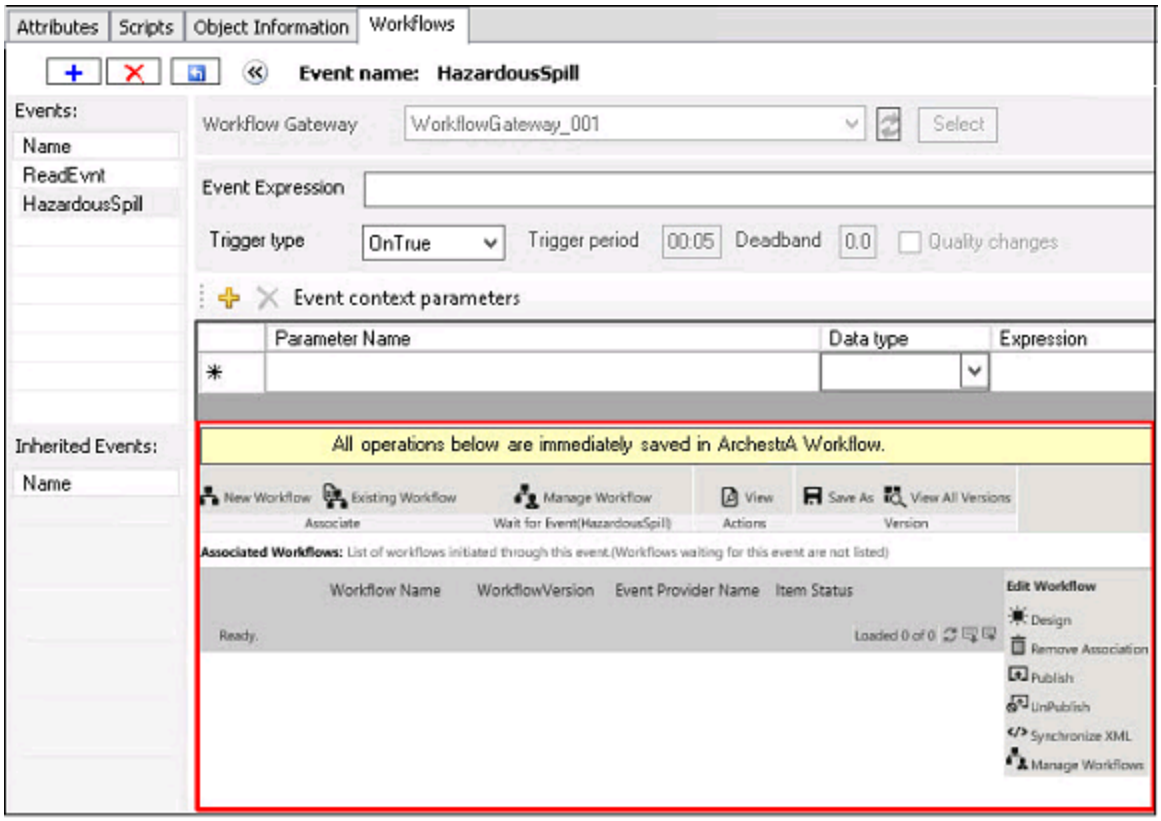
   The event gets added to the Events list.



3. Select a **Workflow Gateway** instance that contains the connection information to AVEVA Work Tasks, and then click **Select**.
4. Select the event from the **Events** list.

   The event association window appears on the lower right of the screen as shown in the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The following options are available in the event association window:

- **New Workflow**: Use this option to create a new workflow to be associated with the event.
- **Existing Workflow**: Use this option to associate an existing workflow with the event.
- **Manage Workflow**: Use this option to open the workflow list for managing the workflow.
- **View**: Use this option to view the selected workflow in a read only mode in the Process Designer. This eliminates the need to unpublish a published workflow to view it.
- **Save As**: When a workflow is first saved, it is saved with version number 1.0. The Save As option enables you to save the workflow with a different version.

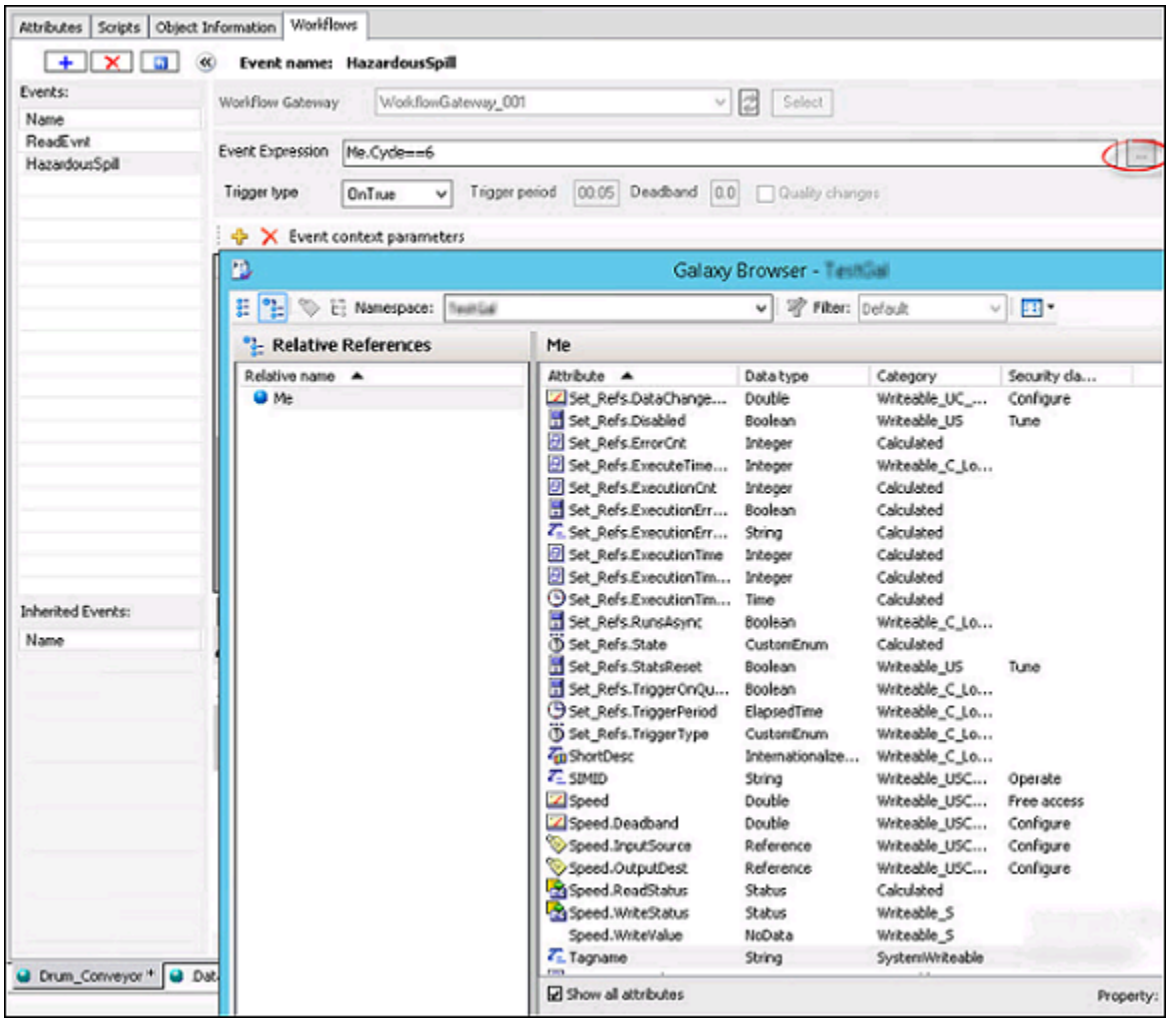
  The versioning scheme for workflows is as follows:
- **Next Version**: This option adds 1 to the lowest level of the last saved workflow version. If the last saved version is 1.1.1, then selecting this option saves the workflow as 1.1.2.
- **Next Minor Version**: This denotes a minor version for the last saved version. If the last saved version is 1, the next minor version would be 1.1. If the last saved version is 1.1, the next minor version would be 1.1.1.
- **Next Major Version**: The next major version increments the workflow version by 1. If the last saved version is 1.1.1, selecting this option assigns 2 as the version number.
- **Specific Version**: Select this option to view a list of the next version numbers under the above three options.
  The version number in Workflows list is updated with the last saved version number (if all the versions are in draft) and the version with which it is published if the workflow is published
- **View All Versions**: This option enables you to view all the versions of the selected workflow.

5. Set the **Event Expression**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For example, set the **Event Expression** as `Me.Cycle == 6`.

6. Set the **Event context parameters** that you want to pass to the workflow. Click to add the parameter.



7. Set the event contextual parameters to be passed to the workflow when the event is fired.

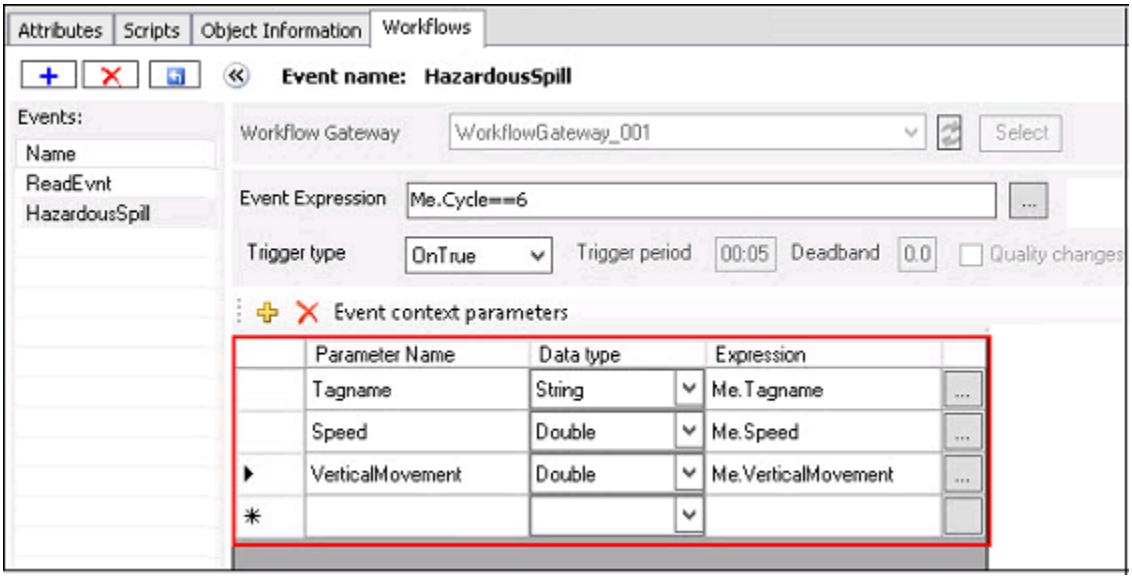
   On True is the default **Trigger type**. The AVEVA Work Tasks Event conditions can be configured using the following trigger types:

   - **On True (Default)**: The workflow gets triggered when the condition specified in the Event Expression is True.
     For example, the Trigger Type is selected as On True and the event expression is set as Me.Cycle==6, the workflow gets triggered only when the event expression is True.
   - **On False**: The workflow gets triggered when the condition specified in the Event Expression is False.
     For example, the Trigger Type is selected as On False and the event expression is set as Me.Cycle<6, the workflow gets triggered only when the event expression is False.
   - **Periodic**: The Time Period field is enabled when the trigger type selected is Periodic. The workflow gets triggered at specific intervals as specified in the Time Period field. The default value for Time Period is 5 minutes. This trigger does not require an expression.
     For example, the Trigger Type is selected as Periodic and Time Period is set as 1:00, the workflow gets triggered periodically after every hour.
   - **Data Change**: The Deadband field is enabled when the Trigger Type is selected as Data Change. The

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Deadband prevents the trigger to be raised again and again, if the event expression value moves around a specific target.
For example, the Trigger Type is selected as Data Change and the event expression is set as Me.Cycle==10. The Deadband value is set as 2. The trigger is raised when the event expression's value reaches 10. The trigger would be raised again only when the event expression's value reaches below 8.

The following image shows the Event context parameters set for the example scenario:



Note: You can use relative referencing support at the Event Context Parameter level and for the attribute read or write at the AttributeReadWrite activity level. You can use relative references like MyContainer, MyPlatform, MyEngine, MyArea, and Me.

Create New Workflow

Create a new workflow using the **Create Workflow** option.

1. Enter the **Title** and **Description** for the workflow, and then click **Save & Continue**.

   The Process Designer window appears.

   The event contextual parameters are passed as the XML variable. You can view the event contextual parameters in the XmlVariables property of Start Activity.
2. Design the workflow.

   When the value of the cycle changes to six, the associated workflow will be triggered.

Associate an Existing Workflow to a Workflow Event

Associate an existing workflow to a workflow event. Only Workflows in draft status can be associated to an event.

To associate an Existing Workflow to a Workflow Event

1. In the Workflows tab, click Associate Existing to associate the existing workflow to a workflow event.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. In the Title drop-down, select the workflow you want to associate.
3. Select the version number from the drop-down list adjacent to the Title drop-down list.
4. Enter the Description.
5. Click Save & Continue.

   The associated process designer window appears.

   The ArchestrA Context XML variable is now added to the workflow, if it is not added already.

You can use the Synchronize XML option to synchronize Event Context Parameters with the ArchestrAXMLData XML variable of the selected workflow.

- Design: Select a workflow and click on Design menu to open the process designer for editing the workflow.
- Remove Association: This will remove selected workflow's association with the event. For deleting the same workflow, go to manage workflow and delete it.
- Publish: This will publish the selected workflow.
- UnPublish: This will unpublish the workflow and change the status of the workflow to draft.

The following operations take place for synchronization:

- All newly added event context parameters in System Platform IDE are added to the XML variable in the Workflow.
- All newly added nodes under AAParameters for the XML variable in Workflow are added to the Event context parameters in ArchestrA.
- If any parameter/node is deleted, it will be ignored in the synchronize operation.
- Only drafted versions of the workflows can be synchronized. If you want to synchronize a published version, go to manage workflow and unpublish it and then synchronize it.

Note: You have to manually map the parameter to the expression.

Wait-for-AAEvent Activity in a Workflow

The Wait-for-AAEvent Activity is used to wait for an ArchestrA event within a workflow. An event should be specified for the specific object in System Platform IDE.

In the following example, an event called **Spill** will occur when the object's Cycle attribute value changes to six. We will configure a process that waits for the event **Spill** to occur. After the event occurs, the context parameters of the event are captured.

1. Create an Event Expression *Me.Cycle==6*.
2. Pass relevant parameters.
3. Click on the **Manage Workflow** menu.

   The Workflow list appears.
4. You can either create a new workflow process which will wait for a Spill event or select a workflow which should wait for the Spill event.
5. Create a new workflow.

   The **Process Designer** window appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Select ArchestrA from the activities drop-down menu.
7. Drag **Wait-for-AAEvent** activity to the designer pane.

   The **Wait-for-AAEvent Set Event Filters** screen appears.
8. Enter the required details.

   Click Save & Close to save the property.

   Note: The Workflows which are associated at the template level, wait for the respective template instances.

   Defined In Object Name in the **Wait-for-AAEvent Set Event Filters** property is used to capture the object or template name in which the specific instance is defined. The instance name is passed while starting the workflow and it can be accessed through variables.

   For example, consider a workflow associated with an alarm template. Workflow is triggered whenever the alarm status is ON. Alarm can have multiple instances like Temperature alarm and Pressure alarm.

   - If the status of both Temperature and Pressure alarm is ON, two instances of the same workflow will be triggered.
   - The workflows internally wait for the respective alarm's status to be OFF. When the Temperature alarm status is OFF, the workflow which was triggered by the Temperature alarm will proceed and it will not affect the other instance of the workflow which was triggered by the Pressure alarm.

     Object Name in the **Wait-for-AAEvent Set Event Filters** property is used to capture the name of the instance, which triggered this event.
9. Click on the icon in the **Create Output XMLVariable** property.

   The **Create Output XmlVariable** screen appears.
10. Enter a unique XML variable name.
11. Click Save & Close.

    An XML Variable with the structure of the ArchestrA context parameters is created and the same is populated in the ArchestrA Data XML variable drop-down menu.
12. Save the property to save the ArchestrA Context XMLVariables.
13. Access the XML variables property of the Start activity for verifying the newly created variable.

Note: Both the properties explained above use ArchestrA context values. When the Enterprise Console is accessed, ArchestrA context related variables needs to be set manually.

AttributeReadWrite Activity

Use the **AttributeReadWrite** activity to perform the read and write operation between AVEVA Work Tasks and System Platform IDE.

Activity Properties:

You can configure the **AttributeReadWrite** activity by specifying appropriate values for the properties in the **Activity Properties** area. Click **Activity Properties** in the toolbar or select the **Properties** option from the short-cut menu of the **AttributeReadWrite** to access the Activity Properties.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name & Description

You can use these properties to specify the name and description for the activity.

- **Name**: Use this property to specify a name for the activity.

  Property Type: Mandatory
- **Description**: Use this property to add a brief description about the activity.

  Property Type: Optional. Set this property only if required.

Configuration

You can use these properties to configure the functionality of this activity.

- **Unique Environment Name**: Identifies the Galaxy and Runtime node on which the galaxy is deployed to perform the read or write operation on the System Platform attribute.
  - You can manually enter the value for this property. This value will be considered for further execution of the current activity.

    Or
  - If the property does not have a value, then the value defined for this property in the "ArchestrAXMLData" XML variable is used during the execution of the **AttributeRead/Write** activity. The "ArchestrAXMLData" XML variable is automatically created when you associate a workflow with an ArchestrA event.

    See also: Configure AttributeReadWrite Activity to Perform Read Operation and Configure AttributeReadWrite Activity to Perform Write Operation.
- **Relative Reference Object**: Relative references are especially useful in templates while passing context parameters to workflows.
  - You can manually enter the value for this property. This value will be considered for further execution of the current activity.

    Or
  - If the property does not have a value, then the value defined for this property in the "ArchestrAXMLData" XML variable is used during the execution of the **AttributeRead/Write** activity. The "ArchestrAXMLData" XML variable is automatically created when you associate a workflow with an ArchestrA event.

    For more information, see Relative Referencing.
- **Retry Count**: Use this property to enter the number of retries, in case of a communication error. If the error is due to invalid data, it will not be retried.

  Property Type: Optional. Set this property only if required.
- **Retry Wait Time**: Use this property to enter retry wait time in seconds, in case of an error.

  Property Type: Optional. Set this property only if required.
- **Attribute Operation**: Define the attribute operation type. The attribute operation type defines whether the action should be read or write.
- **Select Attribute**: Select the attribute type for the Attribution Operation.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Looping Behavior

You can use this property to specify the looping behavior of the activity.

Loop through this action if output is: This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property. Click this property to see the default configured mapped error outputs in red with their check boxes selected.

Configure AttributeReadWrite Activity to Perform Read Operation

You can read entire UDA of array type from System Platform and assign the same to variables of type array within a workflow. You can also read specific index of UDA of type array from System Platform and assign the same to variables within a workflow.

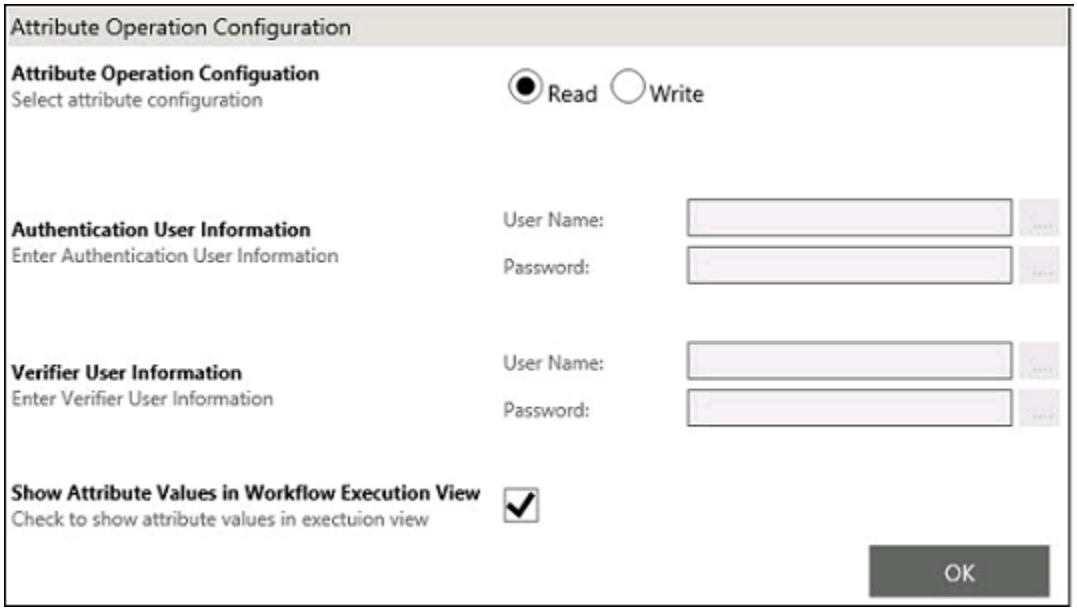
To read a specific index of UDA of type array from System Platform use Expression Builder.

Example: To read the 5th Element of the array , enter `return "UserDefined_001.Array1[4]";` in the expression builder.

To configure AttributeReadWrite activity to perform a read operation

1. Drag-and-drop the **AttributeReadWrite** activity to the Process Designer window.
2. Set the unique environment name from Expression Editor.
3. Click the edit icon in the **Attribute Operation** property.

   The Attribute Operation Configuration dialog box appears.



4. Select **Read**, and then click **OK**.
5. In the **Condition** drop-down menu, select a variable to store the attribute value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

You can add the variable and type from **Variables** property in the **Start** activity.

In a Remote Galaxy, the value of update variable for the **Select Attribute** is in the following format:

Remote or Paired Galaxy_Name:ObjectName.UDA name

In a local Galaxy, the value of update variable for the **Select Attribute** is in the following format:

Me.UDA name

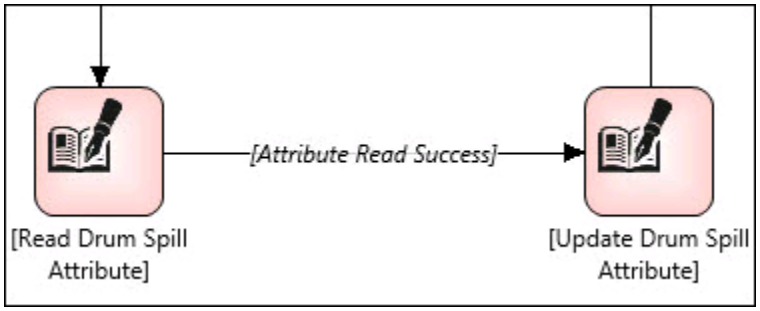
6. In the Actions tree menu, select the **Value** drop-down menu, and then select the Galaxy Browser attributes from which you want the read action to be performed.

   The Galaxy Browser option appears only if you design the workflow in the IDE context.

   The Relative Reference dialog box appears.

   This dialog box displays support for relative referencing. Except for Me, you must manually enter all other relative references. For more details about relative referencing, see Relative Referencing.

7. Create a link between the next activity based on the output options **Attribute Read Success** *or* **Attribute Read Failure**.



8. Publish the workflow.

   You can view the workflow in the Process Execution View dialog box.

   In the **Execution View**, the details of the Read/Write activity executed in a workflow appears in the **Step Details** tab. The **Execution View** page can be opened by double-clicking a top level row in any of the reports. The **View** represents the workflow up to the activity or execution status where the view is opened.

If any error occurs during Read operation, the error message will be logged with an appropriate error code generated from System Platform in the ArchestrA Logger.

For more information about the error logs, see *MxStatusDetail ValuesSystem* under the section *Status and Error Codes* of the *MXAccess Toolkit User's Guide* of *System Platform* documentation.

Configure AttributeReadWrite Activity to Perform Write Operation

You can write entire UDA of array type to System Platform using a variable of array type within a workflow. However, unlike Read operation, you cannot write specific index of UDA of array type.

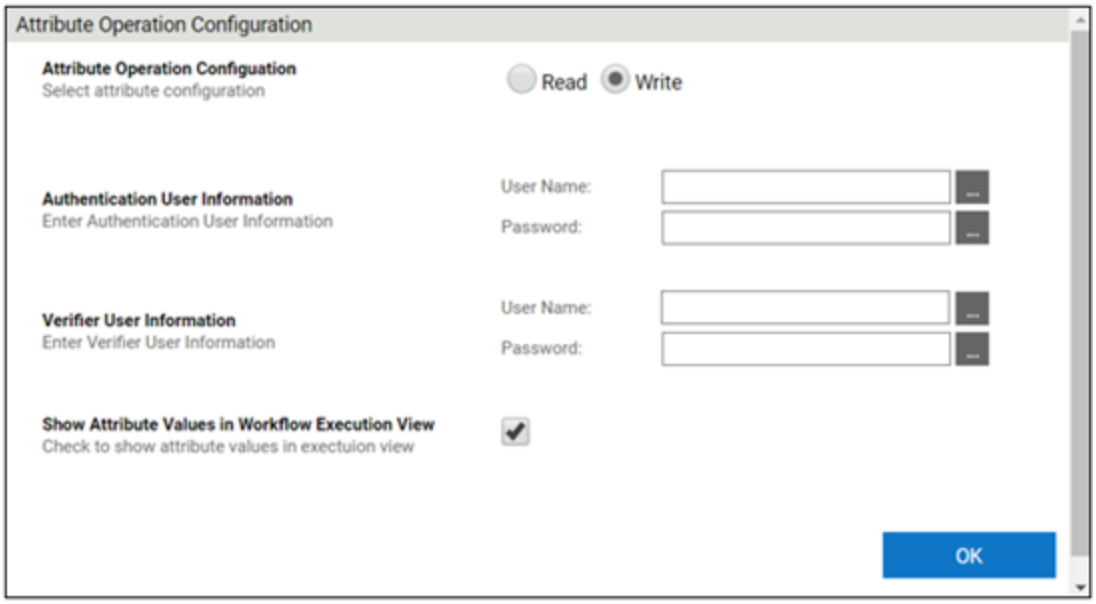
To configure AttributeReadWrite activity to perform a write operation

1. Drag-and-drop the AttributeReadWrite activity to the **Process Designer** page.
2. Set the unique environment name from Expression Editor.
3. Click the edit icon in the **Attribute Operation** property.

   The Attribute Operation Configuration window appears.

4. Select **Write** option in the Attribute Operation Configuration window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

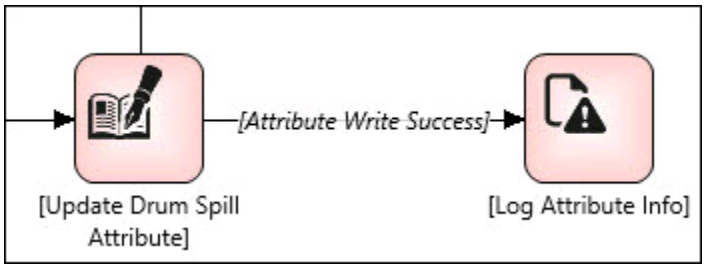


5. Enter the **User Name** and **Password** for **Authentication User Information** to authenticate user information. This credential information is needed for **Operate**, **SecuredWrite**, and **VerifiedWrite** user accounts.
6. Enter the **User Name** and **Password** for **Verifier User Information** to verify user information. This credential information is needed for **VerifiedWrite** user account.
7. Click **OK**.
8. Set the authentication information from **Variable**, **Content**,or **XML variable**.
9. Select the desired **Type** from the drop-down list, and click **Update**.
10. Select the attribute to which the write action needs to be performed in the **Condition** drop-down list. Using expression builder or a variable get the value.

    The variable and type can be added from **Variables** property in **Start** activity.

    For write operation, the Galaxy Browser appears in the Condition drop-down list. The browser appears only if the workflow is designed in the IDE context.

    The Relative Reference dialog box appears. This dialog box displays support for relative referencing. Except for Me, you must manually enter all other relative references. For more details about relative referencing, see Relative Referencing.
11. Create a link between the next activity based on the output options **Attribute Write Success** or **Attribute Write Failure**.



12. Publish the workflow.

    The workflow can be viewed in the Process Execution View page. In the **Execution View**, the details of the Read/Write activity executed in a workflow appears in the **Step Details** tab. The **Execution View** page can be opened by double-clicking a top level row in any of the reports. The **View** represents the workflow up to the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

activity or execution status where the view is opened.

If any error occurs during Write operation, the error message will be logged with an appropriate error code generated from System Platform in the ArchestrA Logger.

For more information about the error logs, see *MxStatusDetail ValuesSystem* under the section *Status and Error Codes* of the *MXAccess Toolkit User's Guide* of *System Platform* documentation.

Relative Referencing

Relative references are especially useful in templates while passing context parameters to workflows. You can use relative references like *MyContainer, MyPlatform, MyEngine, MyArea, and Me* to refer to contained objects within that container.

Relative referencing supports event context at the parameter level and the attribute read or write at the **AttributeReadWrite** activity level.

The following valid relative references refer to the current object:

- Me
- MyContainer
- MyArea
- MyPlatform
- MyEngine

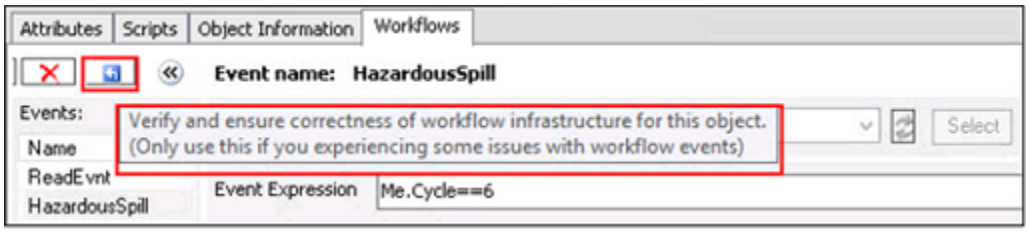
When you use relative references, like *MyContainer*, you can refer to contained objects within that container. For example, a reference to *MyContainer.InletValve.PV* is equivalent to *Tank1.InletValve.PV* in the following hierarchy:

- Tank1: Cannot reference at this level because this is not contained.
- InletValve: Can reference at this level because this object is contained.

Example:

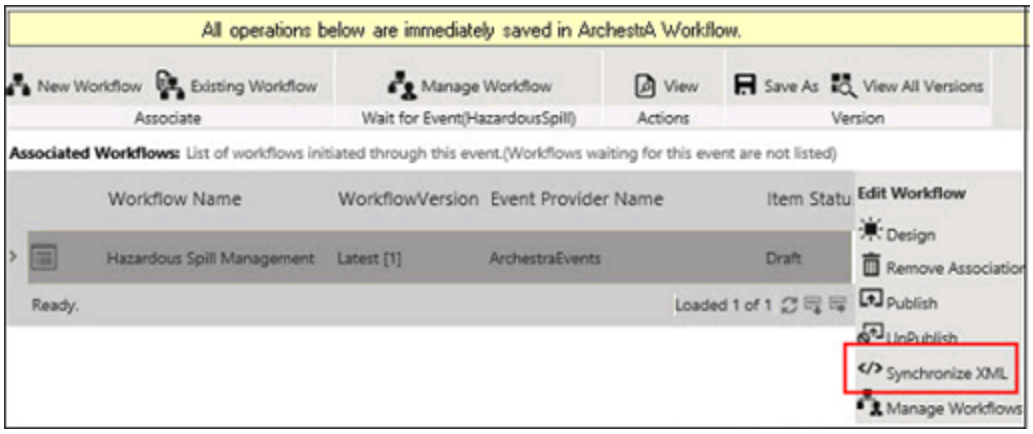
If System Platform-Workflow Connector Version 1.1.1222.0 is used and events to trigger workflows are configured, then do the following steps to use the relative referencing support in the **System Platform-Workflow Connector Version 1.5.1229** for the existing events:

1. Select an existing event, and then click **Verify** to ensure correctness of workflow infrastructure for this object icon.



2. Select a workflow, and then click the **Synchronize XML** option for that workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Deploy the existing gateway object again.

Wait-For-AAEvent Activity

Activity Description:

Use the **Wait-For-AAEvent** activity to wait for a number of events from ArchestrA for a particular time period. You can set the number of events and the time period in the properties of that activity. It can store the event context details into specified XML variables.

Activity Properties:

You can configure the **Wait-For-AAEvent** activity by specifying appropriate values for the properties in the **Activity Properties** area. Click **Activity Properties** in the toolbar or **select** the Properties option from the short-cut menu of the **Wait-For-AAEvent** activity to access the **Activity Properties**.

Name & Description

You can use these properties to specify the name and description for the activity

- **Name**: Use this property to specify a name for the activity.

  Property Type: Mandatory.
- **Description**: Use this property to add a brief description about the activity.

  Property Type: Optional. Set this property only if required.

Configuration

Use these properties to configure the functionality of the activity.

- **Set Event Filters**: Use this property to filter the ArchestrA event. Whenever an event occurs, it checks this filter, and if a match is found, the activity is alerted. When you open the **Properties** association page in System Platform IDE, it populates all the default values.

  Property Type: Mandatory

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Valid for Events**: Use this property to specify the number of events for which the activity must wait. It indicates that the activity must wait till the specified number of events are received.

  Property Type: Optional. Set this property only if required.
- **Select Output XML Variable**: Use this property to set XML Variable for storing event context. A new XML variable is created or an existing XML variable of the similar schema is selected. When an event occurs, this XML variable is populated with event context data and it is used later in the flow of the process. The synchronization option is used if the event parameters in the IDE are modified later. For more information about synchronization, see *User Guide*.

Escalation & Timeout

Use this property to specify escalation and timeout settings for the activity.

- **Timeout for Action**: Use this property to set timeouts for this activity. If specified, the activity will be timed out after the specified time span.

  Property Type: Optional. Set this property only if required.

Miscellaneous

- **Update List XMLVariables**: Use this property to insert records into AVEVA Work Tasks Information List. This property is automatically set when you create an XML variable using **Select Output XML Variable**.

  Property Type: Optional. Set this property only if required.

Looping Behavior

Use this property to specify the looping behavior of the activity.

- **Loop through this action if output is**: Use this property to specify the condition for looping the activity by selecting an activity output from the drop down.

  Property Type: Optional. Set this property only if required.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property. Click on this to see the default configured mapped error outputs in red with their check boxes selected.

Activity Outputs:

The **Wait-For-AAEvent Activity** returns the following values:

- **Event Received**: This output is displayed when an ArchestrA event matching the specified filter condition is received.
- **Action Completed**: This output is displayed when the activity has timed out.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Workflow Features Inside InTouch

This topic provides information on how to use the Workflow Features inside InTouch.

- Initial Steps
- Inbox Inside InTouch
- Workflow Status View Inside InTouch
- BAM Inside InTouch
- Lists Inside InTouch
- Forms Inside InTouch
- List Forms Inside InTouch
- WorkItemInboxCount in InTouch

Initial Steps

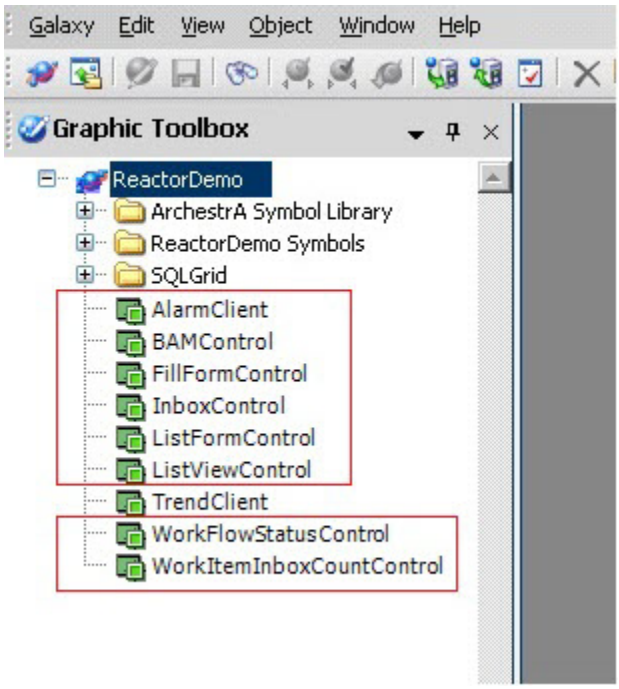
You can use workflow features from Inside InTouch.

1. In the **System Platform IDE**, click **Galaxy**, click **Import**, and then click **Client Control**.

   The Import Client Control(s) dialog box appears.
2. Go to the following path and import `ArchestrA.InTouch.SkeltaControls.dll`:

   C:\Program Files\ArchestrA\Workflow\InTouch\ArchestrA.InTouch.SkeltaControls.dll

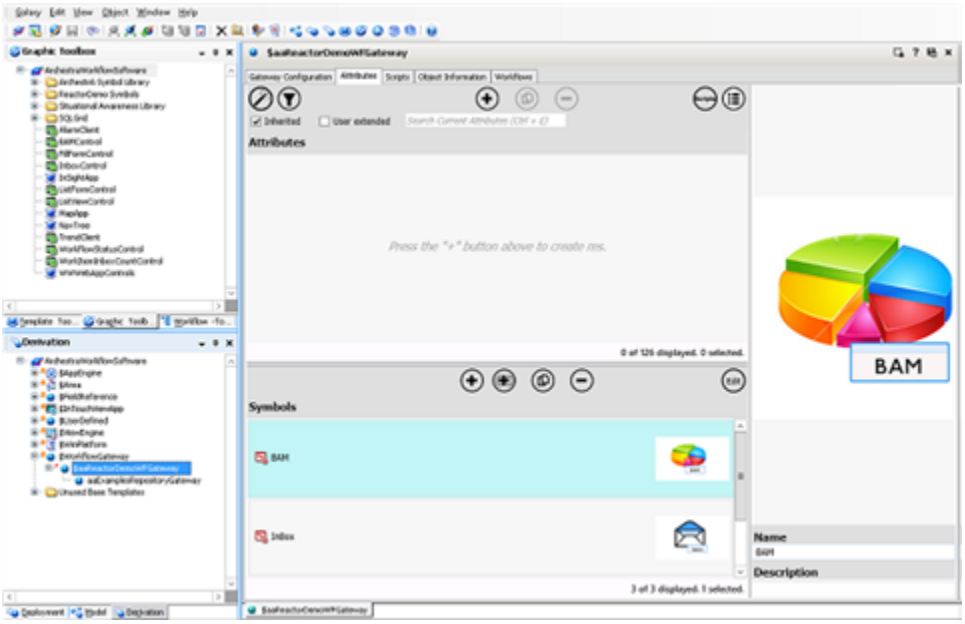
   The imported graphic object appears in the Graphic Toolbox in System Platform IDE as shown in the following image:



   If the same version of the control exists, then you will be prompted to rename the existing controls.
3. To embed the graphics in an ArchestrA object, open the ArchestrA object, click the Attributes tab, and then

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

click **+**.



4. Click the **Add** button, and then enter the name of the graphic symbol that is added in the **Name** field.
5. Click **Edit** to create the symbol in the Graphic Editor.

   The Graphic Editor appears.

6. Click the **Embed graphic** icon.

   The ArchestrA Object browser appears.
7. In the ArchestrA Object browser, select the **Graphic Toolbox**, and then select the required graphic control.

Note:
- The control must be placed on the **Workflow Gateway** object.
- In case the control is placed on a user-defined object, then ensure:
-- The object has an attribute called **ConnectionString**.
--- The value of this **ConnectionString** attribute is same as that of the **ConnectionString** attribute (hidden) of the **Workflow Gateway** object.
- To set up the zoom level, refer to Setting Zoom for the Web Browser Control in System Platform and InTouch.

AIM Authentication for Controls in InTouch HMI

You can use AVEVA Identity Manager (AIM) authentication for AVEVA Work Tasks Controls in **InTouch HMI**. To enable AIM authentication, ensure that in **Central Configuration** when the **Resource Provider** is set to **Active Directory**, the **Use Token Service** is set to the appropriate server. Otherwise, the control will be loaded within the context of the default user, if specified. If the default user is not specified, then the control will be loaded

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

within the context of the user who configured the workflow gateway.

Note:
- Ensure to import the ArchestrA app before configuring AIM for InTouch HMI.
- When using AIM authentication for controls in InTouch HMI, ensure all controls are configured for AIM authentication.
- When the InTouch HMI window having AVEVA Work Tasks control is opened for the first time, it navigates to AIM authentication page. After providing the credentials, it authenticates with AIM, and on successful authentication, the controls get loaded.
- If multiple controls are placed in a single InTouch HMI window, then for the first time you may see AIM authentication page on each AVEVA Work Tasks controls. To avoid authenticating each control, do as follows:
-- Open a single AVEVA Work Tasks control in the InTouch HMI window.
-- After AIM authentication with one control, close the window, and then re-open it.

To configure System Platform IDE Security

1. In the **System Platform IDE**, on the **Galaxy** menu, click **Configure**, and then click **Security**.

   The **Security** dialog box appears,
2. On the **Authentication Mode** tab, select **OS User based** and **OS Role based**.
3. On the **Users** tab, under **Authorized Users Available**, add users, and then under **Associated Roles for Administrator**, map **Administrator** role.
4. Click **OK**.

   The **System Platform IDE** will close and reopen.

Sometimes, you may face issues related to Anti-forgery. These issues might be because of cache. To resolve this issue, follow the below steps:

1. Close System Platform IDE.
2. Open browser window.
3. Clear browser cache.
4. Close all browser windows.
5. Open System Platform IDE and try accessing controls.

Inbox Inside InTouch

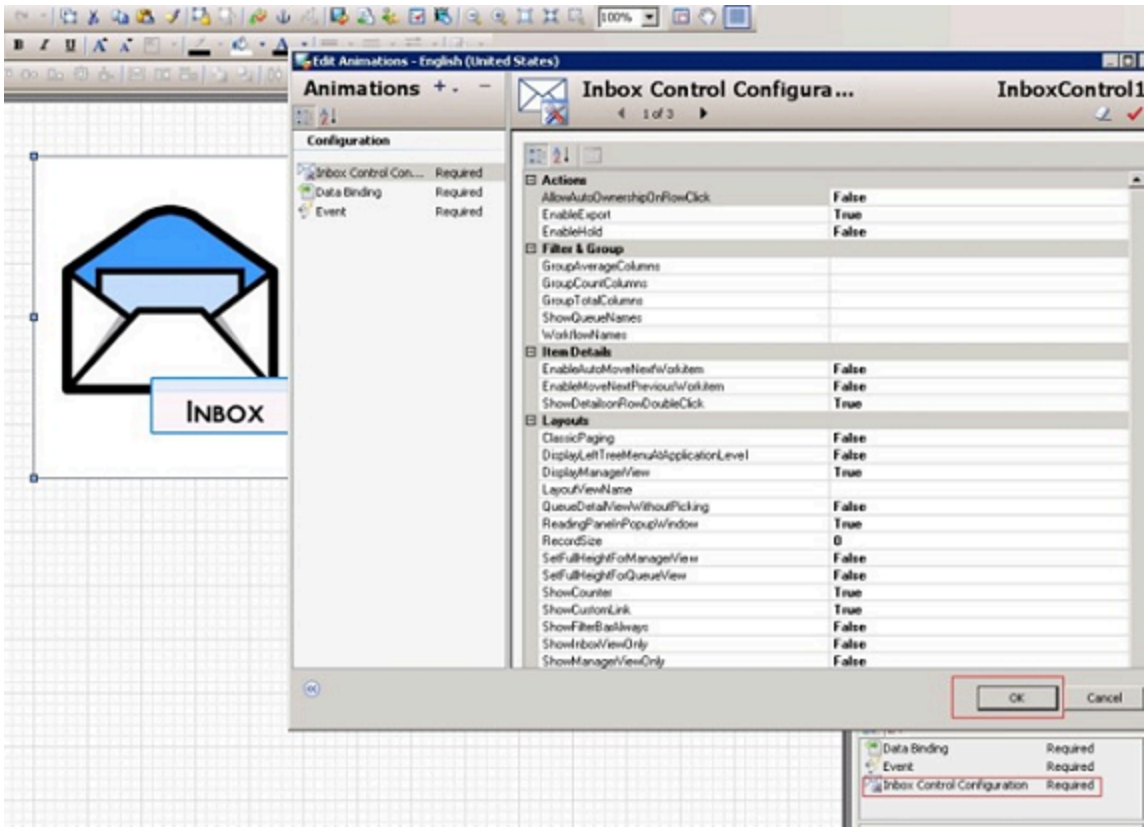
Follow steps 1-7 listed in the Initial Steps and continue:

1. Select the control in the editor.

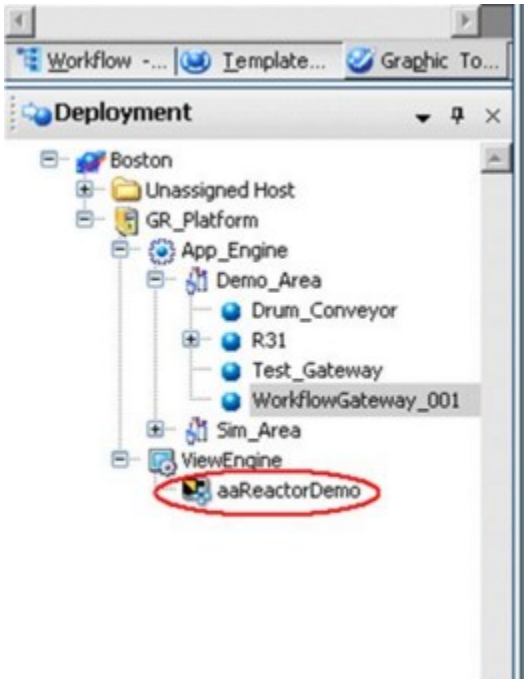
   The Edit Animations window appears.
2. In the Animations pane, double-click and open Inbox Control Configurations.

   A new configuration page appears. You can configure the Inbox Control in this page.
3. After configuring the Inbox, click OK.
4. Save and close the editor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



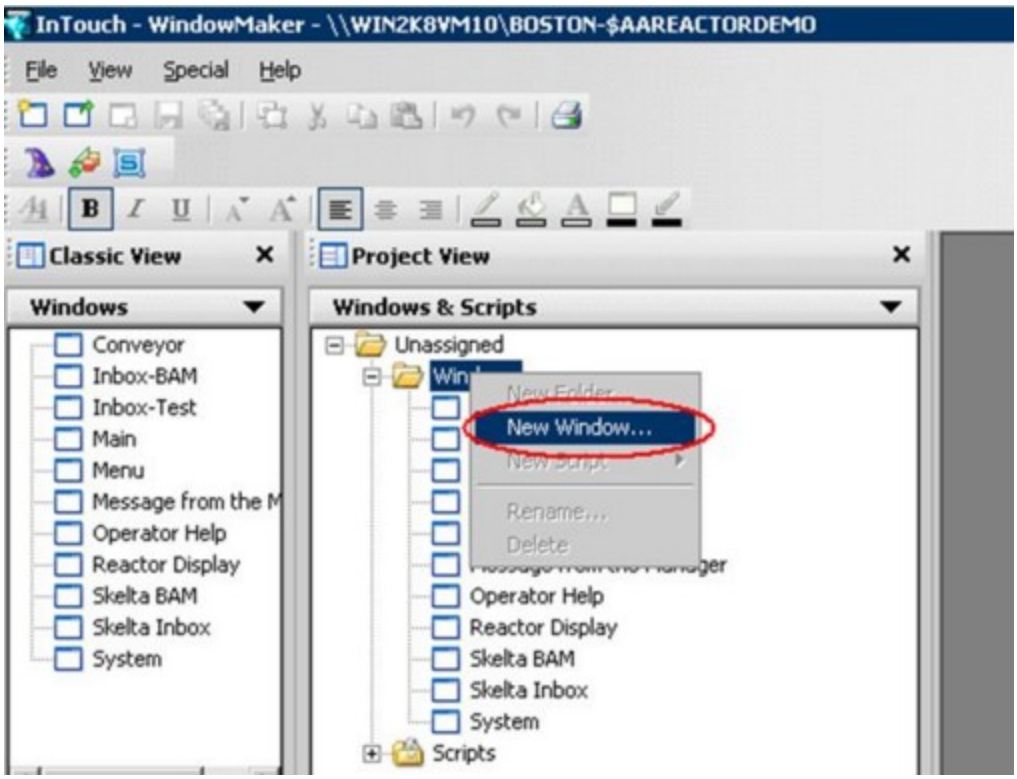
5. Select the instance and deploy in the System Platform IDE.



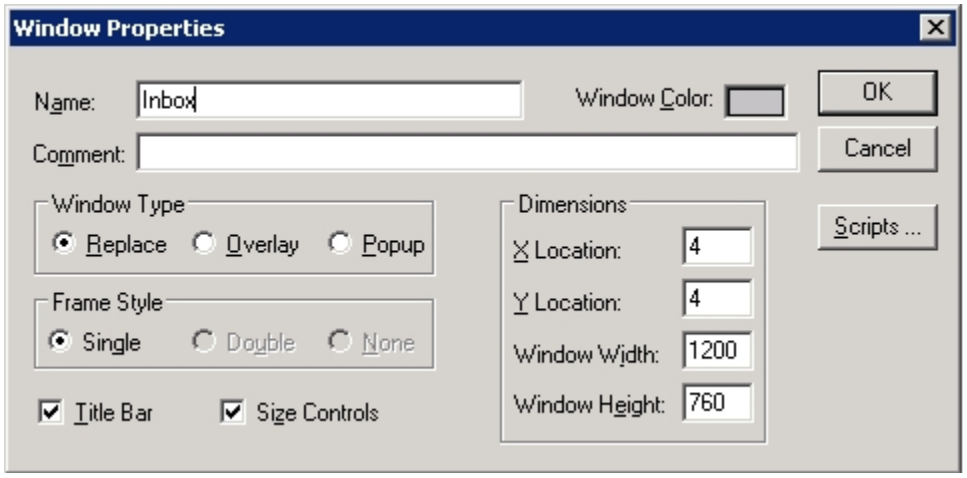
6. Double-click the $InTouchViewApp instance (aaReactorDemo) under ViewEngine from the System Platform IDE.

   The InTouch - WindowMaker window appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

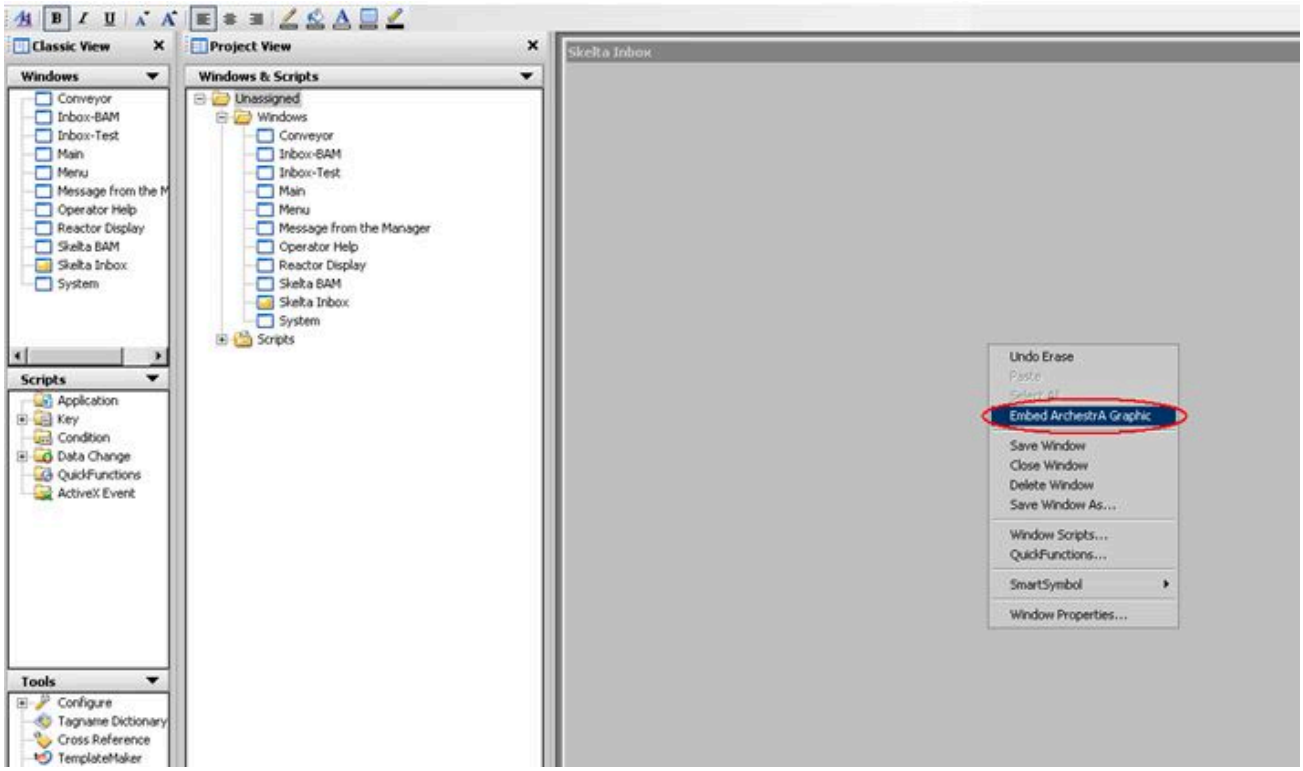


7. Create a new window for the Inbox control.
8. Provide the name and size for the window.

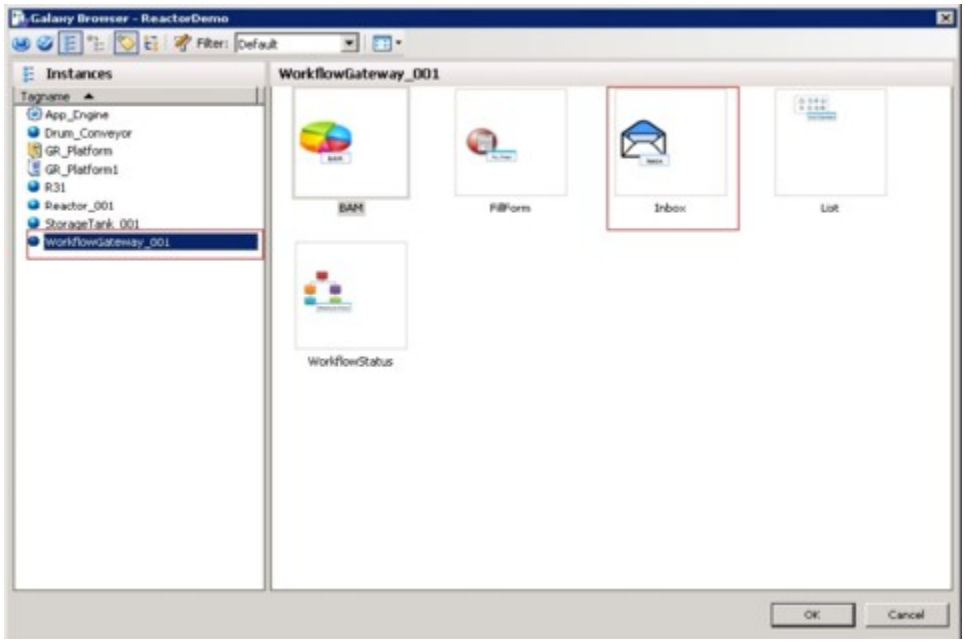


9. Right-click inside the newly opened window, and then select **Embed ArchestrA Graphi**c.
   The ArchestrA Galaxy Browser appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

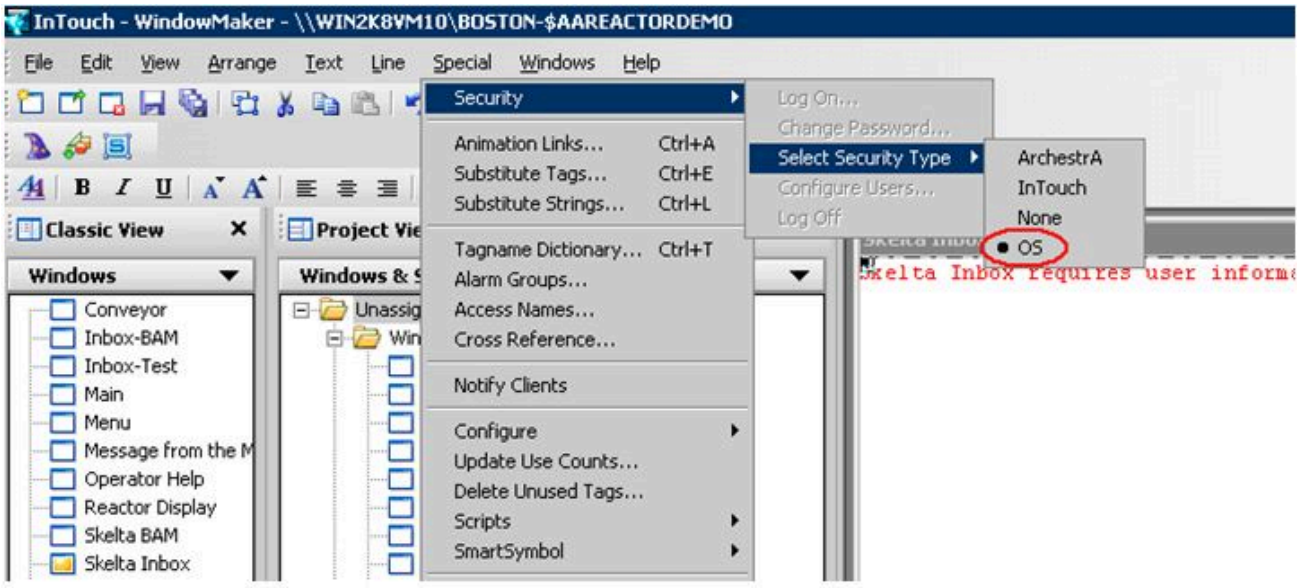


10. Select the Instances tab, and then select the object to which the graphic symbol is added. Select the name of the required graphic object (Inbox).



11. Enable security for the Inbox window, if required.

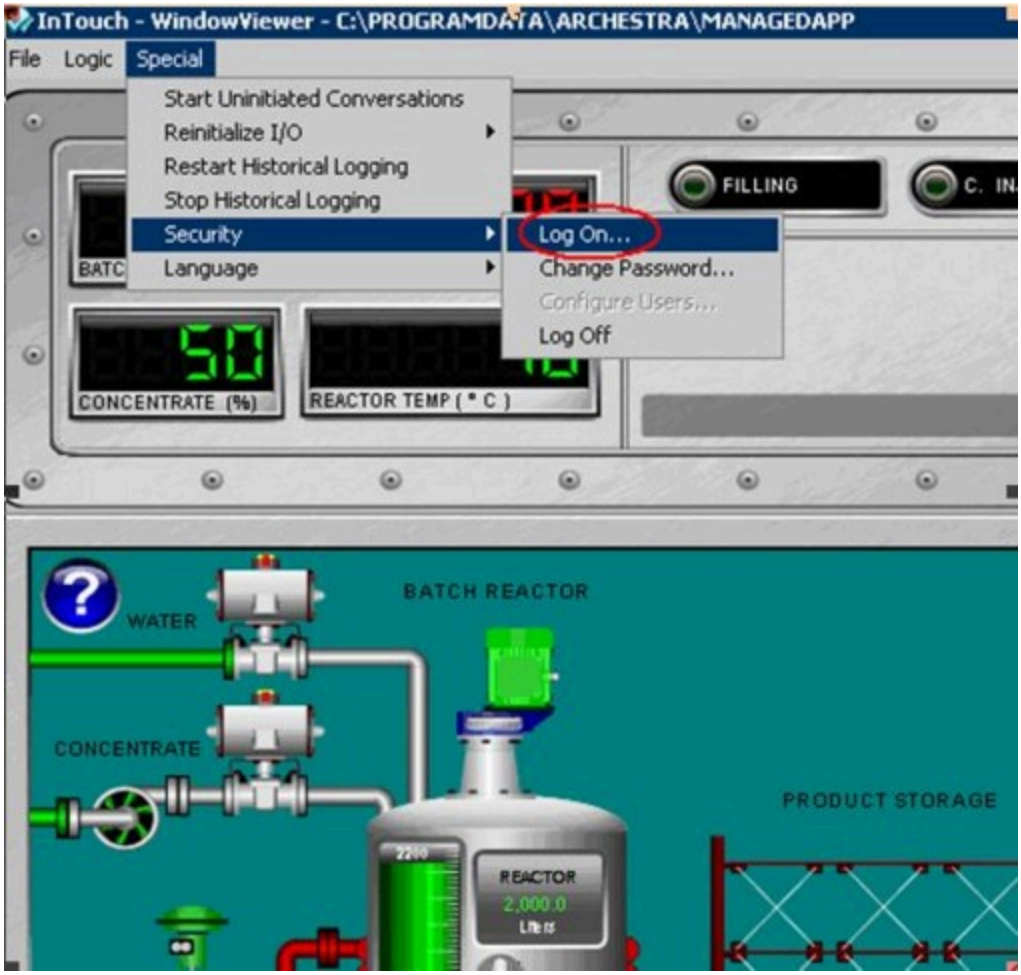
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

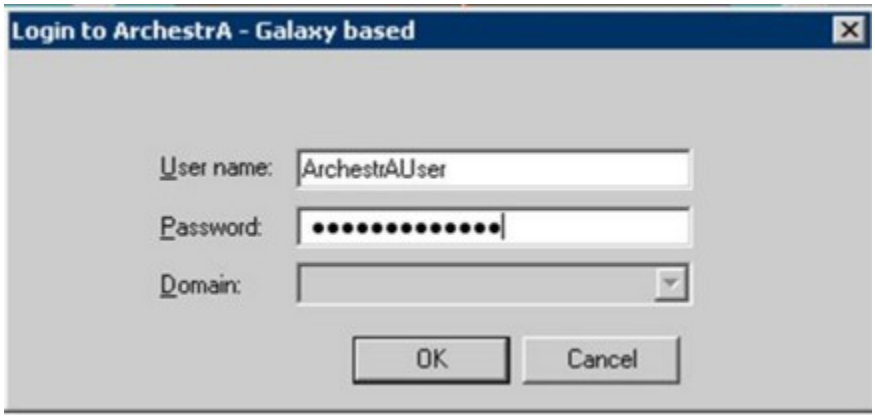


12. Save the window and exit InTouch WindowMaker.
13. Save and close the object in the System Platform IDE.
14. Deploy the $InTouchViewApp instance.
15. Open the InTouch Viewer.
16. Click Special, click Security, and then click Log On.

    The login dialog box appears.

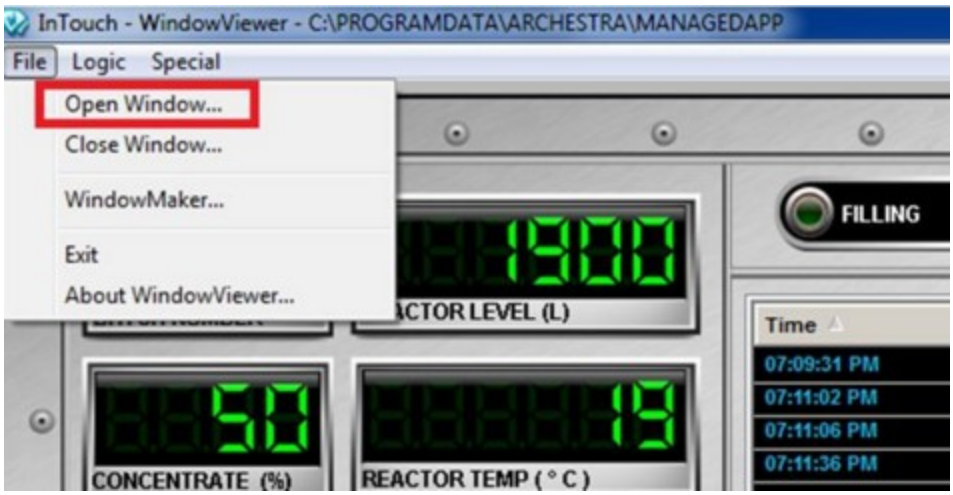
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

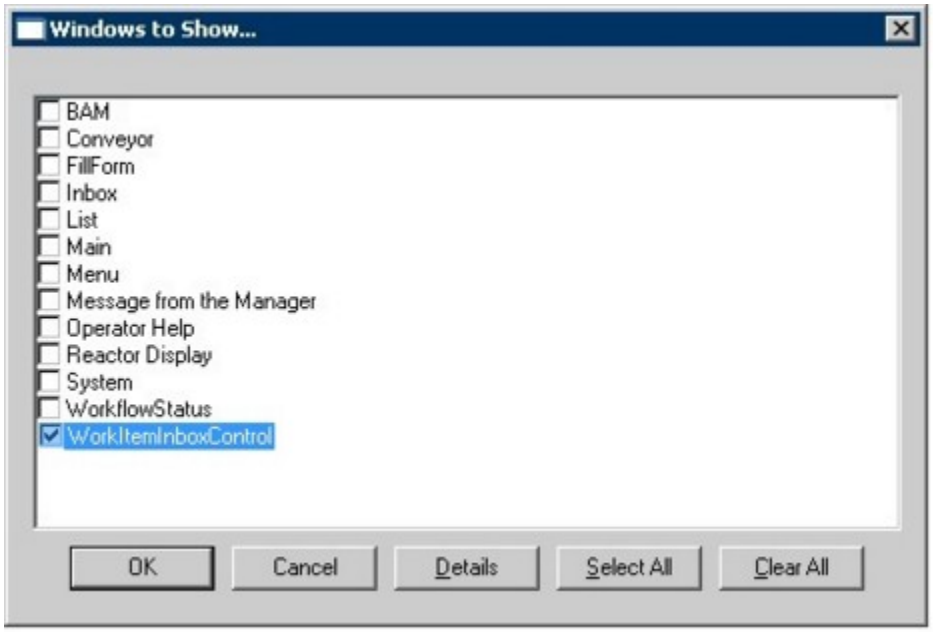




17. Enter the credentials, and then click OK.
18. Open the Inbox (the one created) window from the file menu.
    The Inbox Control loads in the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

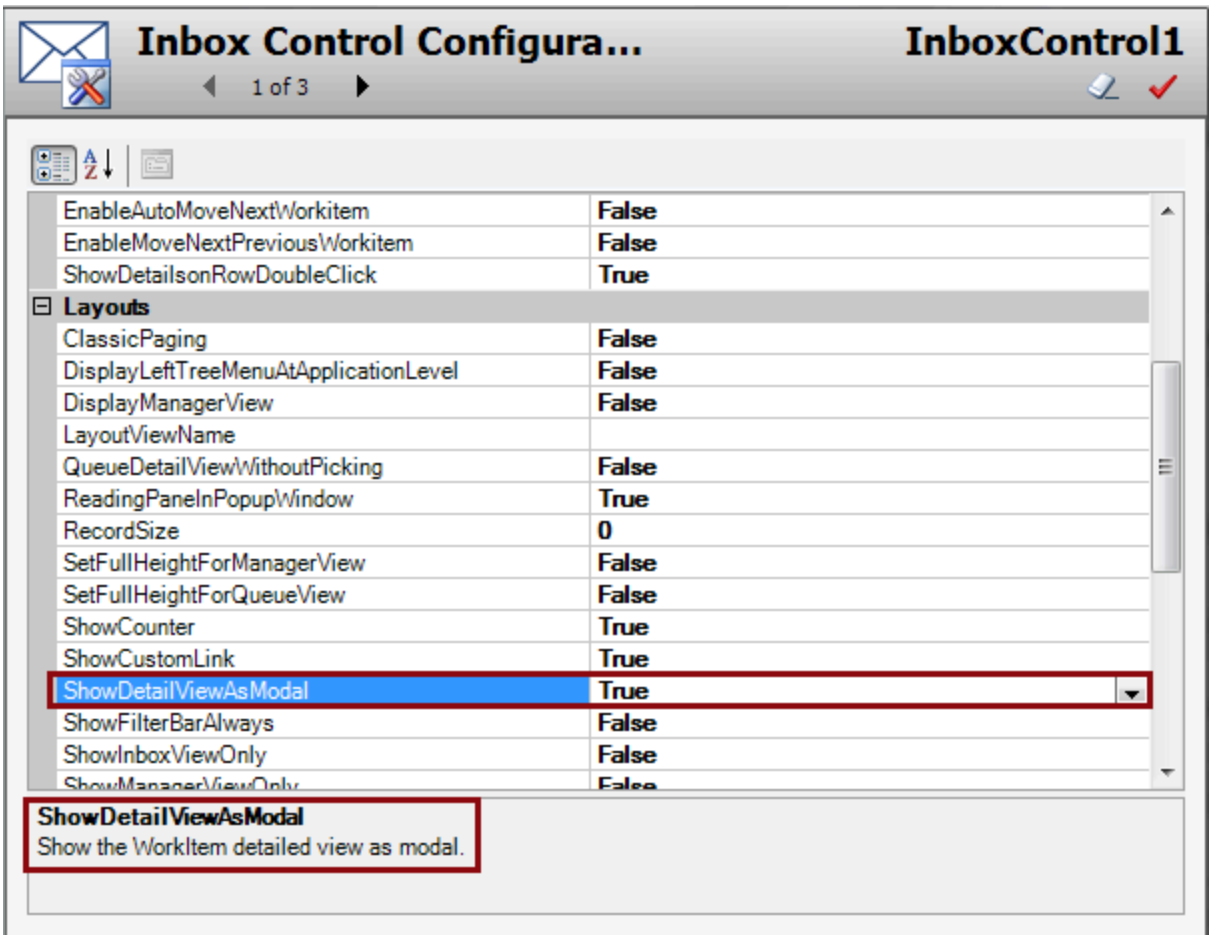




Modal or Modaless Window Property

When you click a work item to edit, you can choose the editor to appear as a modal window or a modaless window. Change the value of the parameter, **ShowDetailViewAsModal** to *True* if you want the editor to open as a modal window, as shown in the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



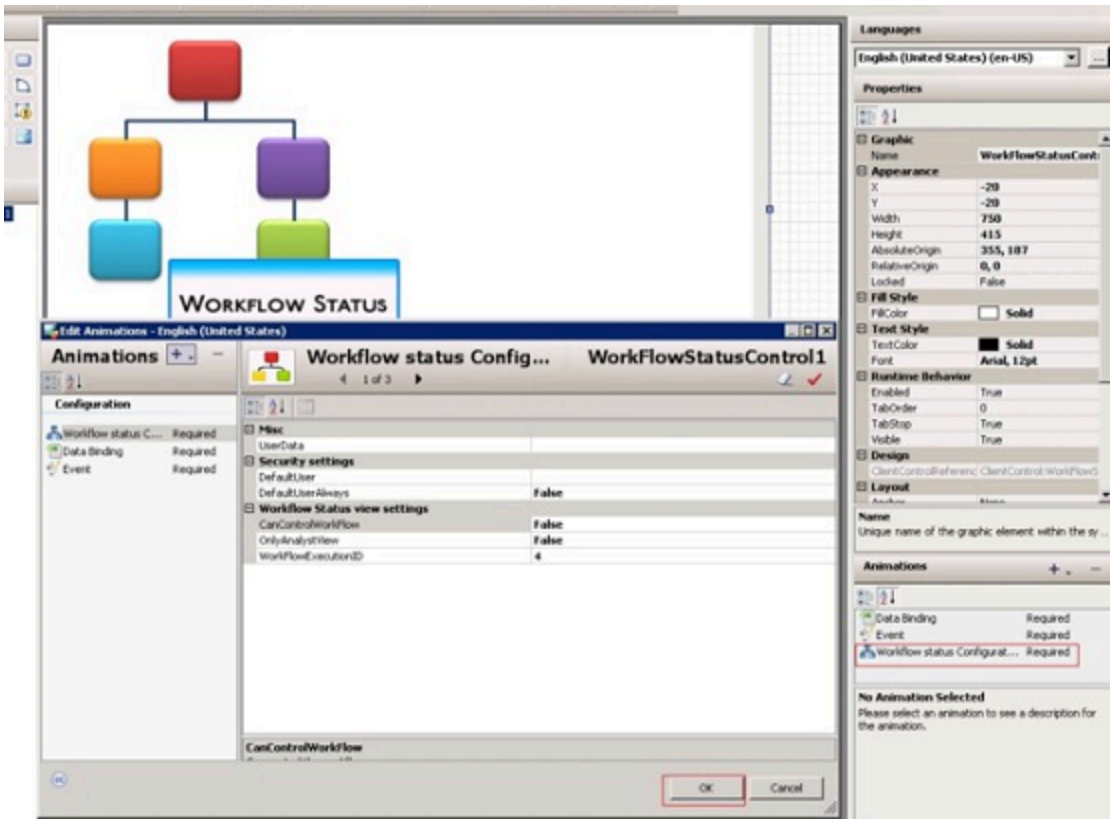
Note: The Inbox Control size must be more than the size of the Work Item Detail View modal window.

Workflow Status View Inside InTouch

Follow Step 1 to Step 7 listed in the Initial Steps and continue:

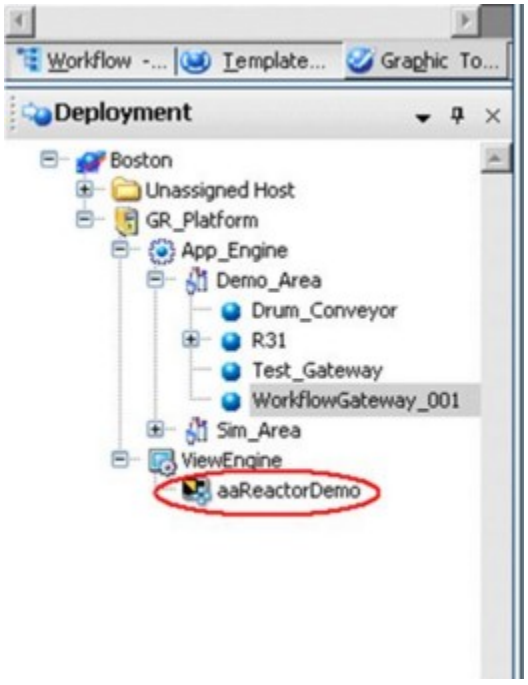
1. Select the control in the editor.
   The Edit Animations window appears.
2. In the Animations pane, double-click and open Workflow status Configuration.
   A new configuration page appears. You can configure the Control in this page.
3. After configuring the Workflow Status, click OK.
4. Save and close the editor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



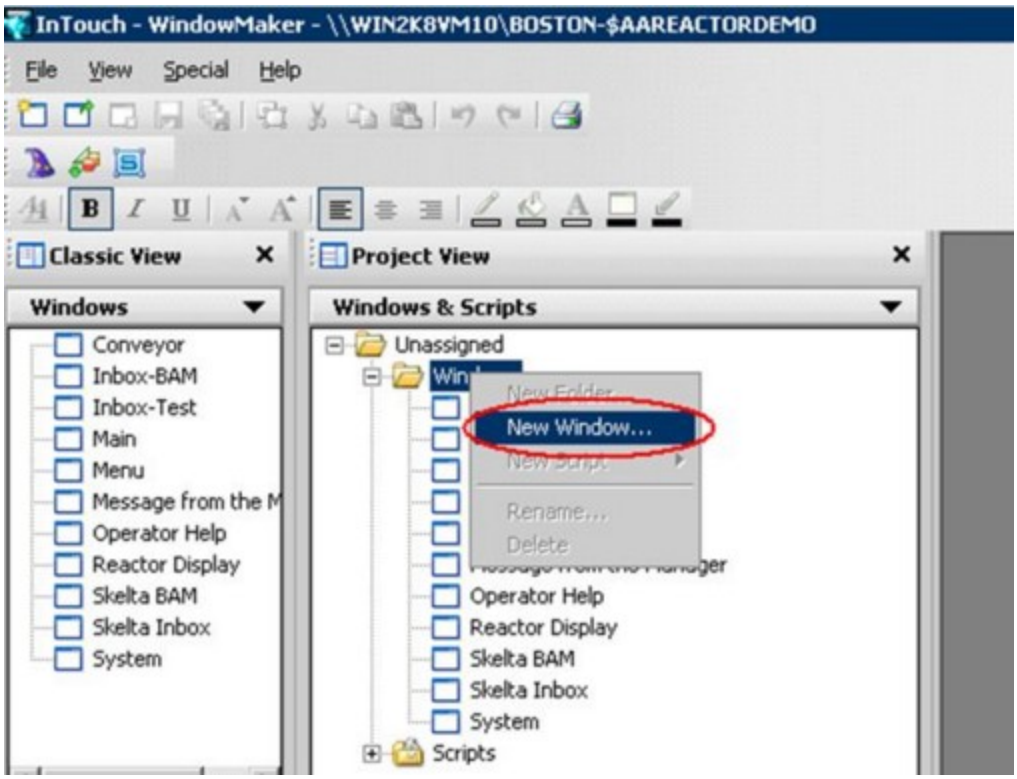
5. Select the instance and deploy in the System Platform IDE.
6. Double click the $InTouchViewApp instance (aaReactorDemo) under ViewEngine from the System Platform IDE.

   The InTouch WindowMaker window appears.

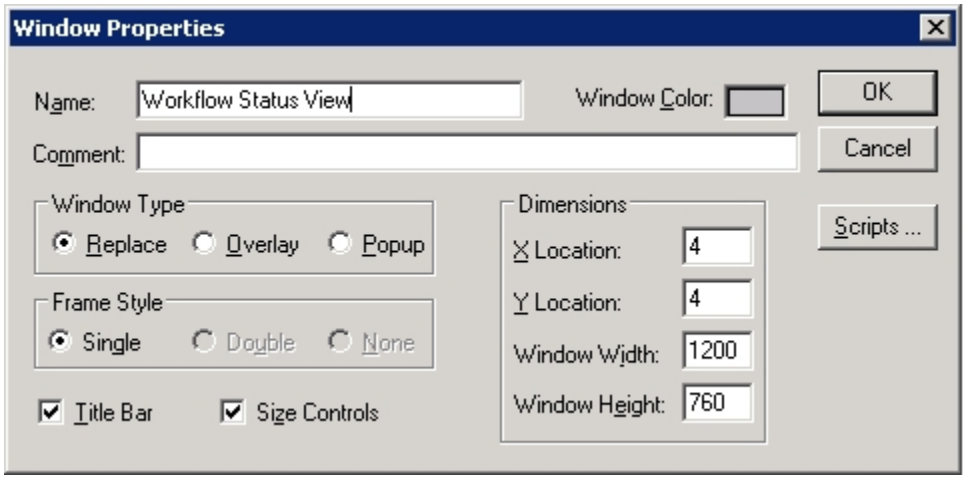


7. Create a new window for the Workflow Status View control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

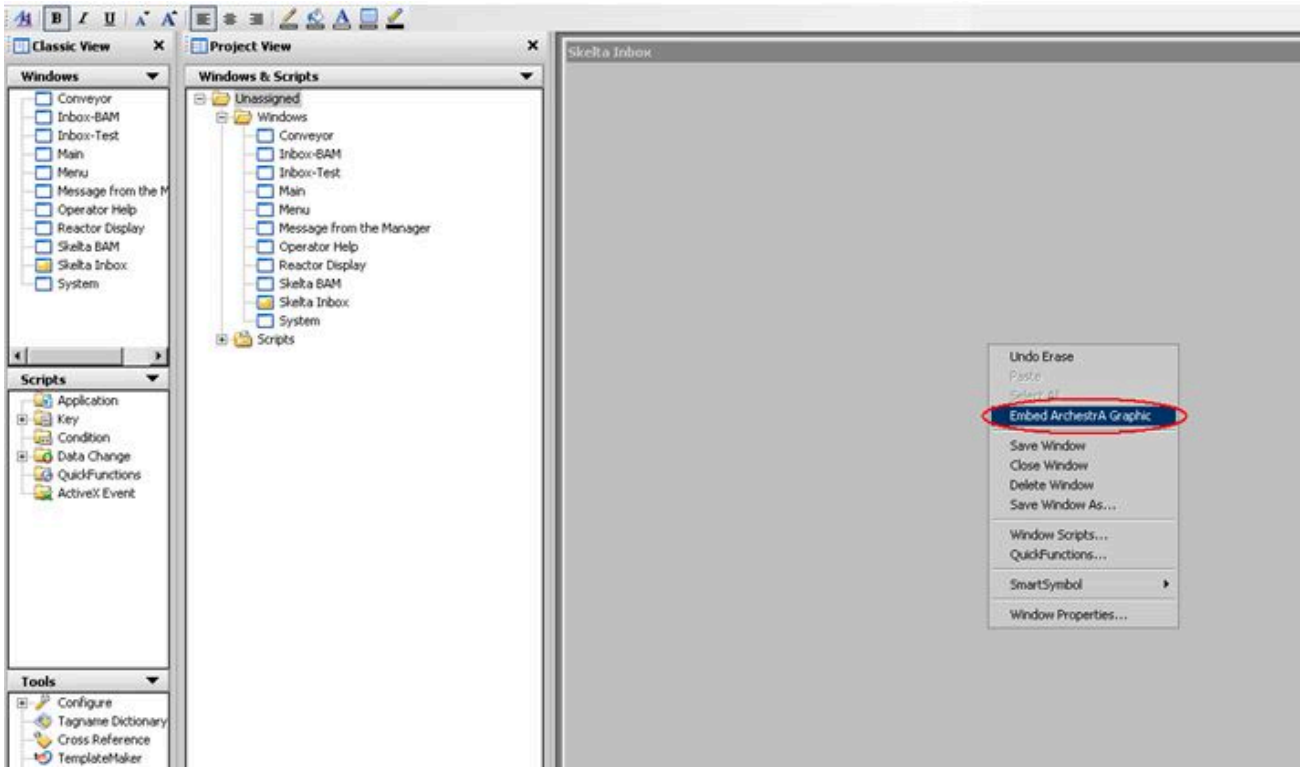


8. Provide the name and size for the window.

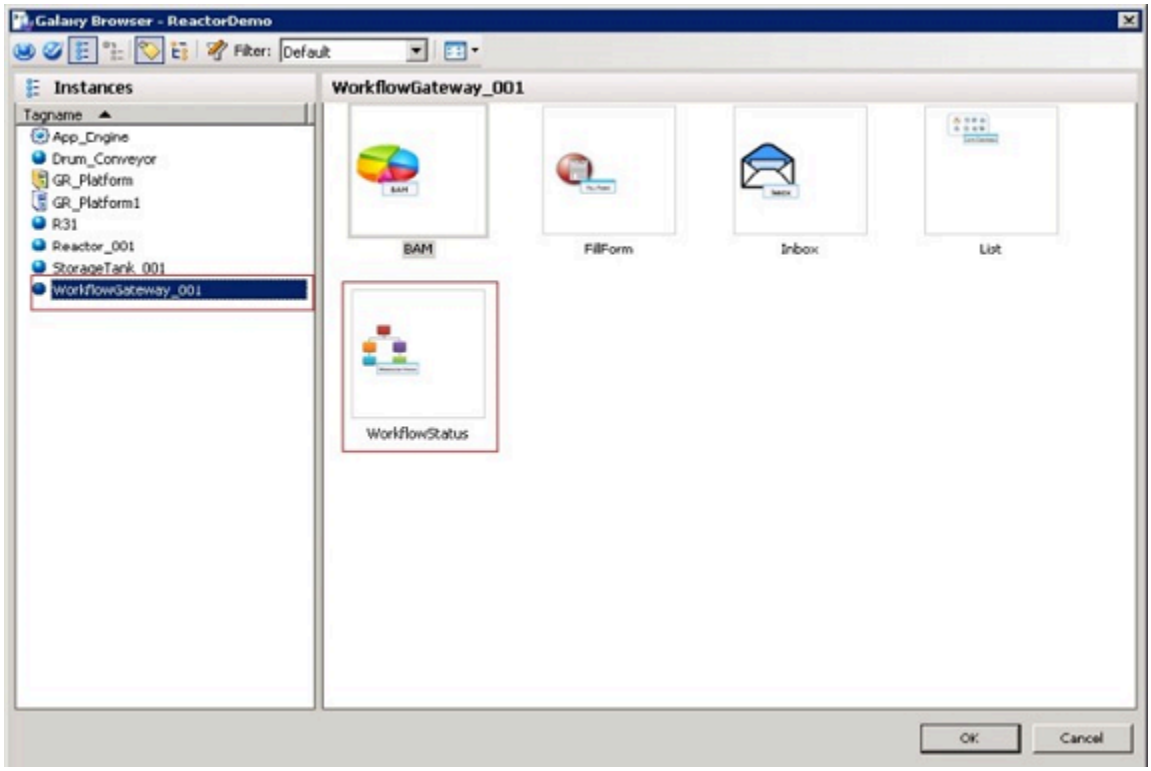


9. Right-click inside the newly opened window and select "Embed ArchestrA Graphic" option.
   The ArchestrA Galaxy browser appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

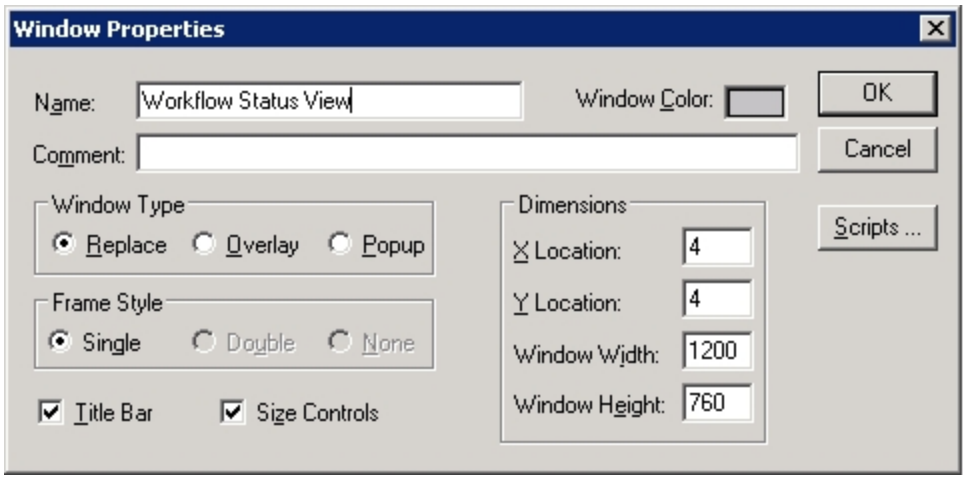


10. Select the Instances tab, and select the object to which the graphic symbol has been added.

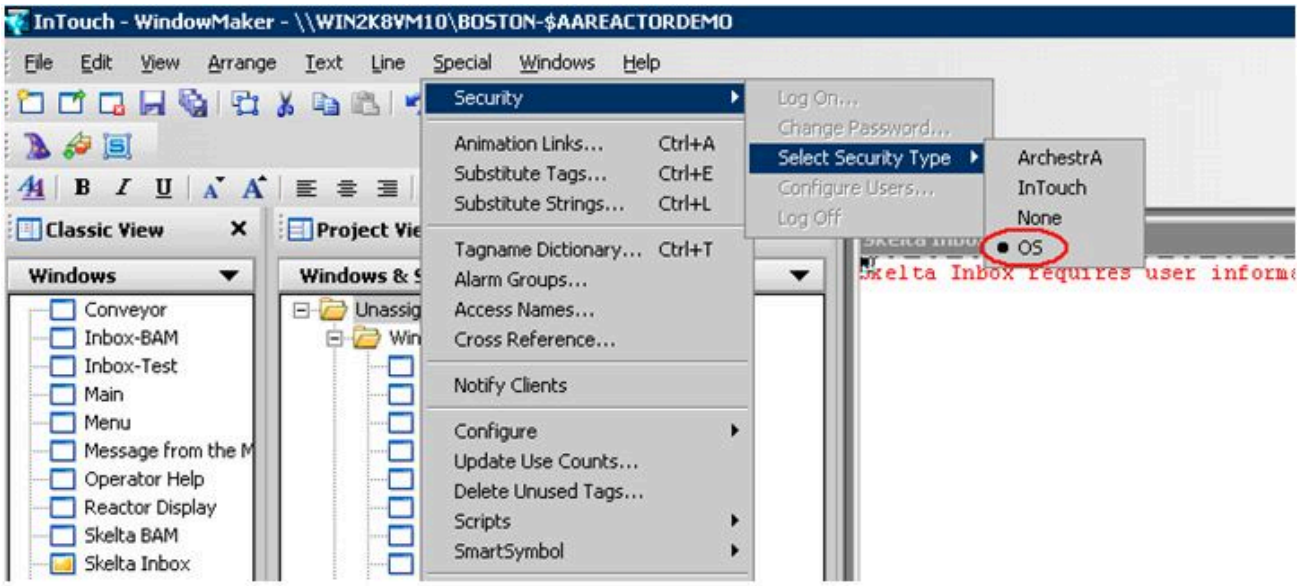


11. Select the name of the graphic object (Workflow Status View) that has to be displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

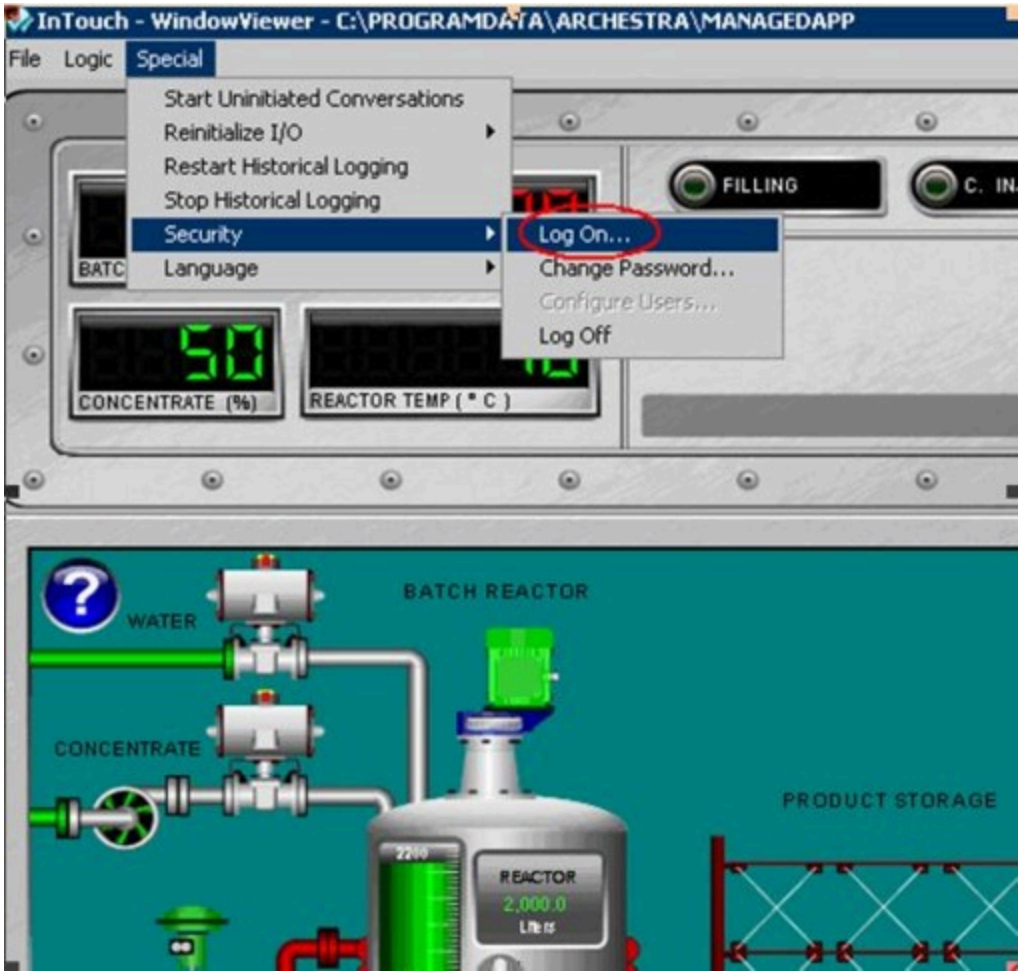


12. Enable security for the Workflow window, if required.

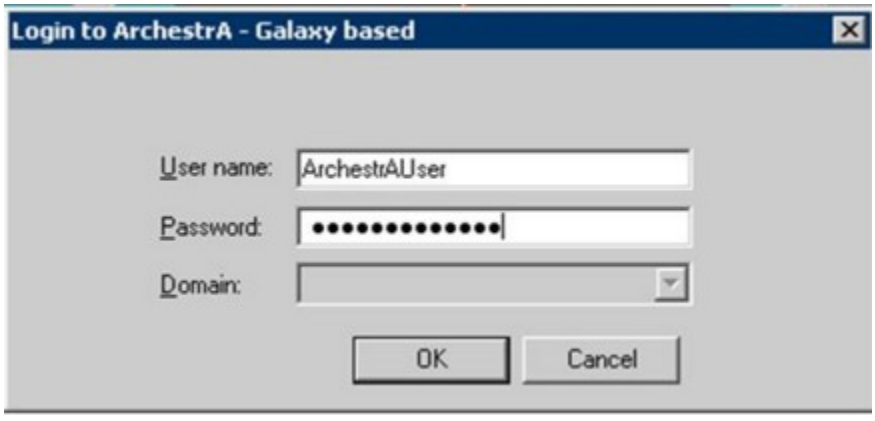


13. Save the window, and exit InTouch WindowMaker.
14. Save and close the object in the System Platform IDE.
15. Deploy the $InTouchViewApp instance.
16. Open the InTouch Viewer.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

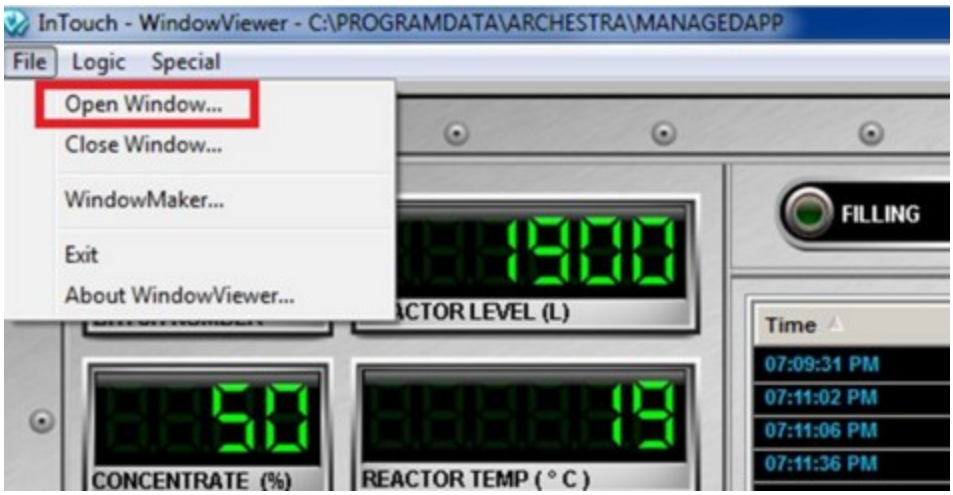


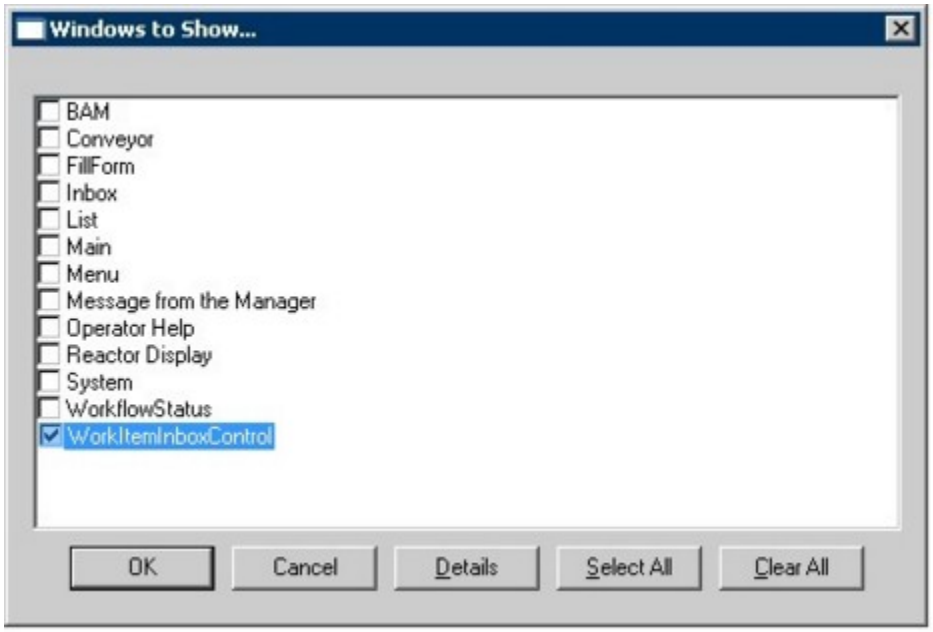
17. Click Special, click Security, and then click Log On.
    The login dialog box appears.
18. Enter the Credentials and click OK.



19. Open the created Workflow Status View window from the file menu.
    The Workflow Status View Control loads in the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



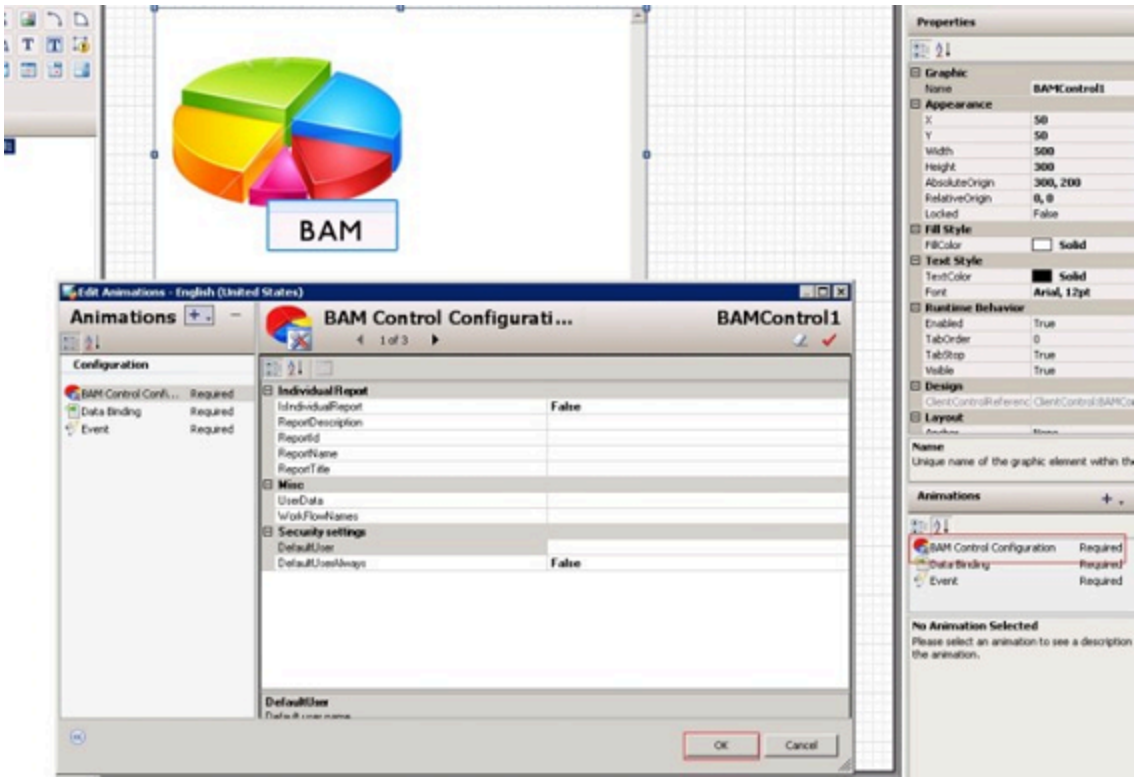


BAM Inside InTouch

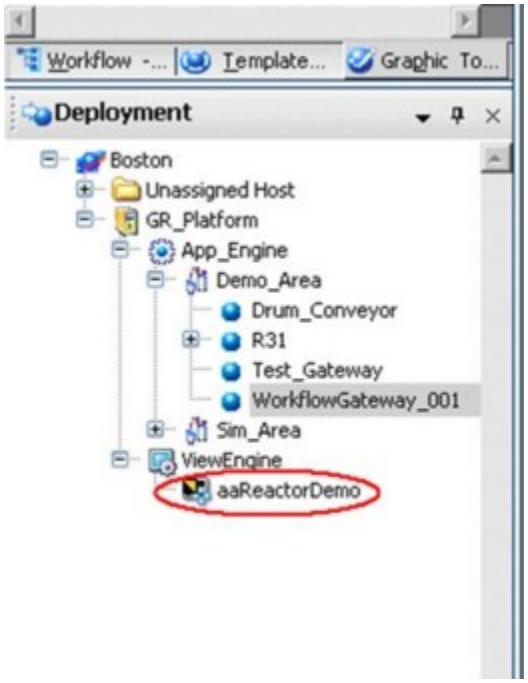
Follow Step 1 to Step 7 listed in the Initial Steps and continue:

1. Select the control in the editor and double click on BAM Configuration under Animations. This will open up a new configuration page from where the BAM control can be configured. After configuring the BAM Configuration, click "OK". Save and close the Editor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

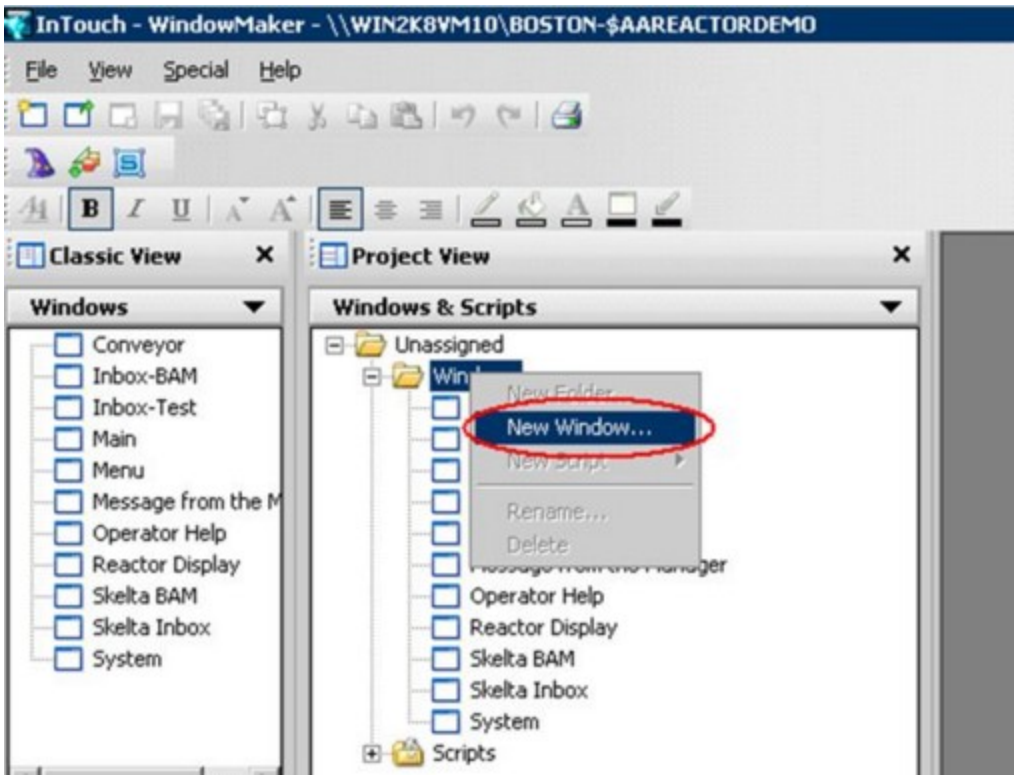


2. Select the instance and deploy in the System Platform IDE, double click the $InTouchViewApp instance (aaReactorDemo) under ViewEngine from the System Platform IDE.

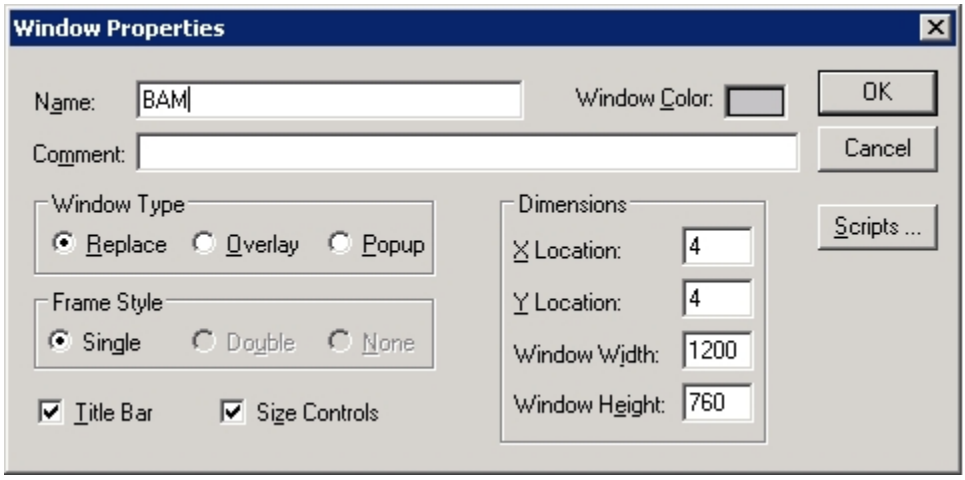


3. The InTouch WindowMaker appears. Create a new window for the BAM control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

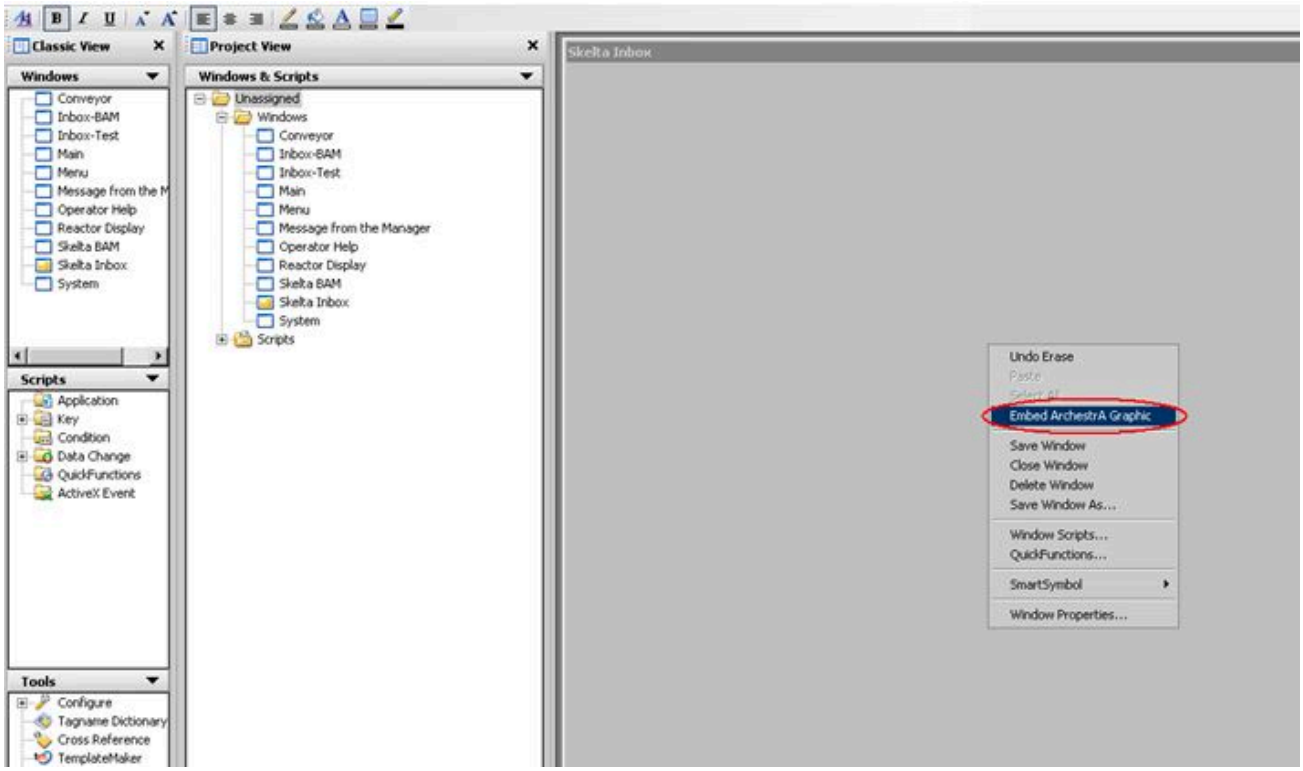


4. Provide the name and size for the window.

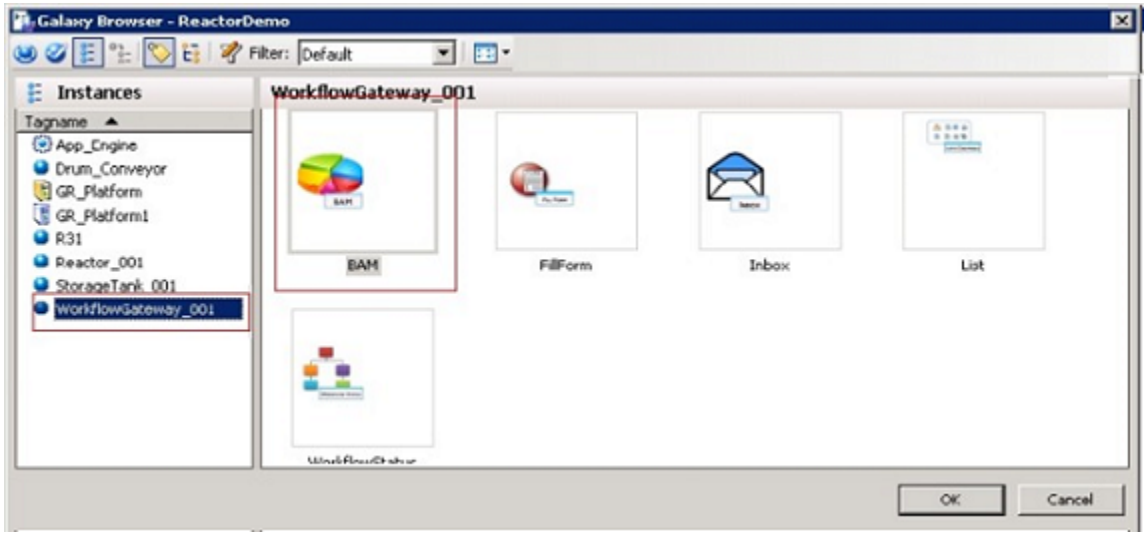


5. Right-click inside the newly opened window and select "Embed ArchestrA Graphic" option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

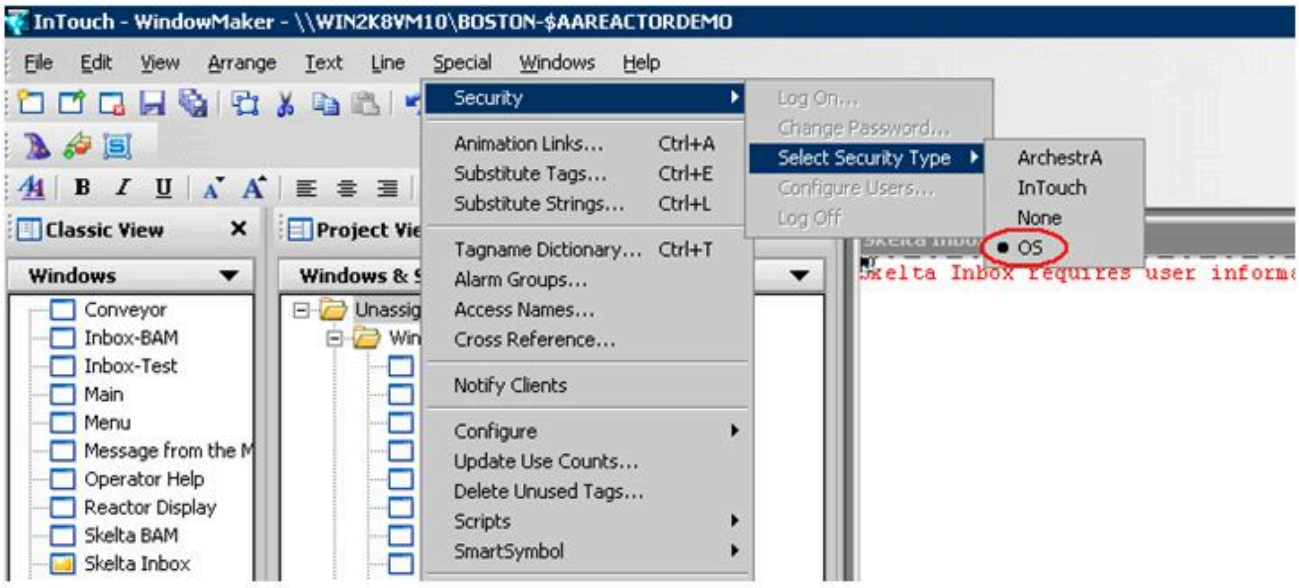


6. This opens up the ArchestrA Galaxy Browser. Select the "Instances" tab and select the object to which the graphic symbol has been added. Select the name of the graphic object (BAM) that has to be displayed.



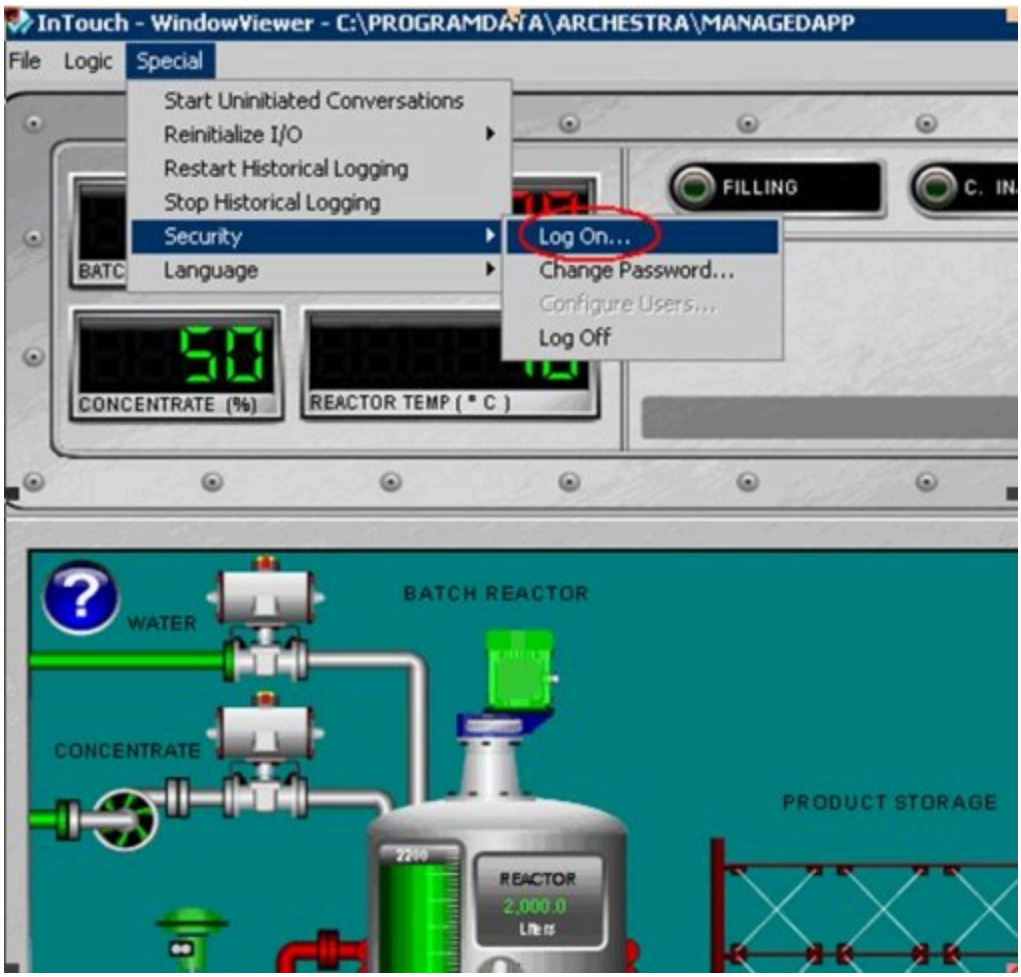
7. Enable security for the BAM window if required.

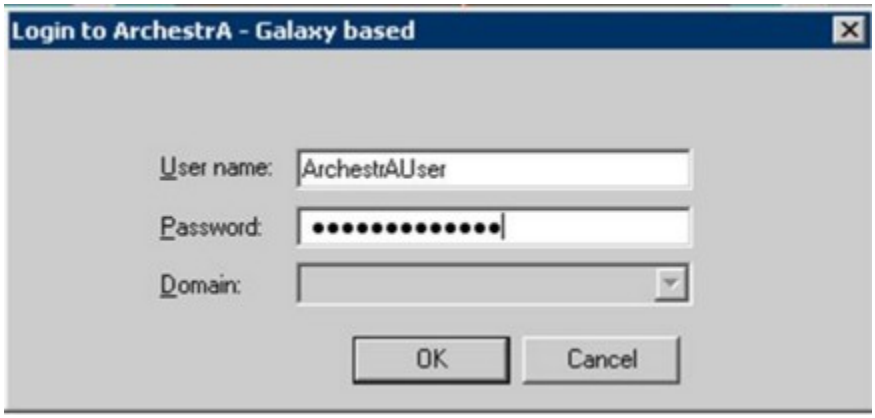
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



8. Save the window and exit InTouch WindowMaker.
9. Save and close the object in the System Platform IDE.
10. Deploy the $InTouchViewApp instance.
11. Open the InTouch Viewer and navigate to Special-> Security->Log On.
12. Enter the Credentials and click OK.

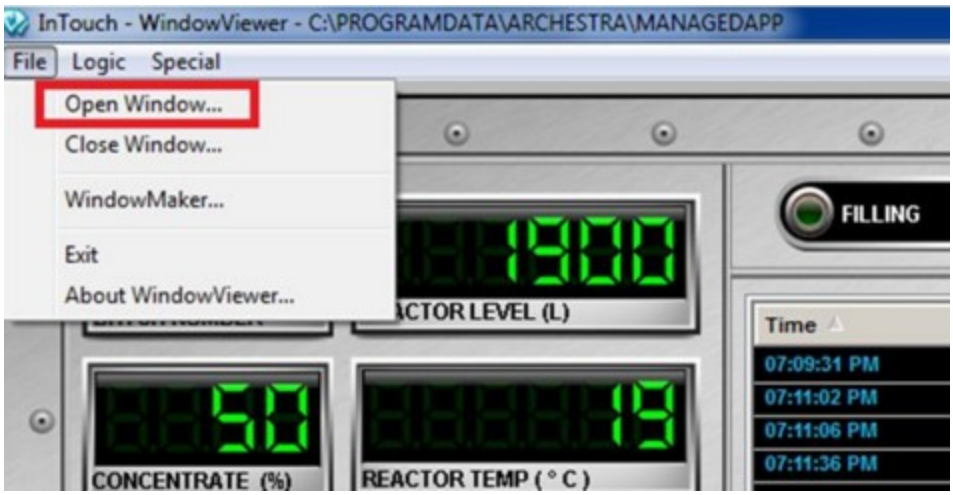
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

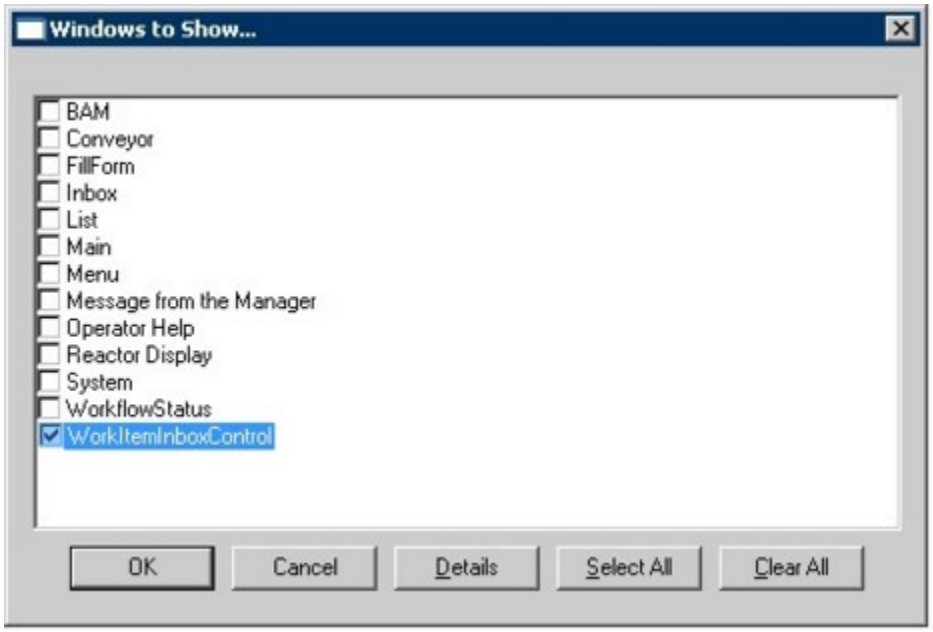




13. Open the created BAM window from the file menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.





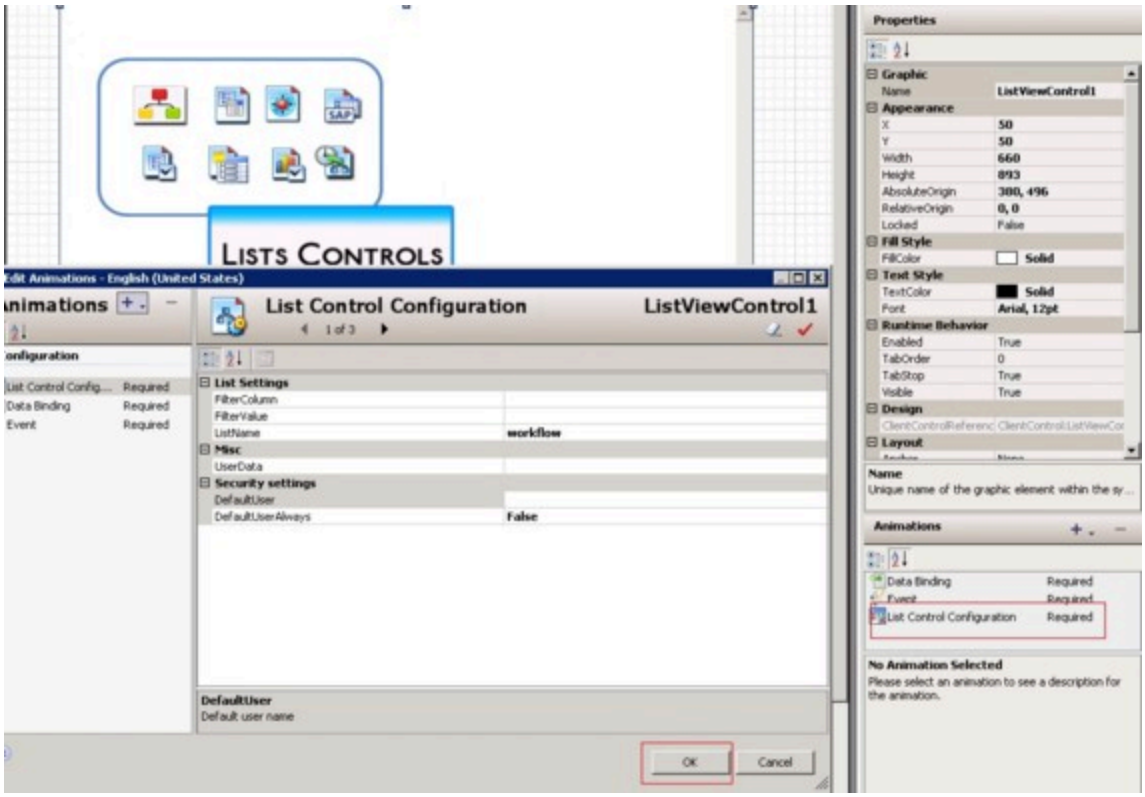
14. The BAM Control loads in the window.

Lists Inside InTouch

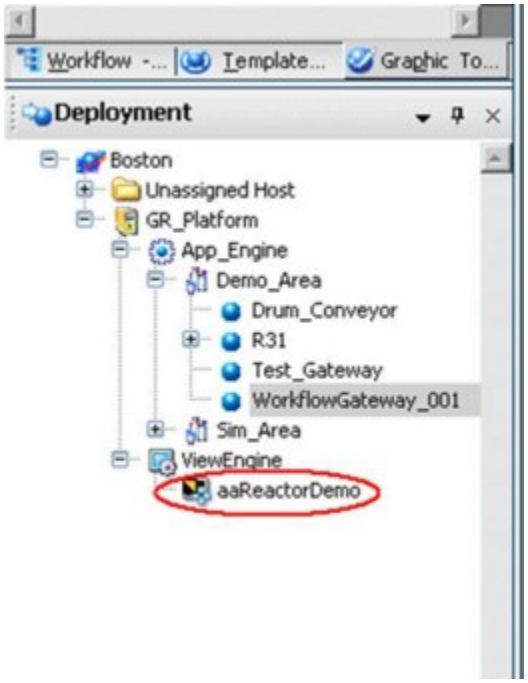
Follow Step 1 to Step 7 listed in the Initial Steps and continue:

1. Select the control in the editor and double click on Lists Configuration under Animations. This will open up a new configuration page from where the Lists control can be configured. After configuring the Lists, click OK. Save and close the Editor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



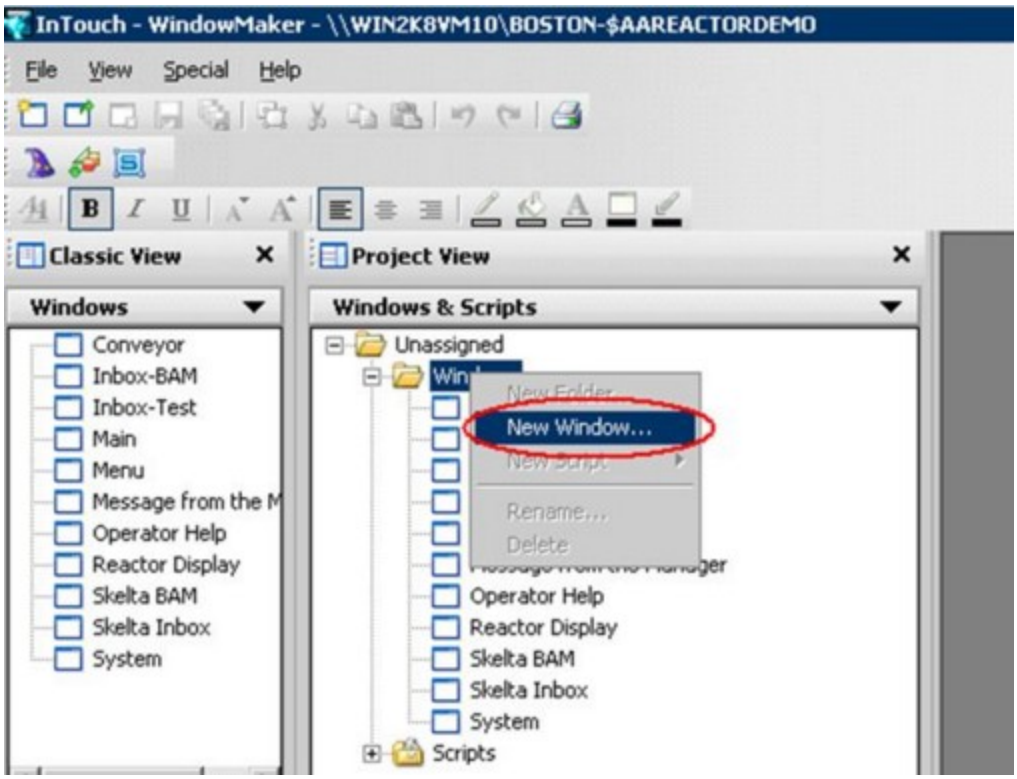
2. Select the instance and deploy in the System Platform IDE.
3. Double-click the $InTouchViewApp instance (aaReactorDemo) under ViewEngine from the System Platform IDE.



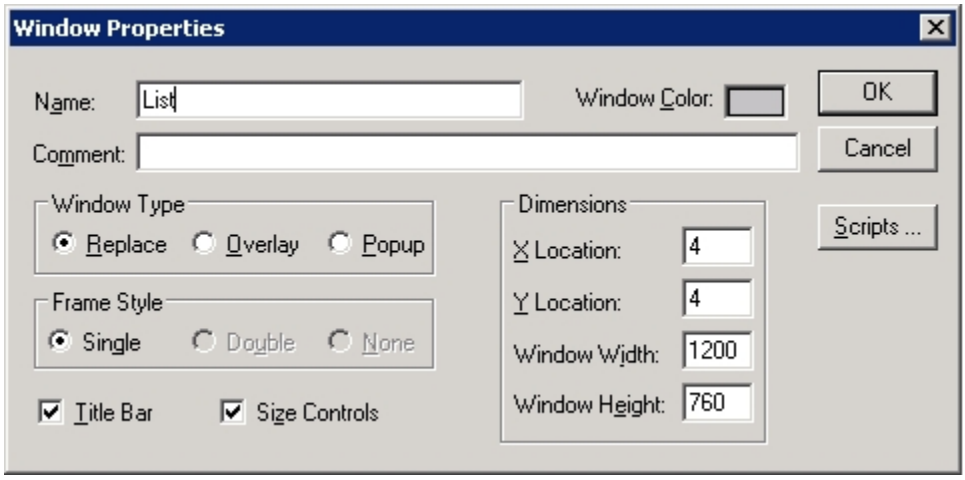
The InTouch WindowMaker appears.

4. Create a new window for the Lists control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

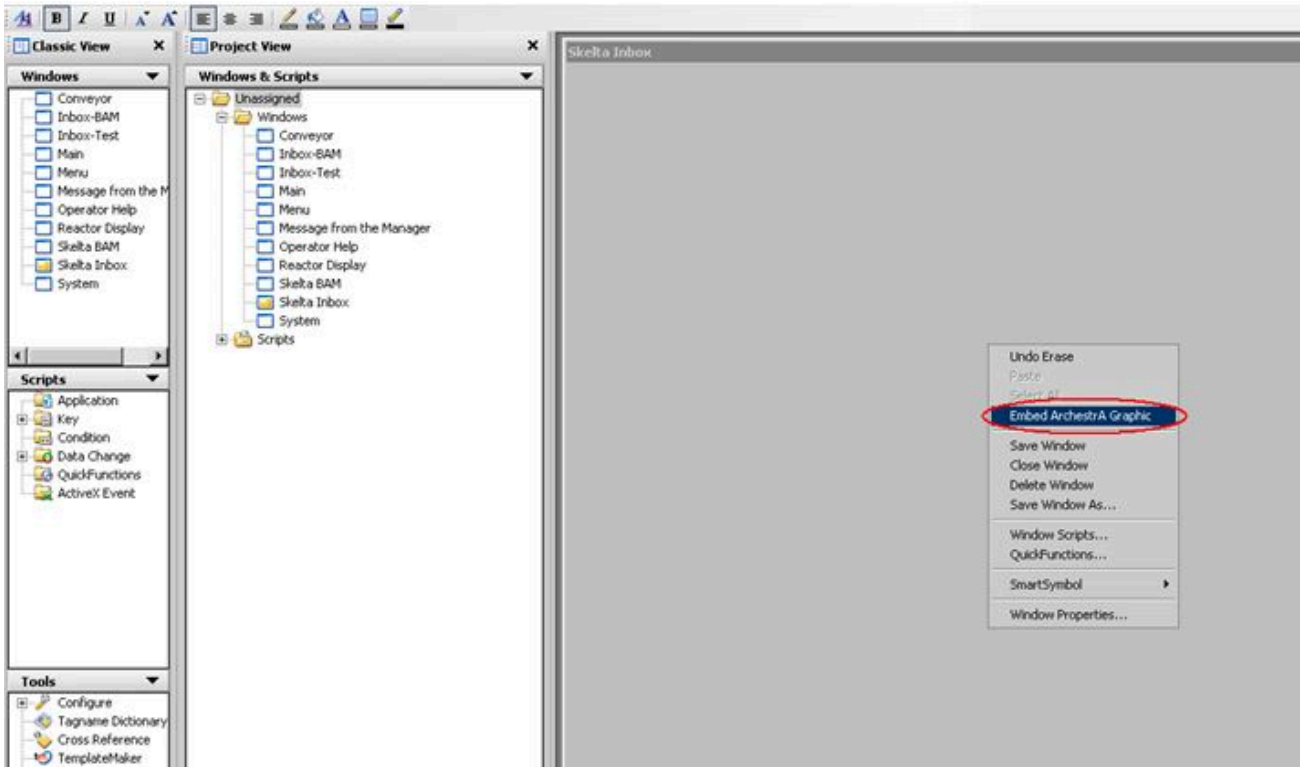


5. Provide the name and size for the window.



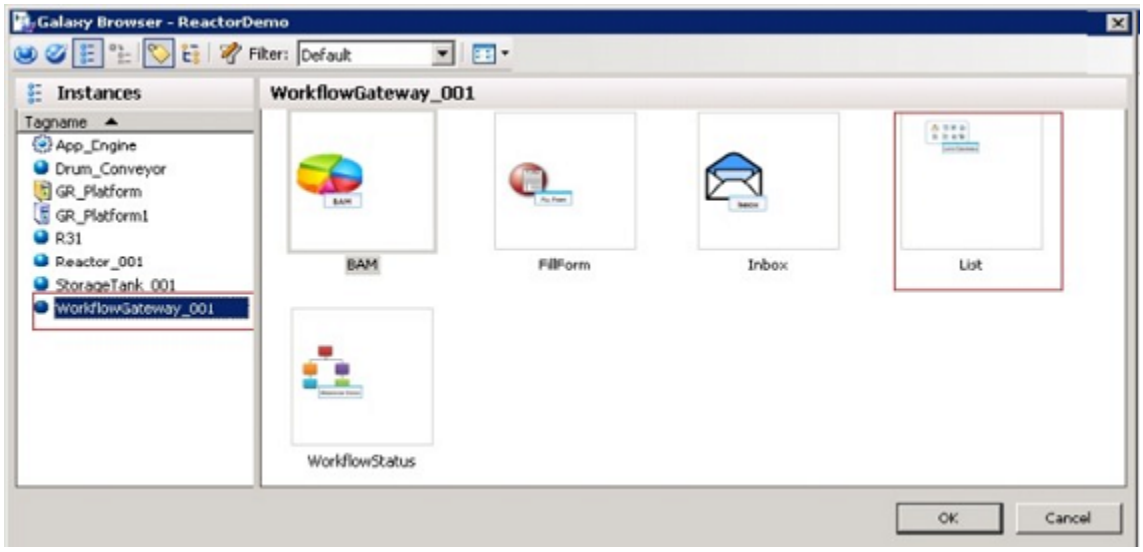
6. Right-click inside the newly opened window and select the Embed ArchestrA Graphic option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



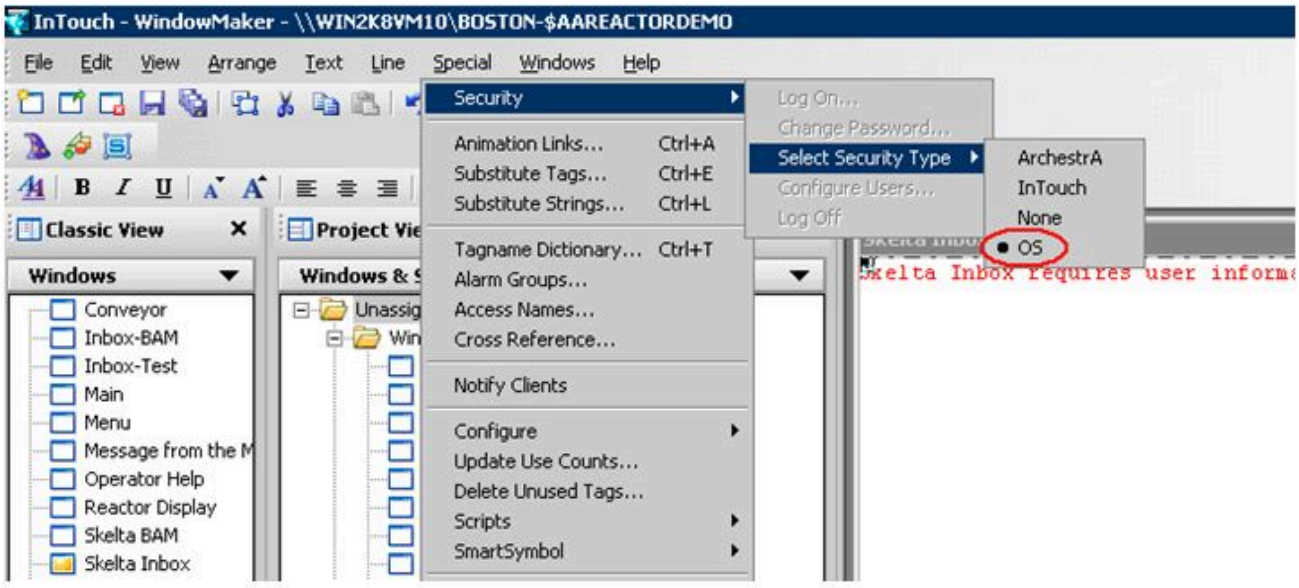
This opens up the ArchestrA Galaxy Browser.

7. Select the Instances tab and select the object to which the graphic symbol has been added.
8. Select the name of the graphic object (Lists) to be displayed.



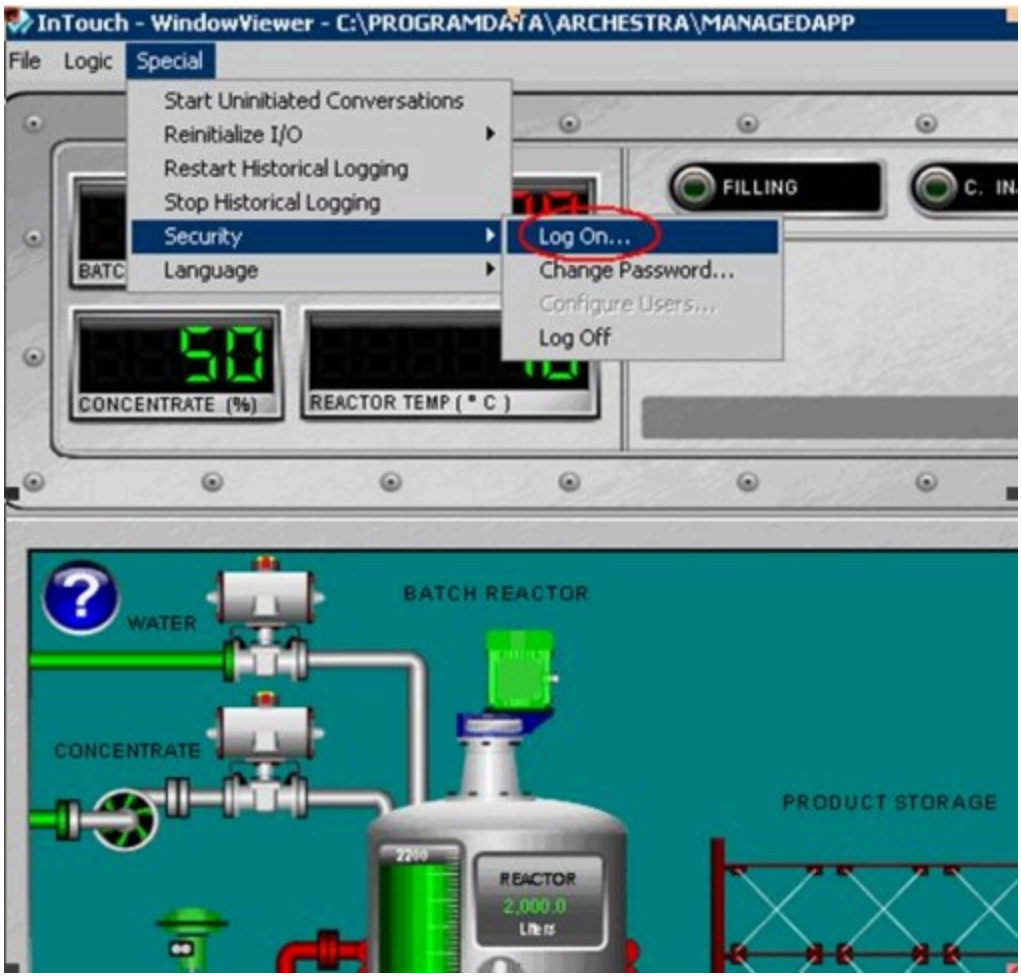
9. Enable security for the Lists window, if required.

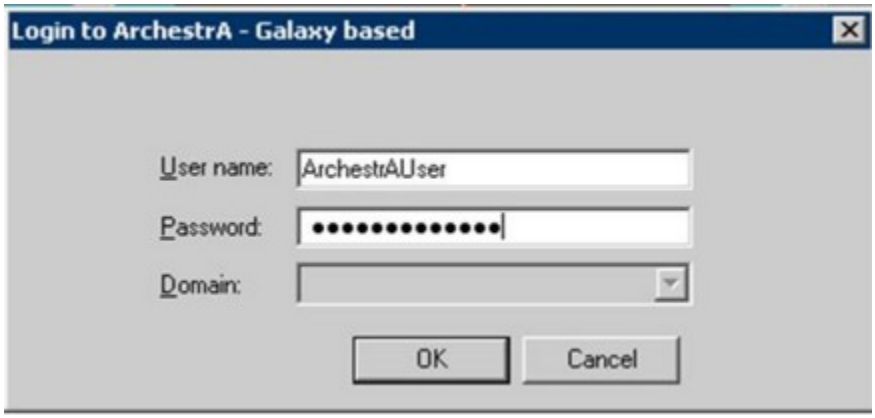
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



10. Save the window and exit InTouch WindowMaker.
11. Save and close the object in the System Platform IDE.
12. Deploy the $InTouchViewApp instance.
13. Open the InTouch Viewer and navigate to Special-> Security->Log On.
14. Enter the credentials and click OK.

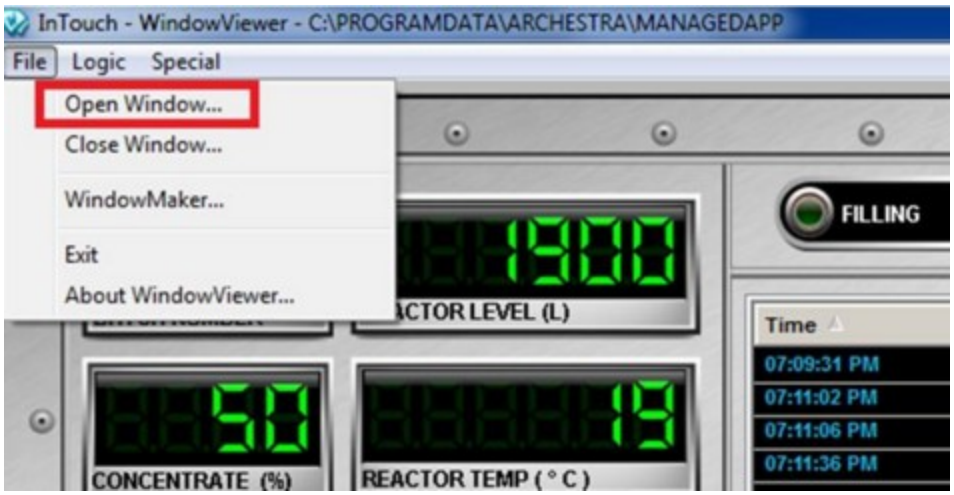
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

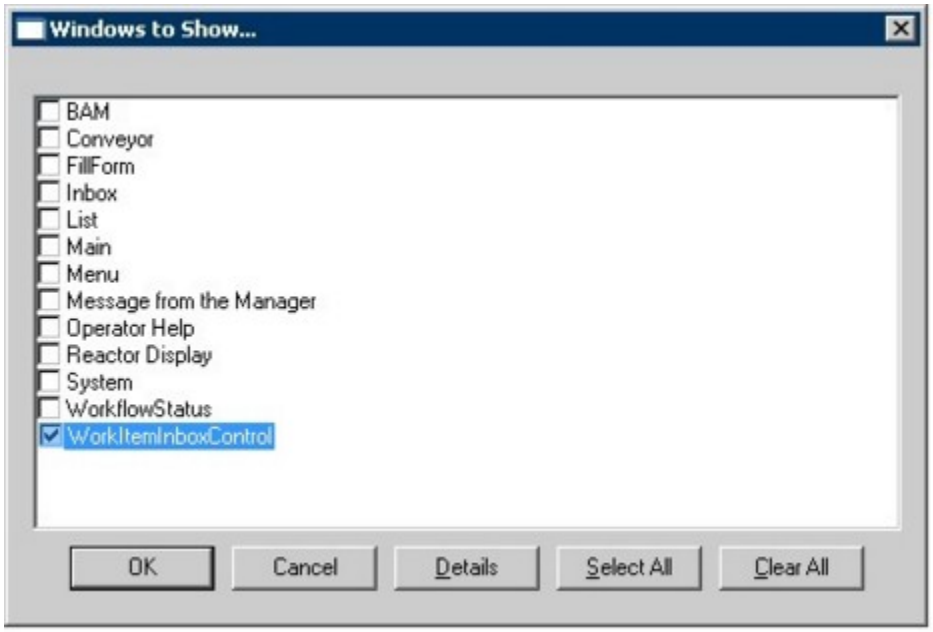




15. Open the created Lists window from the File menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.





16. The Lists Control loads in the window.

Forms Inside InTouch

Follow Step 1 to Step 7 listed in the Initial Steps and continue:

1. Select the control in the editor and double-click **Forms Configuration** under **Animations**.

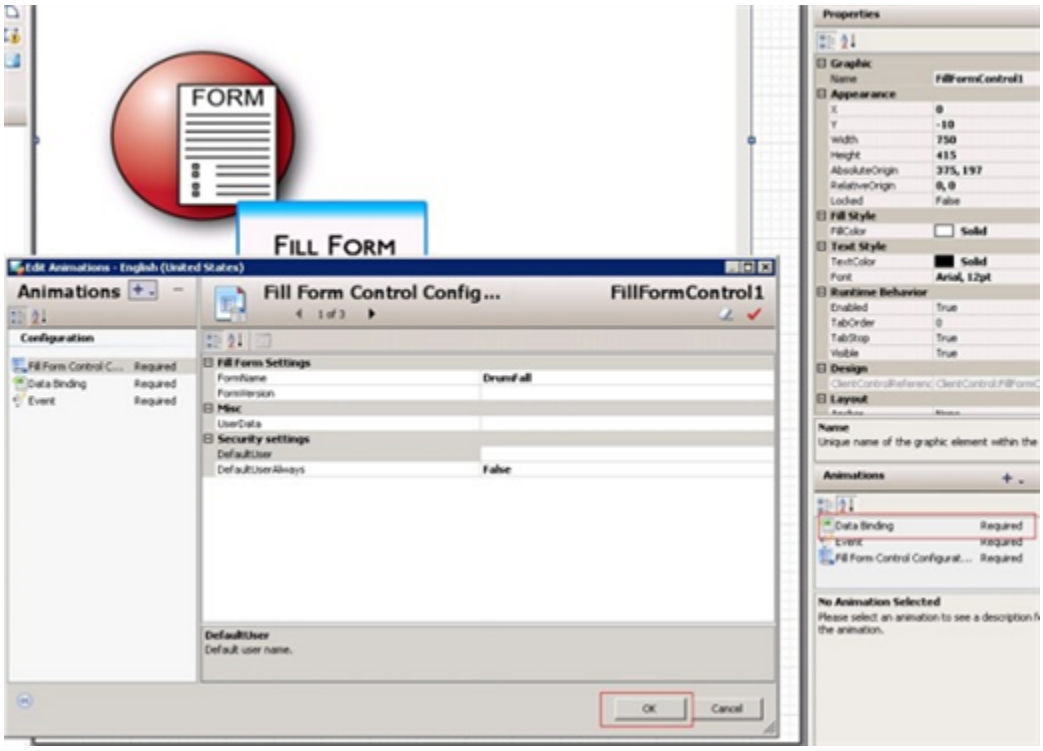
   This will open up a new configuration page from where the Forms control can be configured.
2. Under **Fill Form Settings**, perform the following:
   a. In the **Form Name** field, enter the form name.
   b. In the **Form Version** field, enter the version of the form, which must be considered by the Fill Form.

      For the Fill Form control to consider the latest published version of the form, enter the value as **0** or leave this field blank.

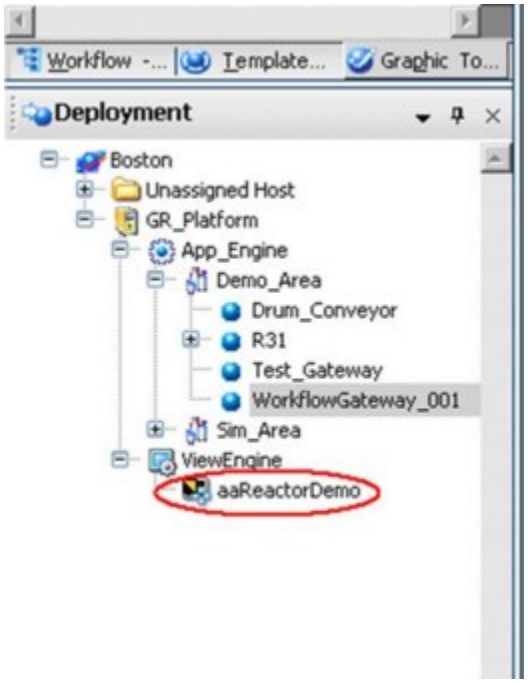
      If you enter an invalid version, then the "The given version does not exist for the form" message appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. After configuring the Forms, click **OK**.
4. Save and close the Editor.



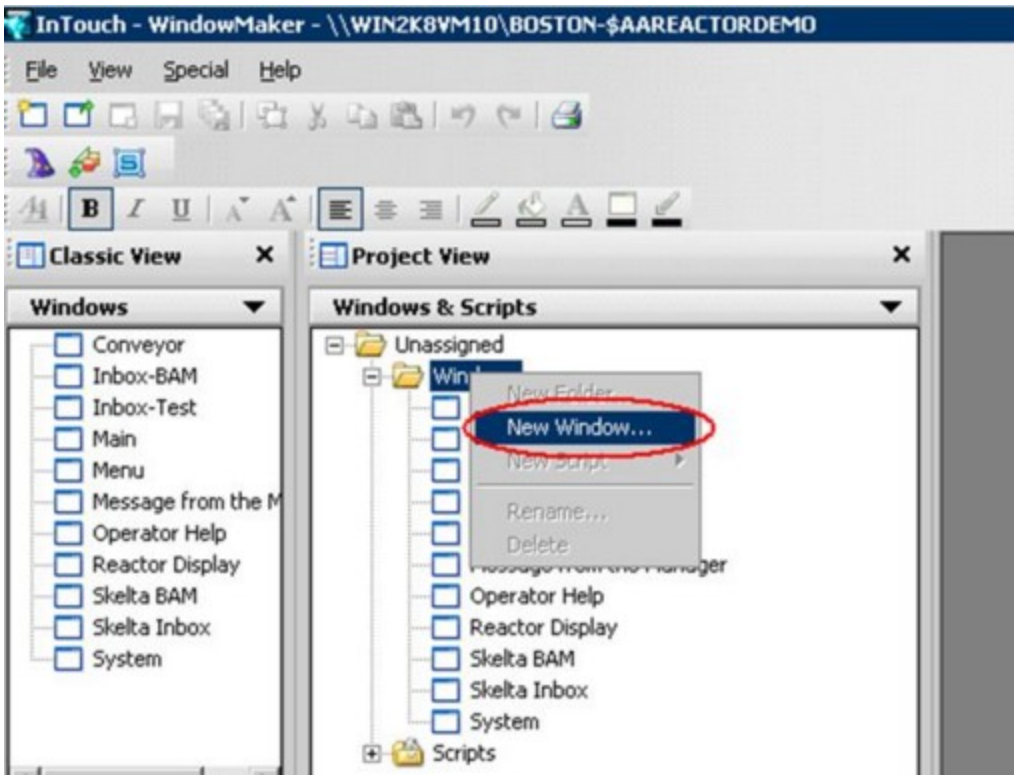
5. Select the instance and deploy in the System Platform IDE.
6. Double-click the $InTouchViewApp instance (aaReactorDemo) under ViewEngine from the System Platform IDE.



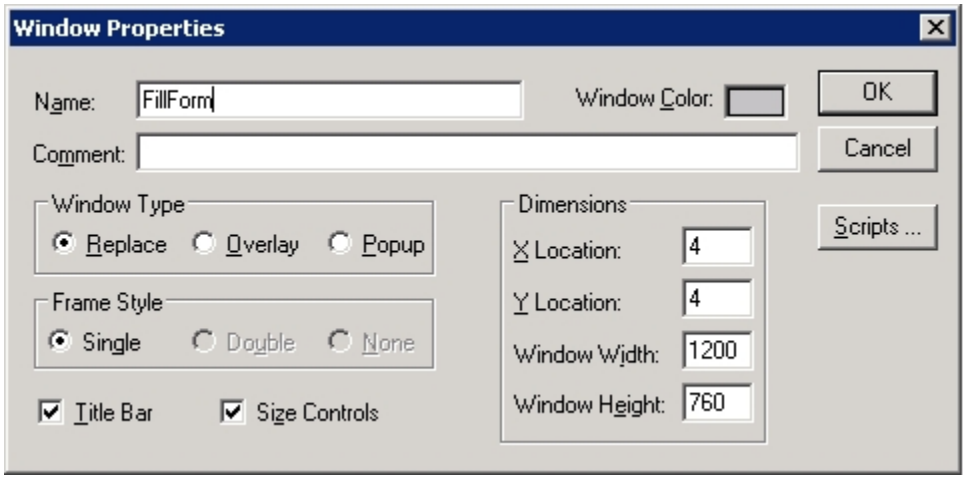
The InTouch WindowMaker appears.

7. Create a new window for the Forms control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

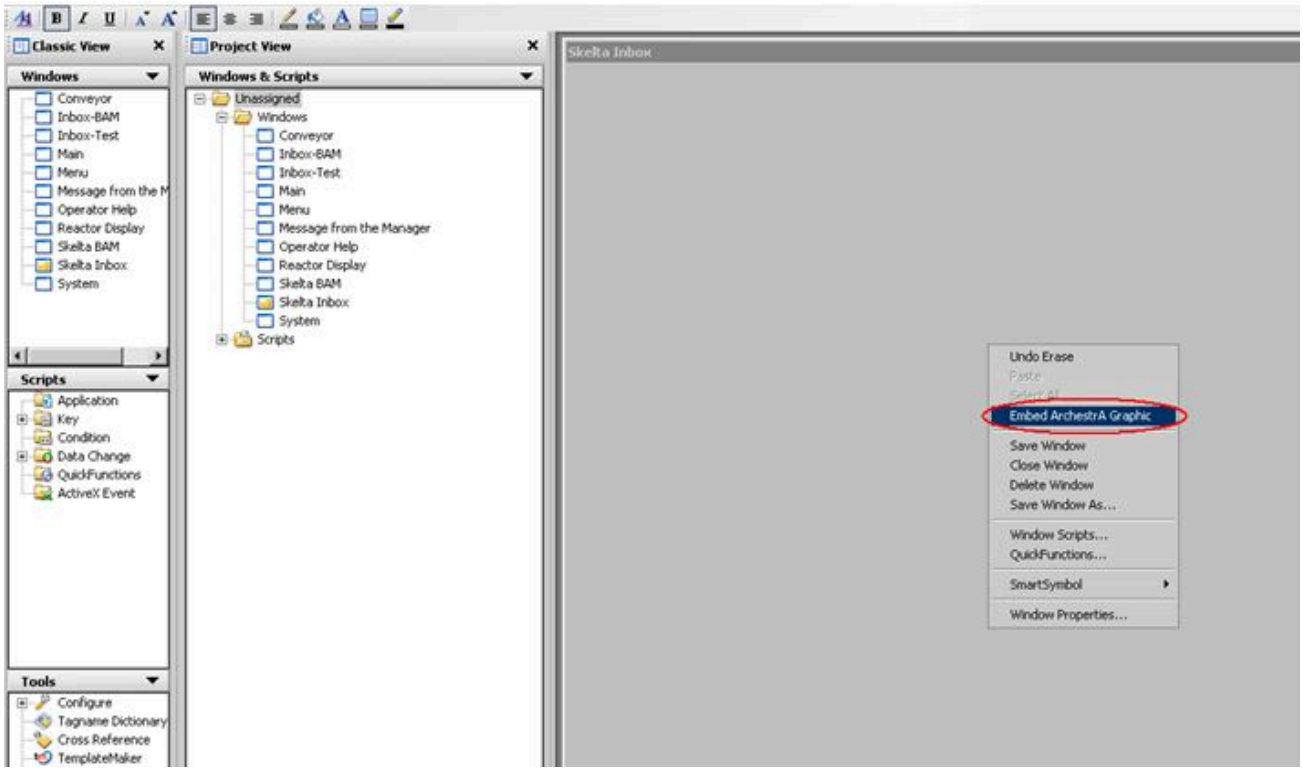


8. Provide the name and size for the window.



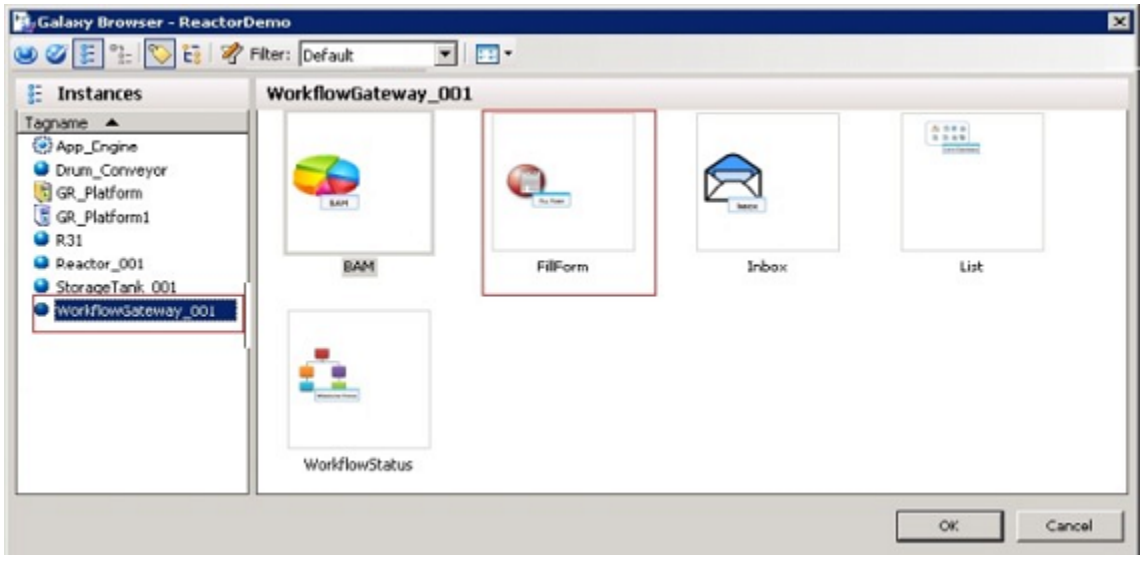
9. Right-click inside the newly opened window and select the Embed ArchestrA Graphic option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



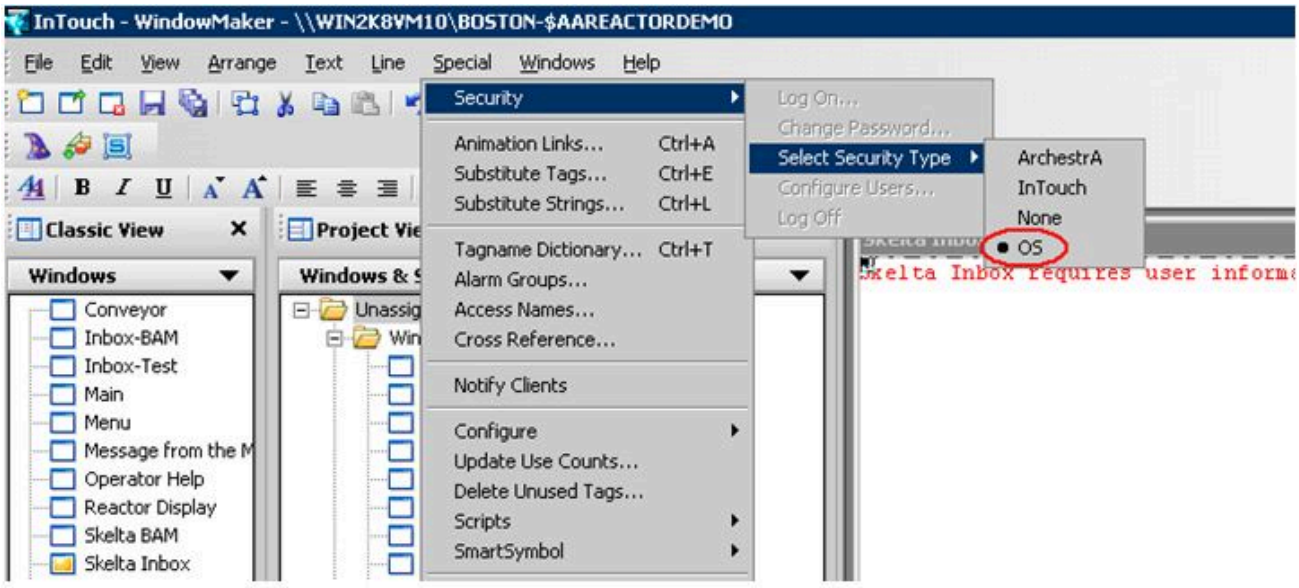
This opens up the ArchestrA Galaxy Browser.

10. Select the Instances tab and select the object to which the graphic symbol has been added.
11. Select the name of the graphic object (Forms) that has to be displayed.



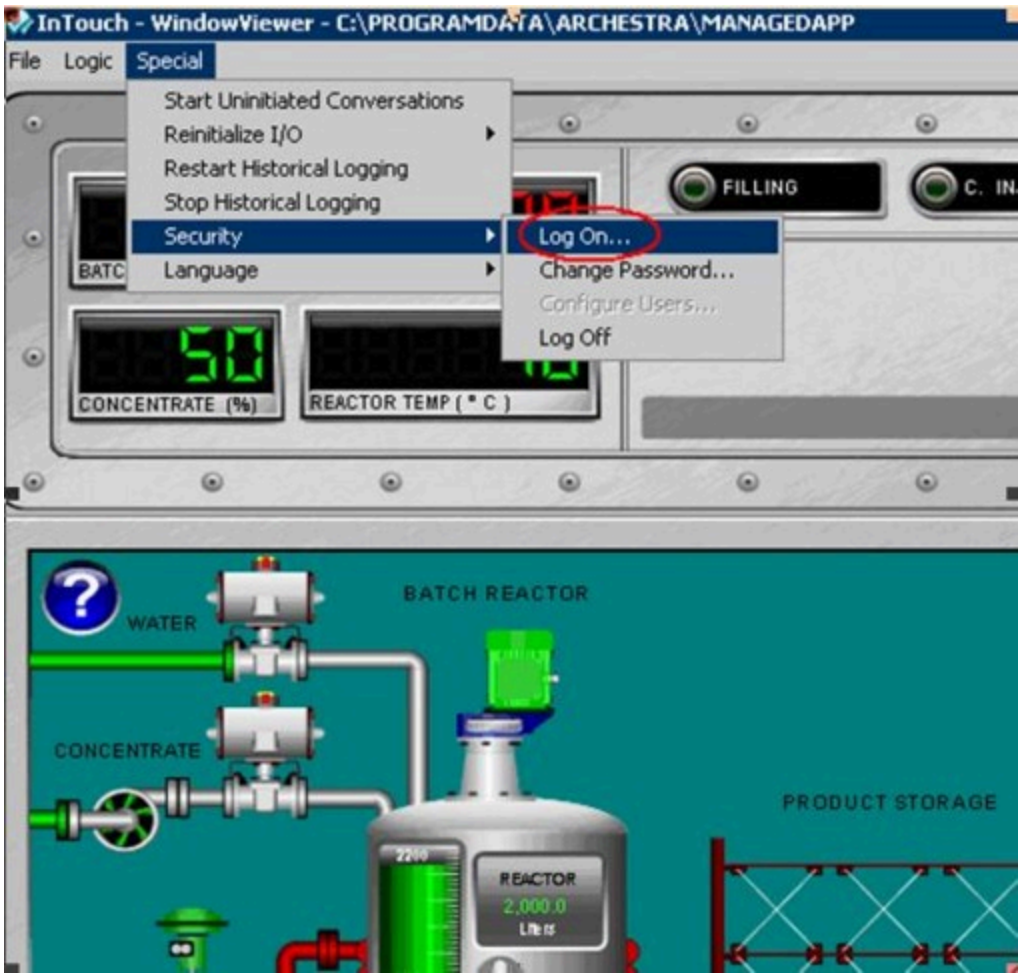
12. Enable security for the Forms window, if required.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

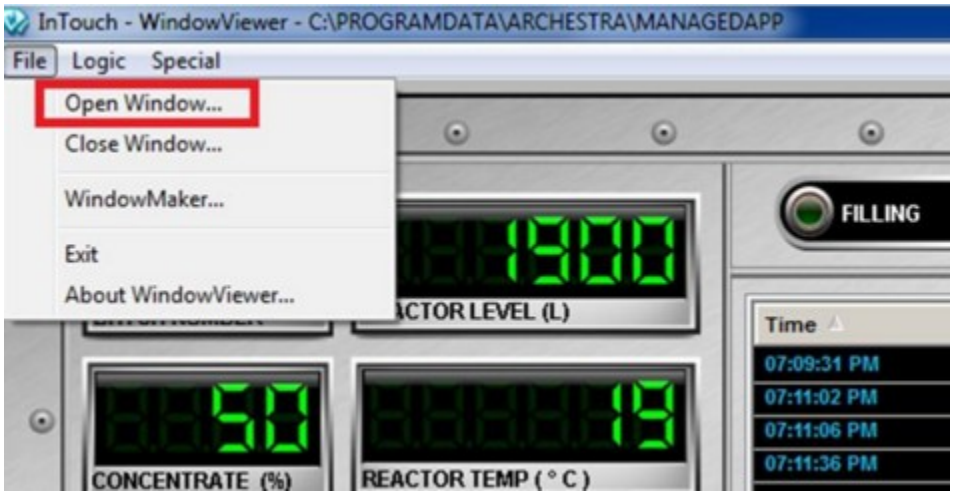


13. Save the window and exit InTouch WindowMaker.
14. Save and close the object in the System Platform IDE.
15. Deploy the $InTouchViewApp instance.
16. Open the InTouch Viewer and navigate to Special-> Security->Log On.
17. Enter the credentials and click OK.

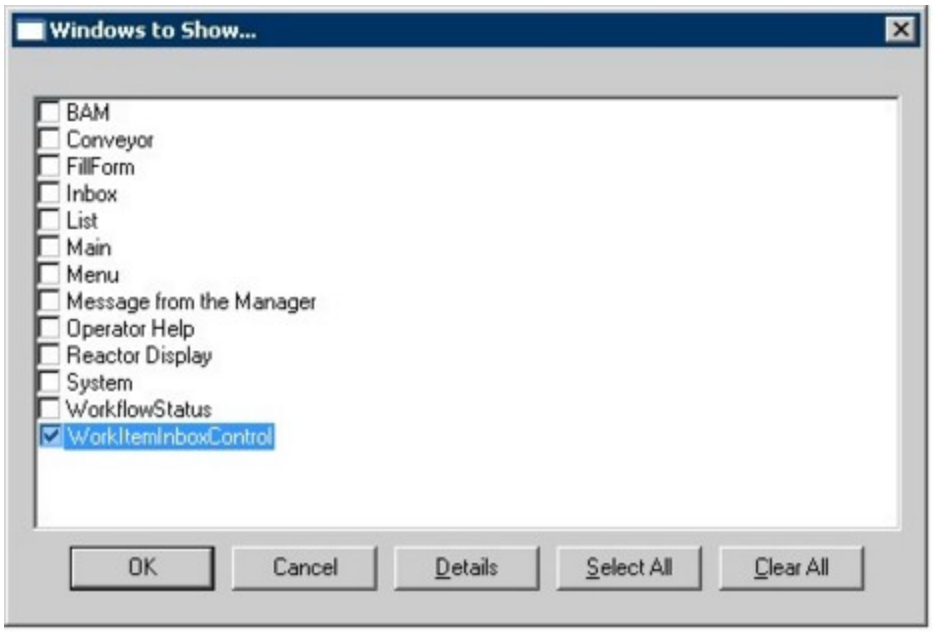
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



18. Open the created Forms window from the File menu.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The Forms Control loads in the window.

Note: InTouch does not support the execution of **window.resizeTo(x, y)** JavaScript function.

Pass Contextual Information in Forms from InTouch

You can pass real-time values from InTouch symbols to the InTouch Fill Form symbol.

Passing Contextual Information in Forms from **InTouch**, through attributes, is supported only for **String** data type.

Requirement

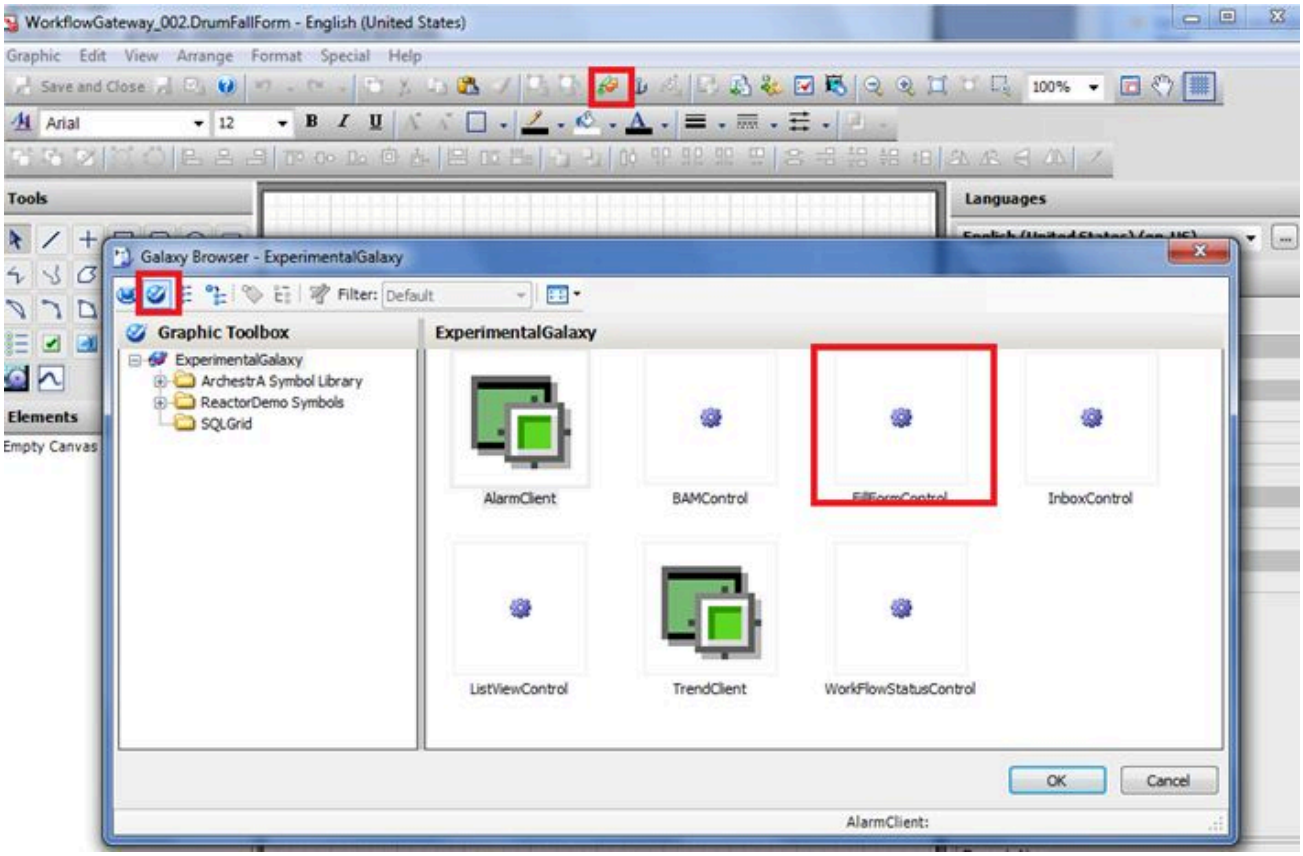
From InTouch Windows viewer, the run time values of InTouch symbols should be able to populate in a AVEVA Work Tasks form symbol at a predefined event.

Procedure

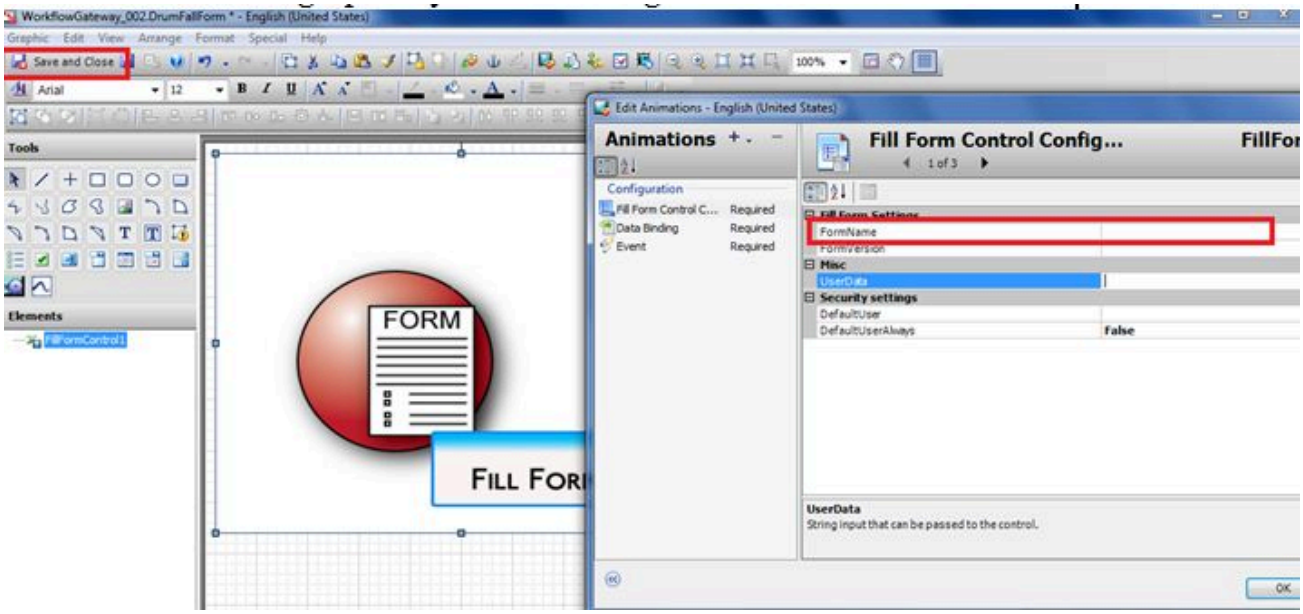
The steps to achieve this requirement will be explained with the Reactor Demo example that can be created by default while creating Galaxy.

1. Create the form to be filled on the Intouch event in the repository that is connected to the WorkflowGateway object. In this example, a "DrumFall" form is created with two text inputs that will display the Drum count and the Conveyor speed.
2. From the System Platform IDE, create a UDA of the type string and user writable to the WorkflowGateway object.
3. From the System Platform IDE, open the WorkflowGateway object instance and select the graphics tab. Create a new graphic object named DrumFallForm.
4. Double-click the symbol name and open the ArchestrA graphic symbol editor.
5. On the top of the graphic editor, click on the Embed Graphic icon which will pop up the Galaxy Browser.
6. Select the Graphic Tool box icon on the top left corner of the Galaxy Browser.
7. Select the FillFormControl symbol and click OK.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



8. Click on the graphic editing area which will paste the FillForm symbol in the editor.
9. Double-click on the graphic symbol to configure. Provide the form name to be filled.



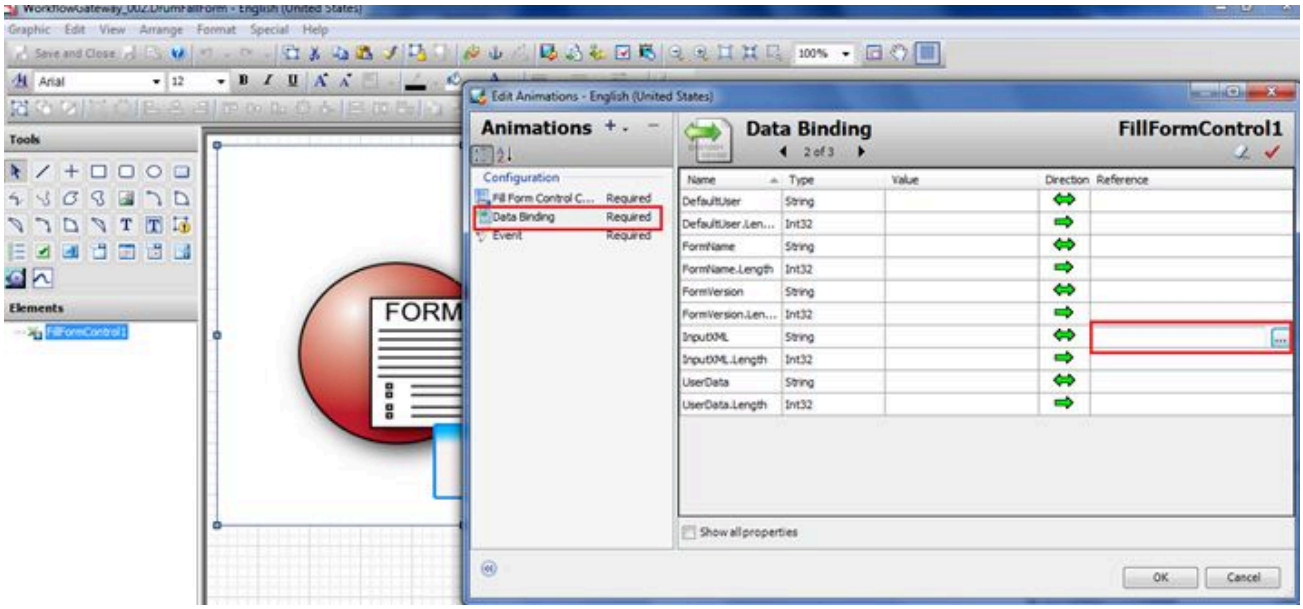
10. Click on the DataBinding tab of the Animations window. For the "InputXML" field, click on the button in the "Reference column".

    This opens the Galaxy Browser.

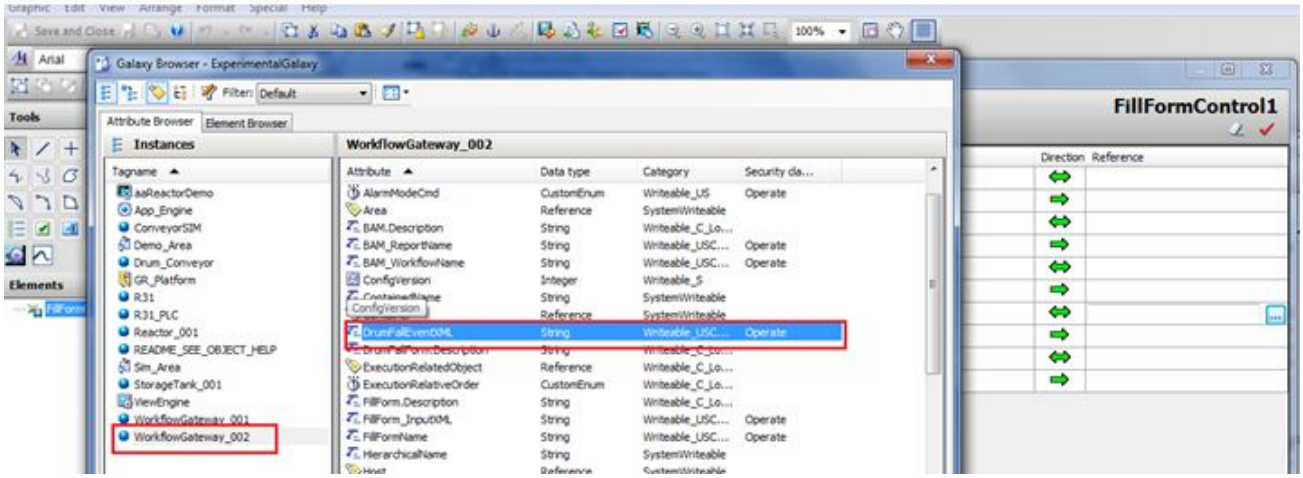
    Note: The value from the Value script takes precedence over the InputXML (that is the value for the controls)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

and the Default Value, and the InputXML takes precedence over the Default Value for the controls on the form.



11. From the Galaxy Browser, select the object on which the UDA "DrumFallEventXML" was created and press OK.



12. Click OK and then click on Save and Close on the Graphic Editor.

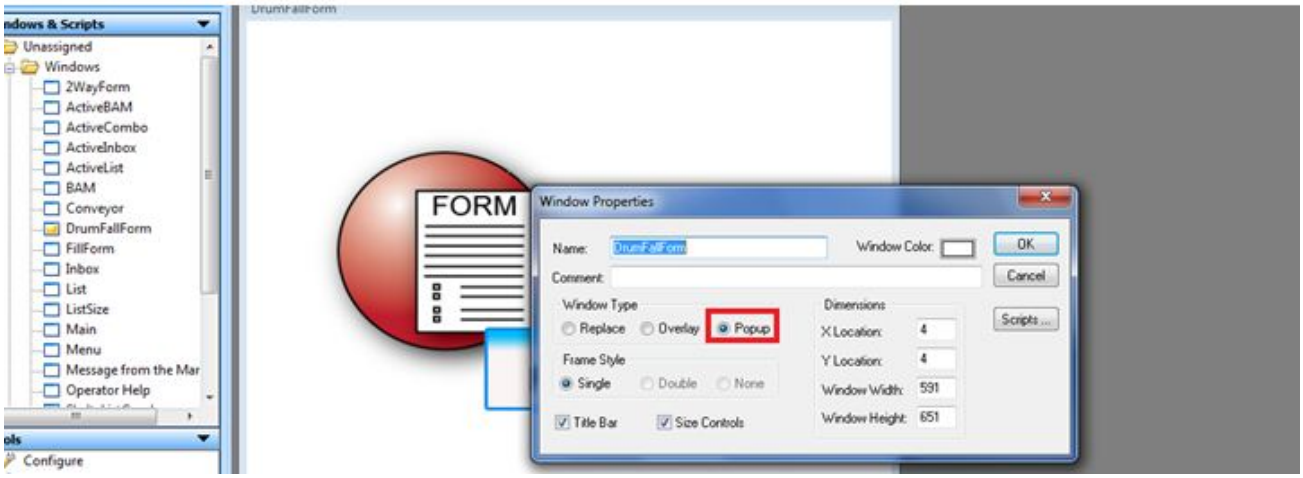
    This will successfully create the FillForm graphic control.

    Deploy the WorkflowGateway object.

13. Create a window in the InTouch window maker. Double-click on the ViewEngine object (aaReactorDemo) to open the windows maker.

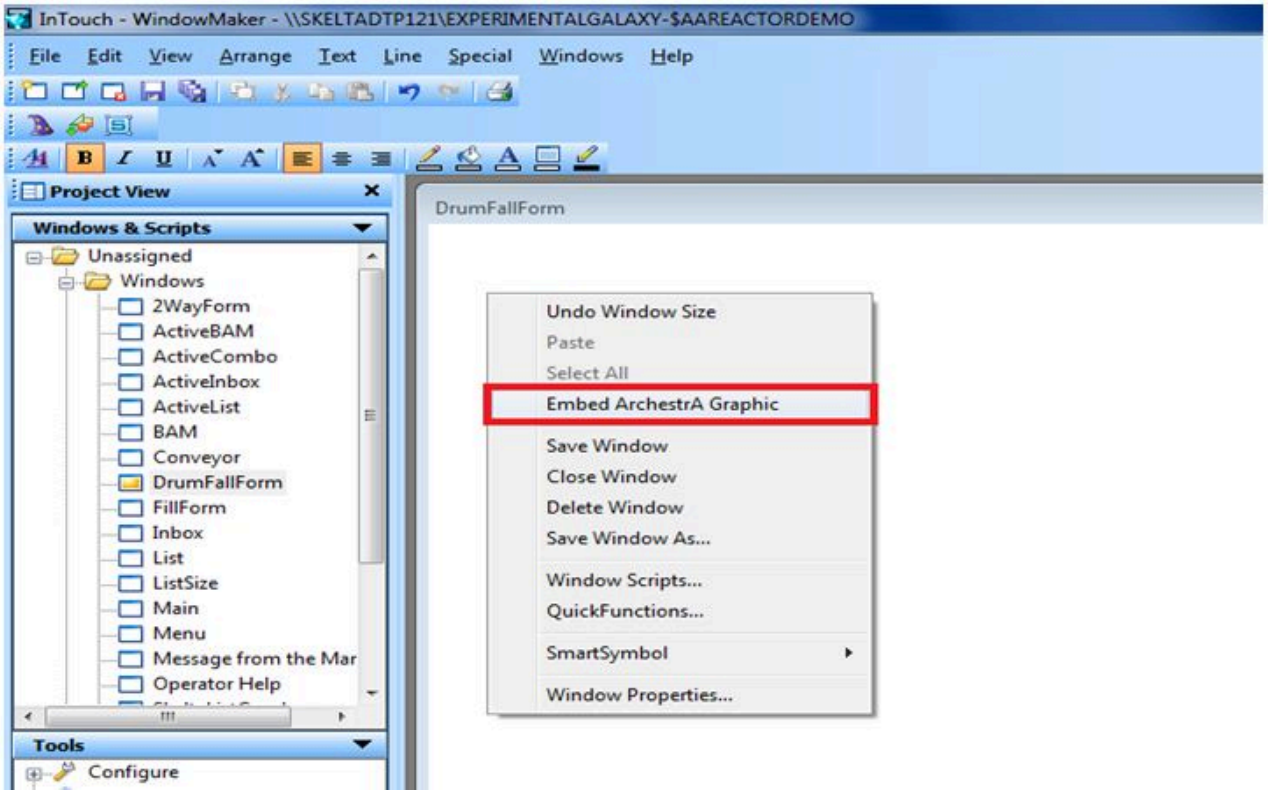
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

14. Create a new window from the windows maker by selecting File> New window, and name it "DrumFallForm". Set the Window type as "Popup".

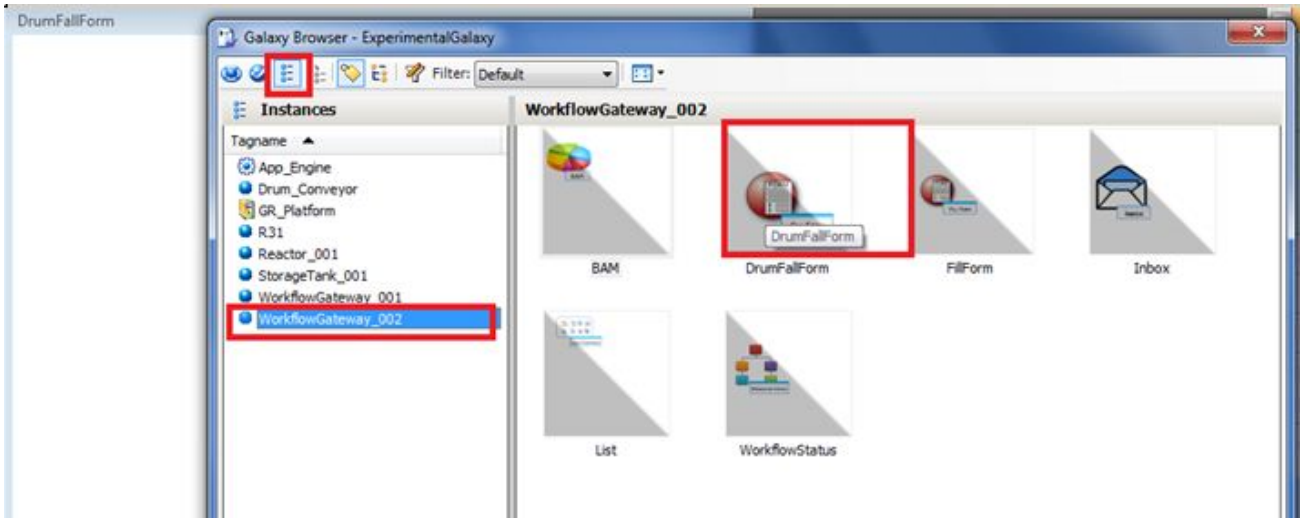


15. Right-click inside the window area and select "Embed ArchestrA Graphic". This opens the Galaxy Browser.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

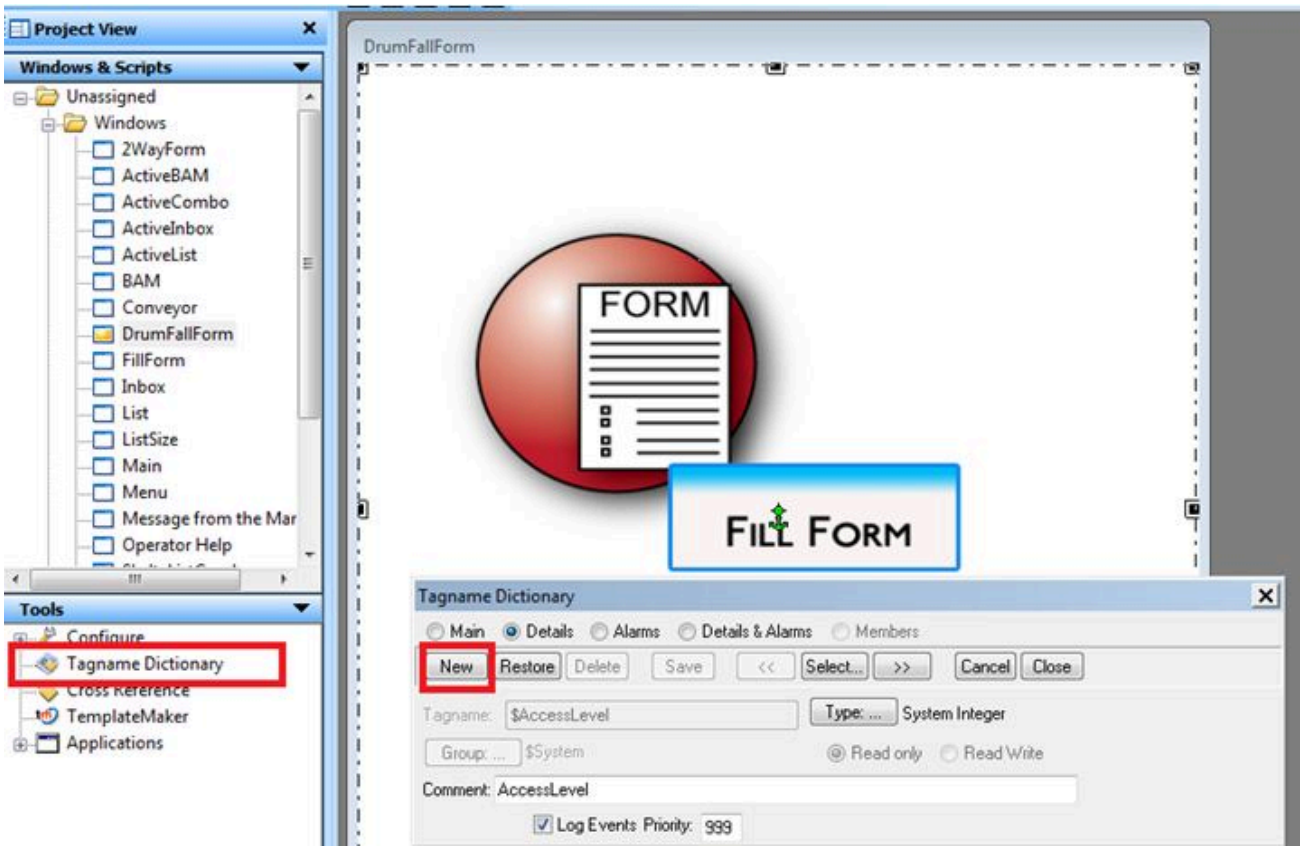


16. From the Galaxy Browser, select the Object instances icon and select the instance to which the Fill Form control has been added. Double-click the "DrumFallForm" symbol. Click on the window area which will paste the symbol to the window.



17. Next, create tags. From the Tools window, double-click to open the "Tagname Dictionary" and click on New.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

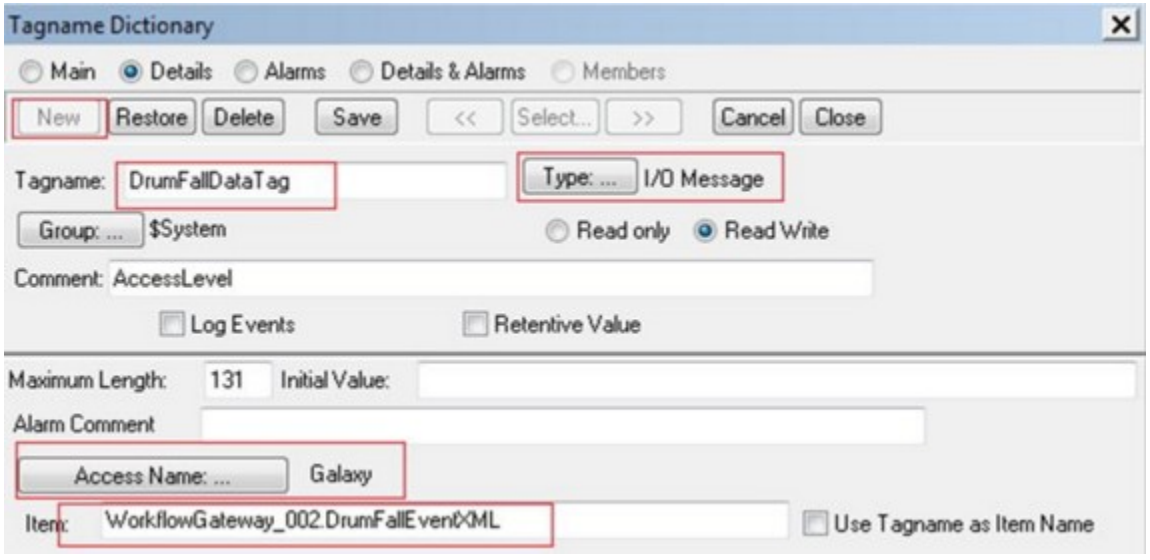


18. Type a name for the tag (DrumFallDataTag) and select the "type" as I/O "Message".

    Set the Access Name property as "Galaxy".

    In the Item property, give the name "Objectname.UDA" to which the tag should write values. (WorkflowGateway_002.DrumFallEventXML)

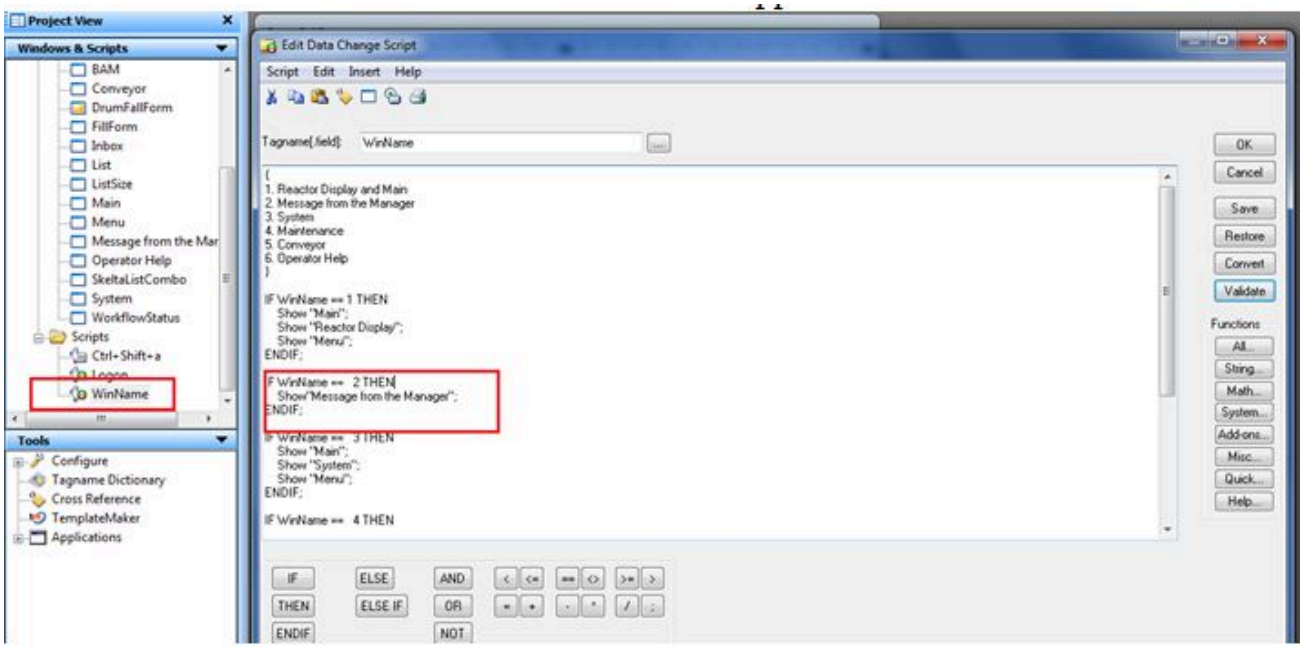
19. Click Save and Close to save and close the Tagname Dictionary.



20. Write the required script.

    From the Windows and Scripts box, double-click to open the "WinName" script. The condition "WinName==2" is when the drum fall happens.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



In this condition, first assign the data to the DrumFallDataTag created. The data should be in XML format which should match the syntax of the instance XML of the "DrumFall" form, when previewed. As the maximum length of an attribute in System Platform cannot exceed 1023 characters, pass only the XML for the required controls. Add the required dynamic values from the Drum Conveyor in the XML as shown.

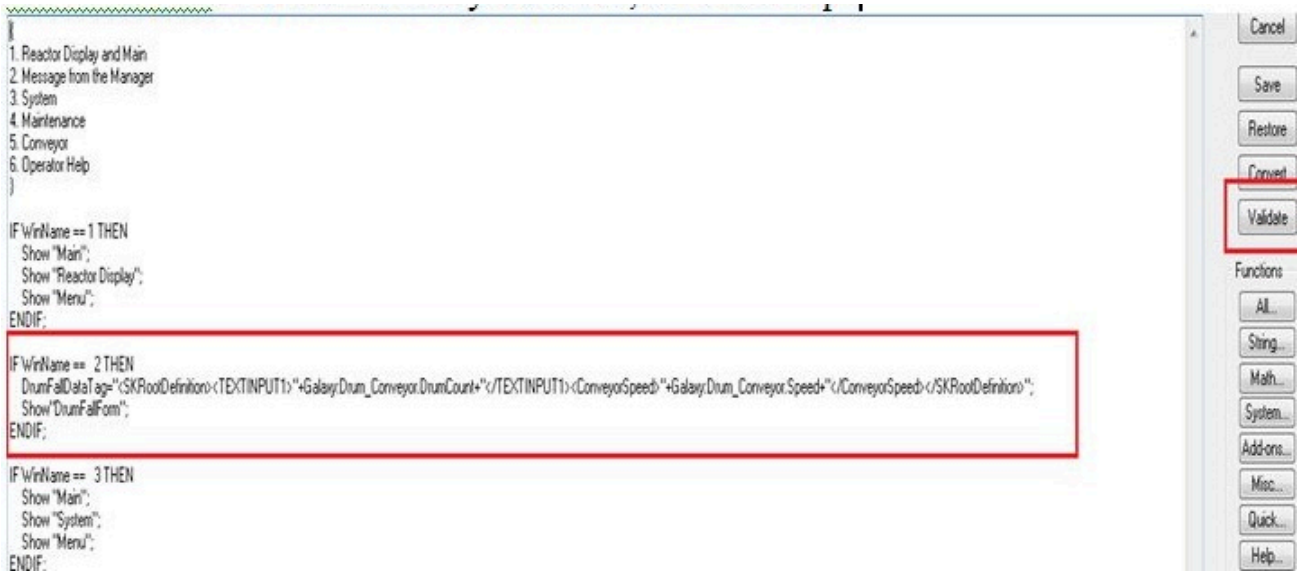
For example,

DrumFallDataTag="<SKRootDefinition><TEXTINPUT1>"+Galaxy:Drum_Conveyor.DrumCount+"</TEXTINPUT1><Conve

21. Write the script to open the DrumFallForm window.

    For example, Show"DrumFallForm";

22. Click Validate and if successfully validated, save the script.



23. Save all the open windows and exit the window maker.
24. Deploy the ViewEngine object (aaReactorDemo).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Run the conveyor example. The form pops up.

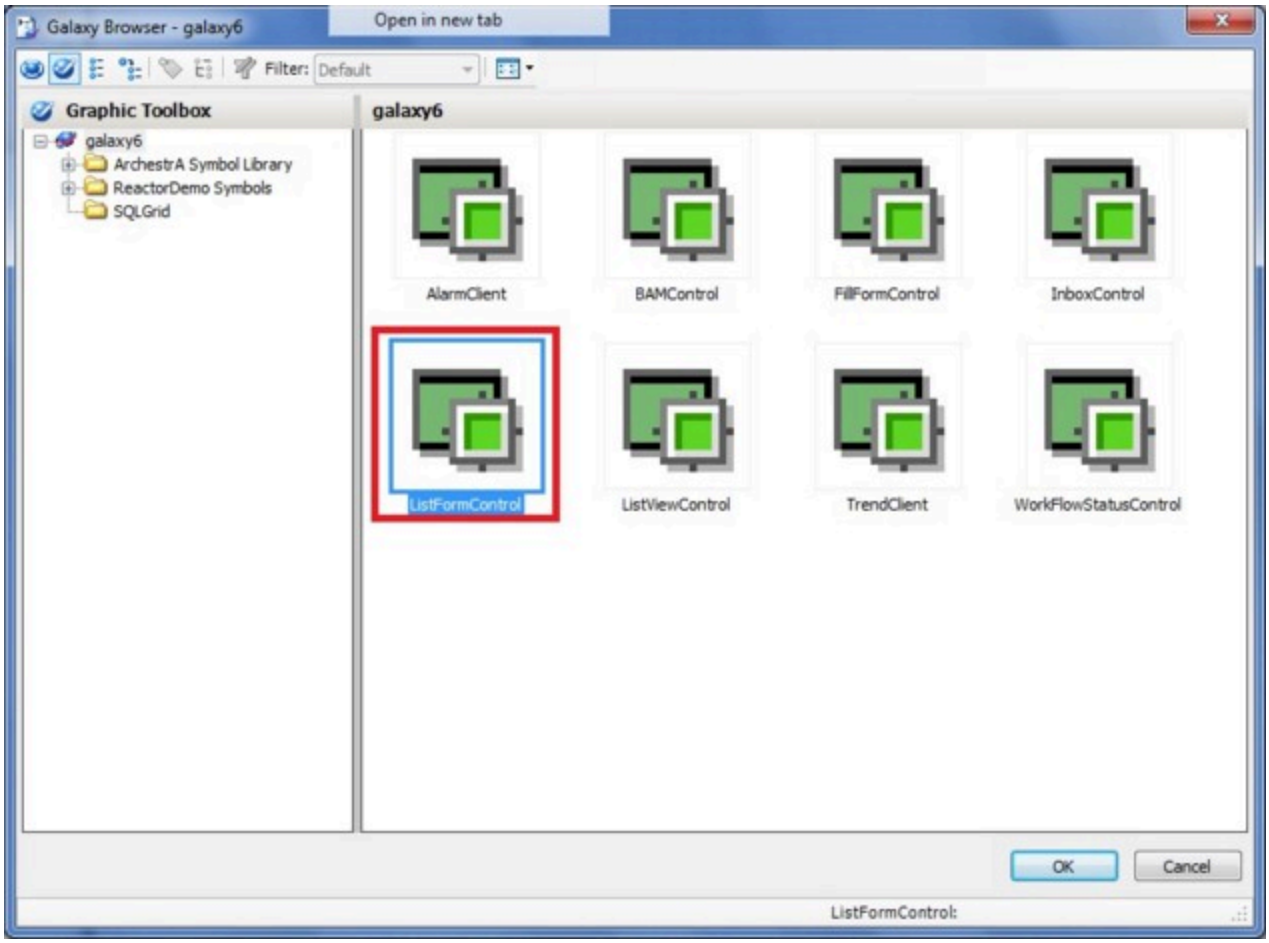
1. Open the InTouch viewer and login with the credentials.
2. Click on the Open window and select the Conveyor window.
3. Change the control from Auto to Manual, and manually transfer a drum and make it fall.
   The DrumFallForm will get displayed with the current drum count and the conveyor speed at the time of fall.

List Forms Inside InTouch

Follow Step 1 to Step 7 listed in the Initial Steps and continue:

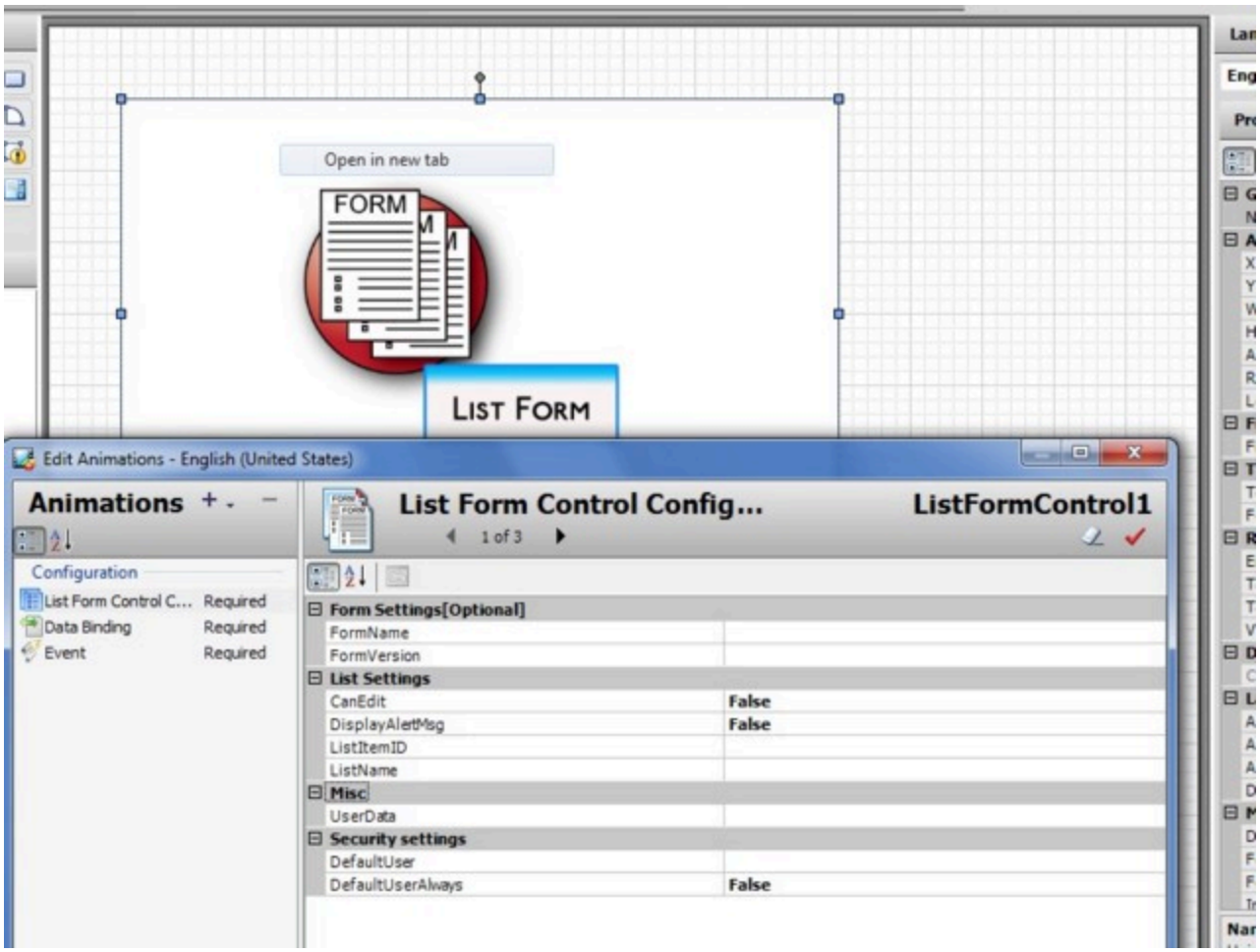
1. Select ListFormControl in the Galaxy Browser.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



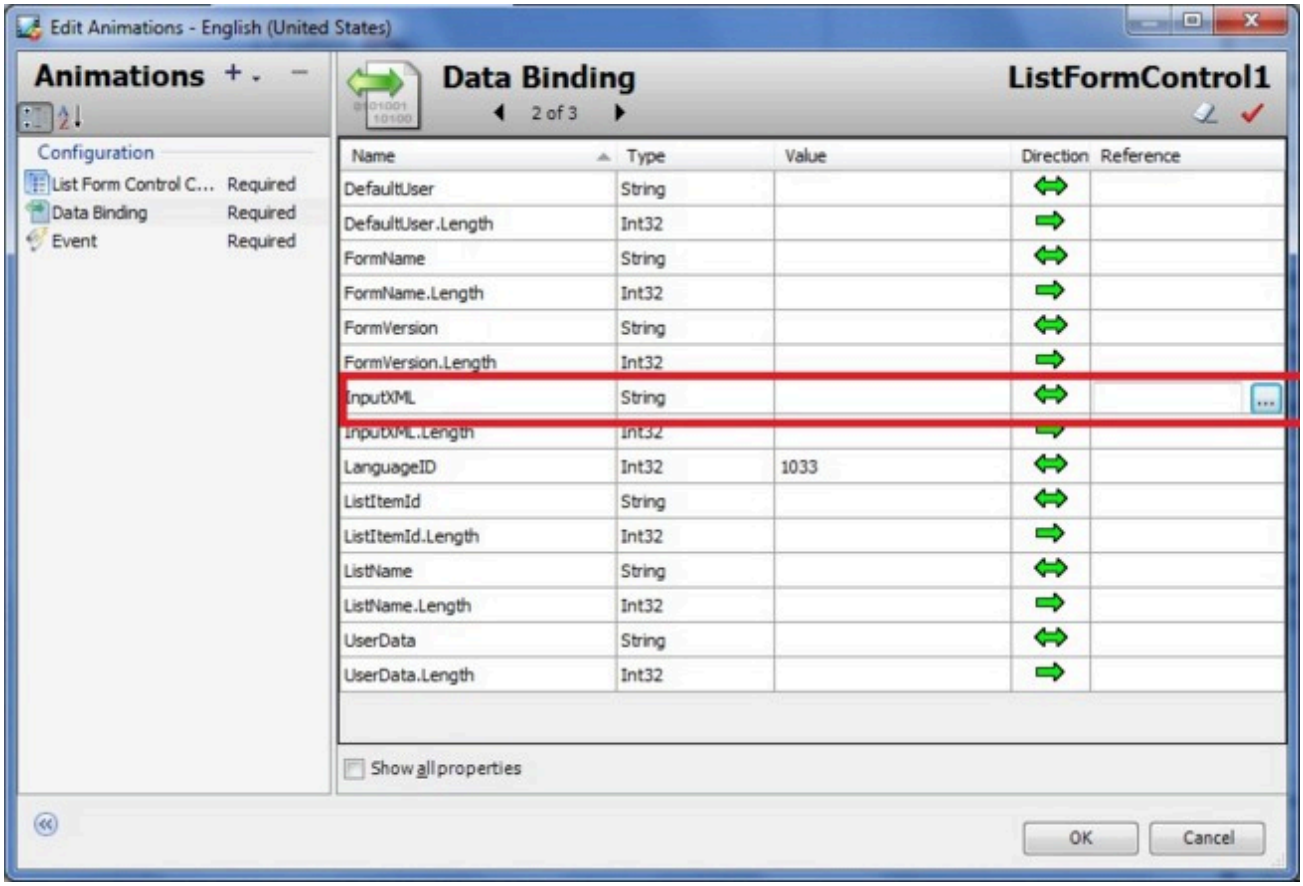
2. Animations. This will open up a new configuration page from where the List Form control can be configured. After configuring the List Form, click "OK".

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



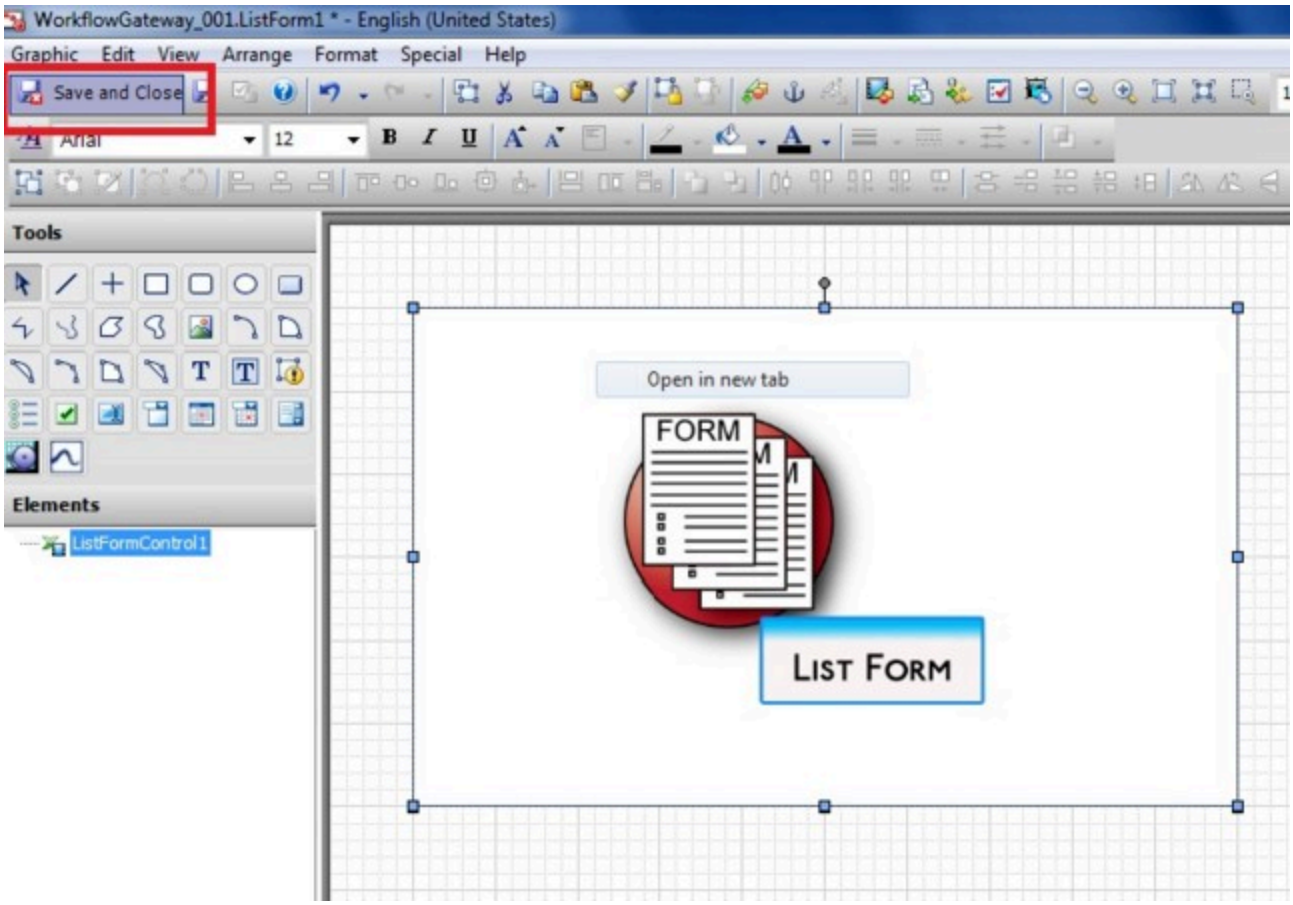
You can set the form instance XML for a List Form in Add Item mode dynamically. The InputXML property of the control should be data-binded to the UDA containing the form instance XML. For more information about the procedure, refer to Pass Contextual Information in Forms from InTouch.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

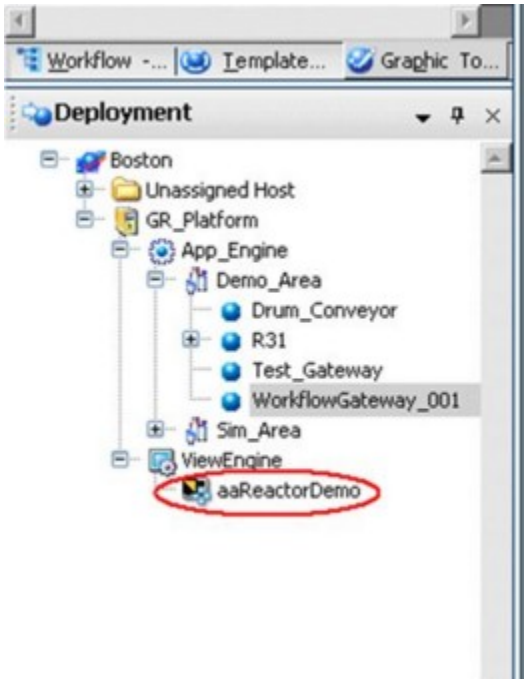


3. Save and close the Editor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



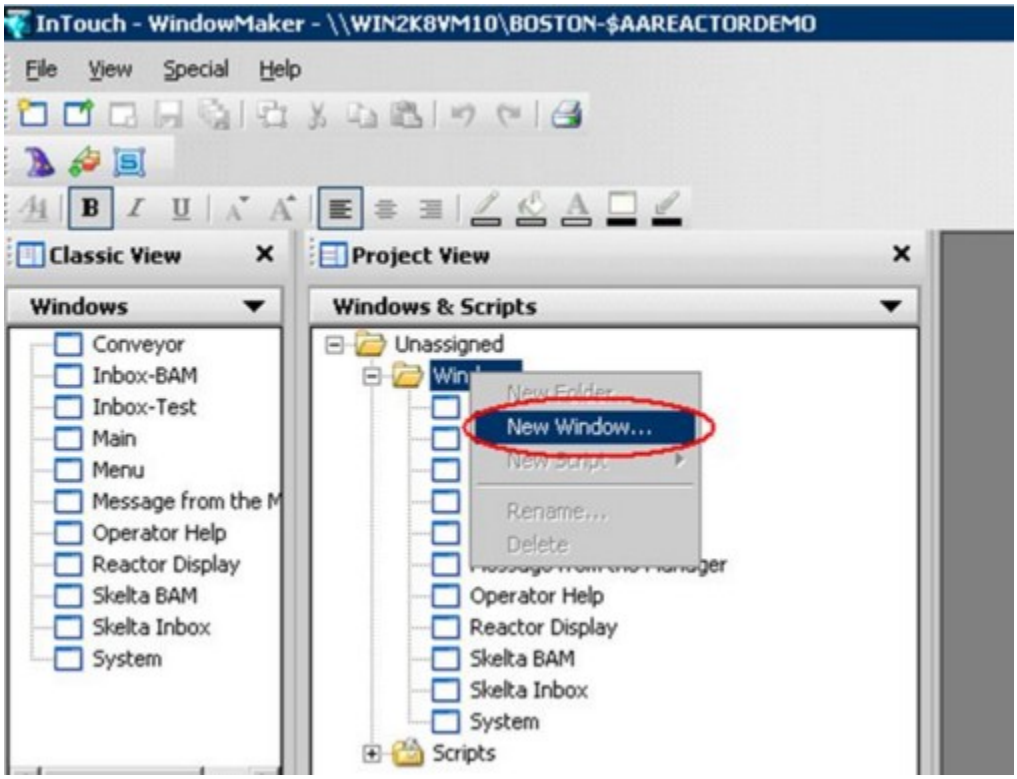
4. Select the instance and deploy in the System Platform IDE.
5. Double-click the $InTouchViewApp instance (aaReactorDemo) under ViewEngine from the System Platform IDE.



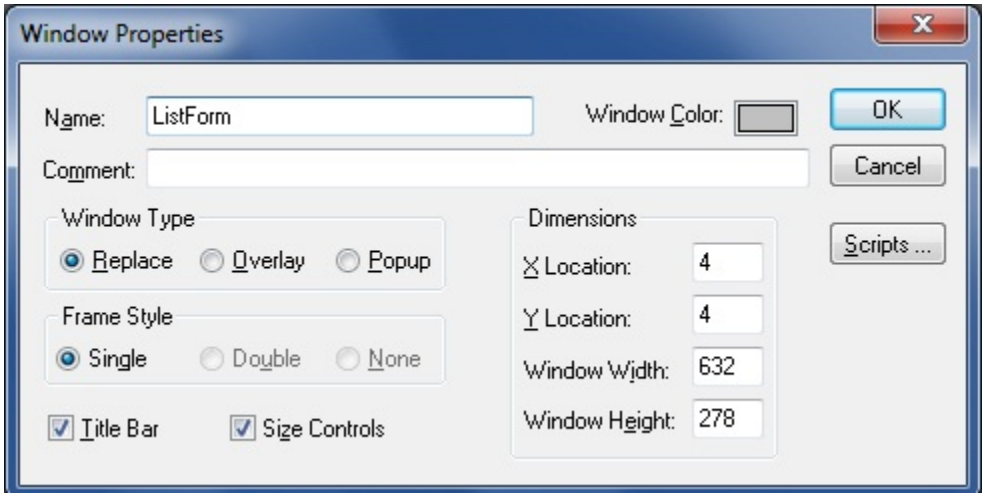
The InTouch WindowMaker appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Create a new window for the Lists control.

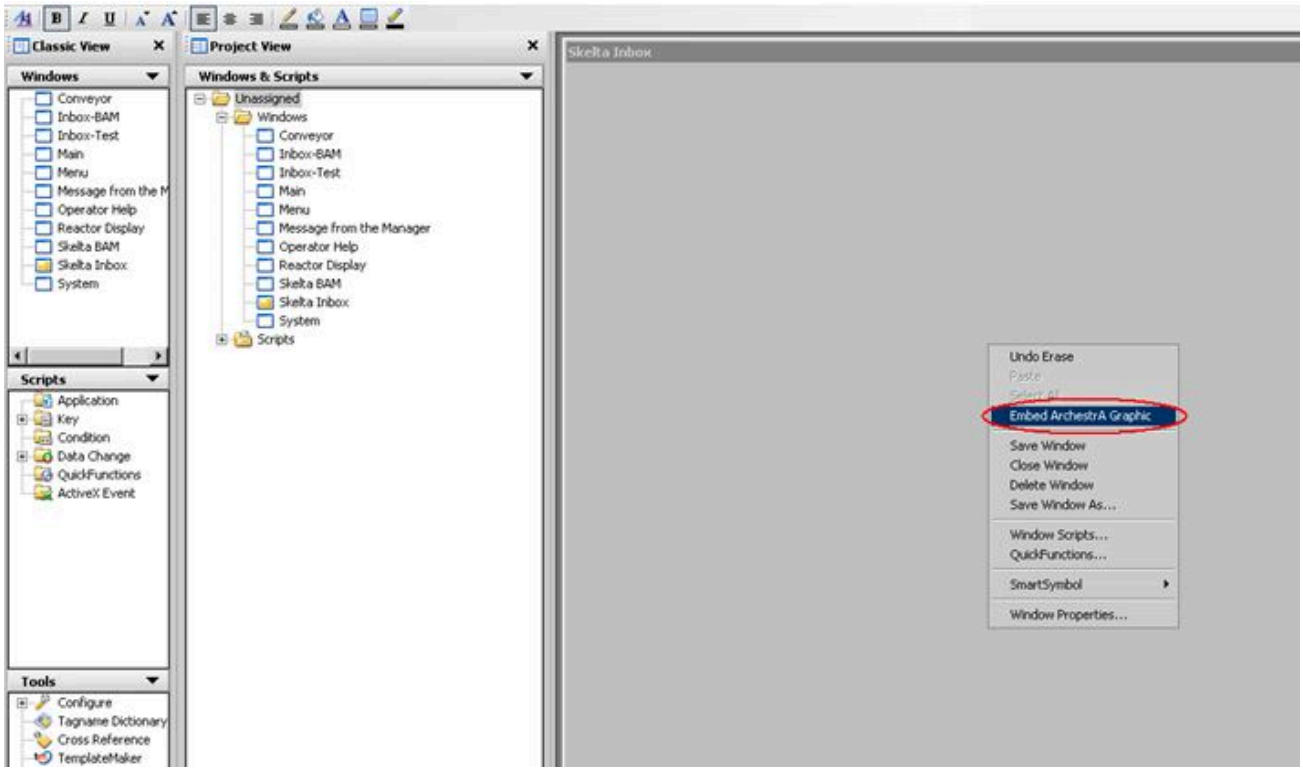


7. Enter the name as "ListForm" and click OK.



8. Right-click inside the newly opened window and select "Embed ArchestrA Graphic" option.

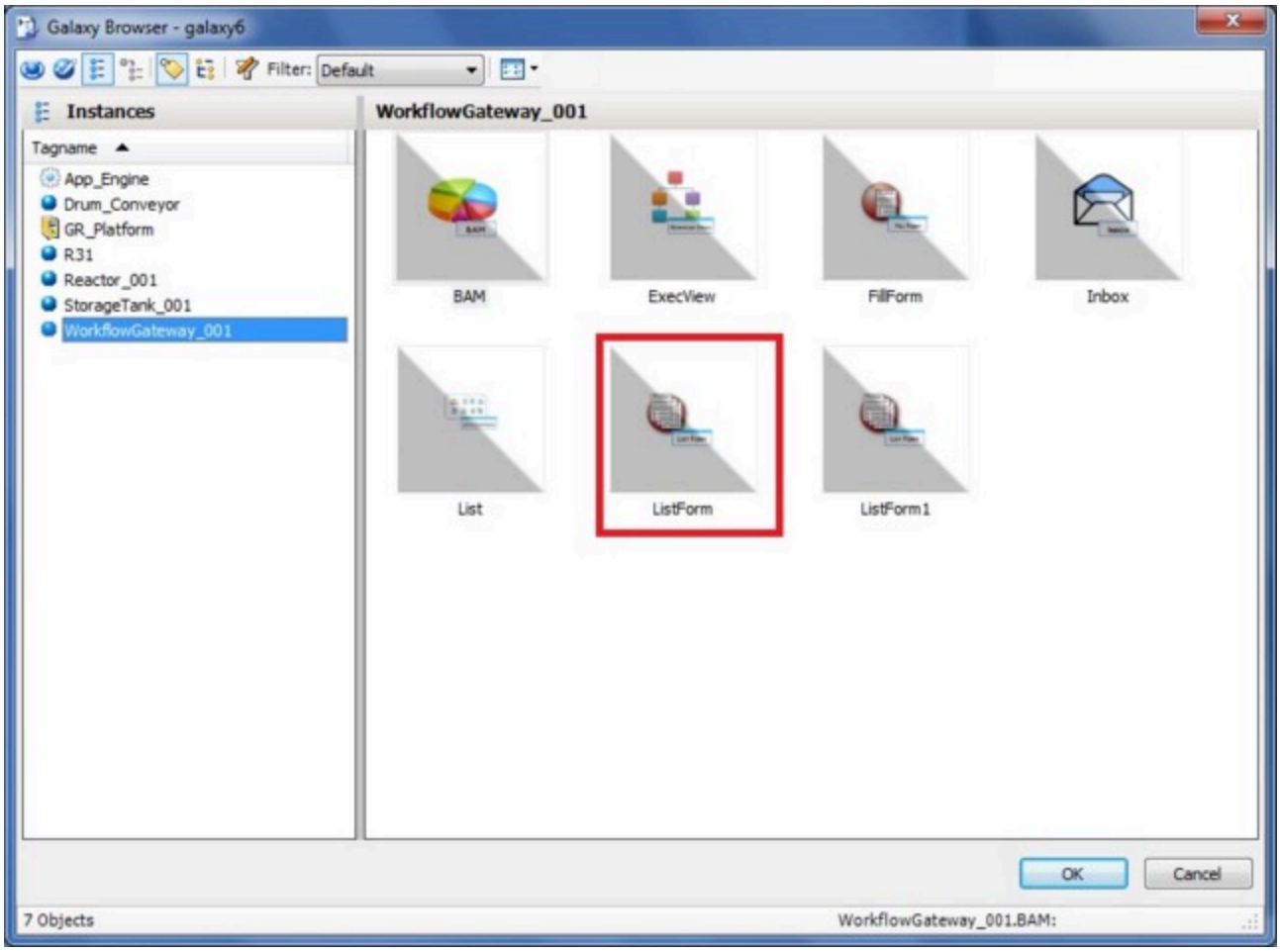
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



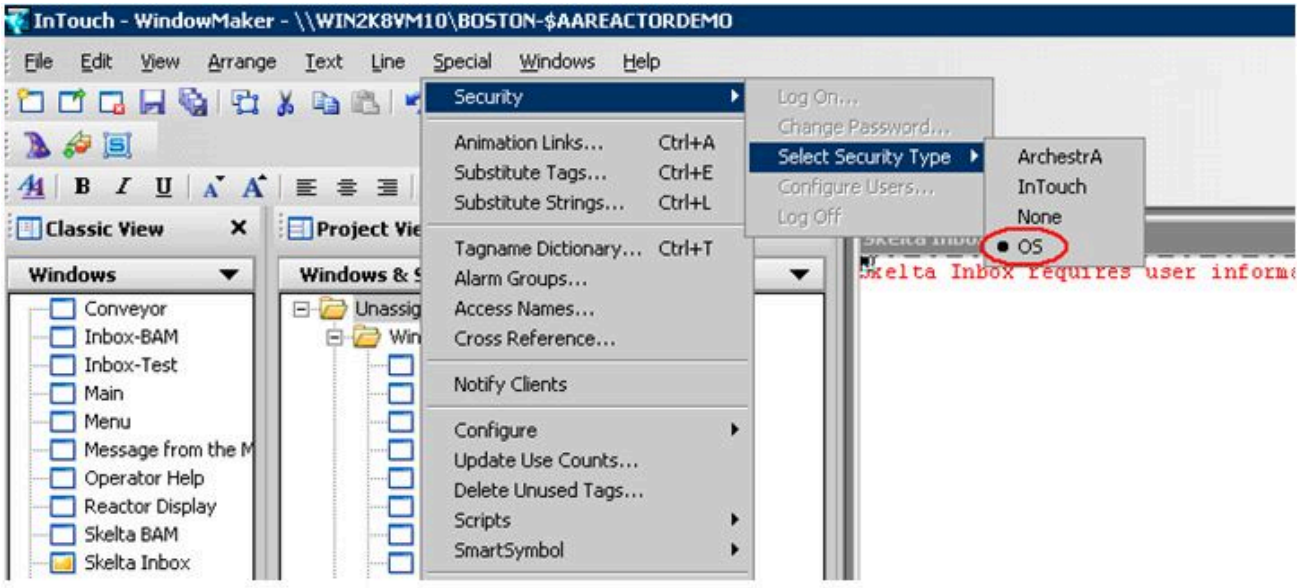
This opens up the ArchestrA Galaxy Browser.

9. Select the Instances tab and select the object to which the graphic symbol has been added.
10. Select the name of the graphic object (ListForm) that has to be displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



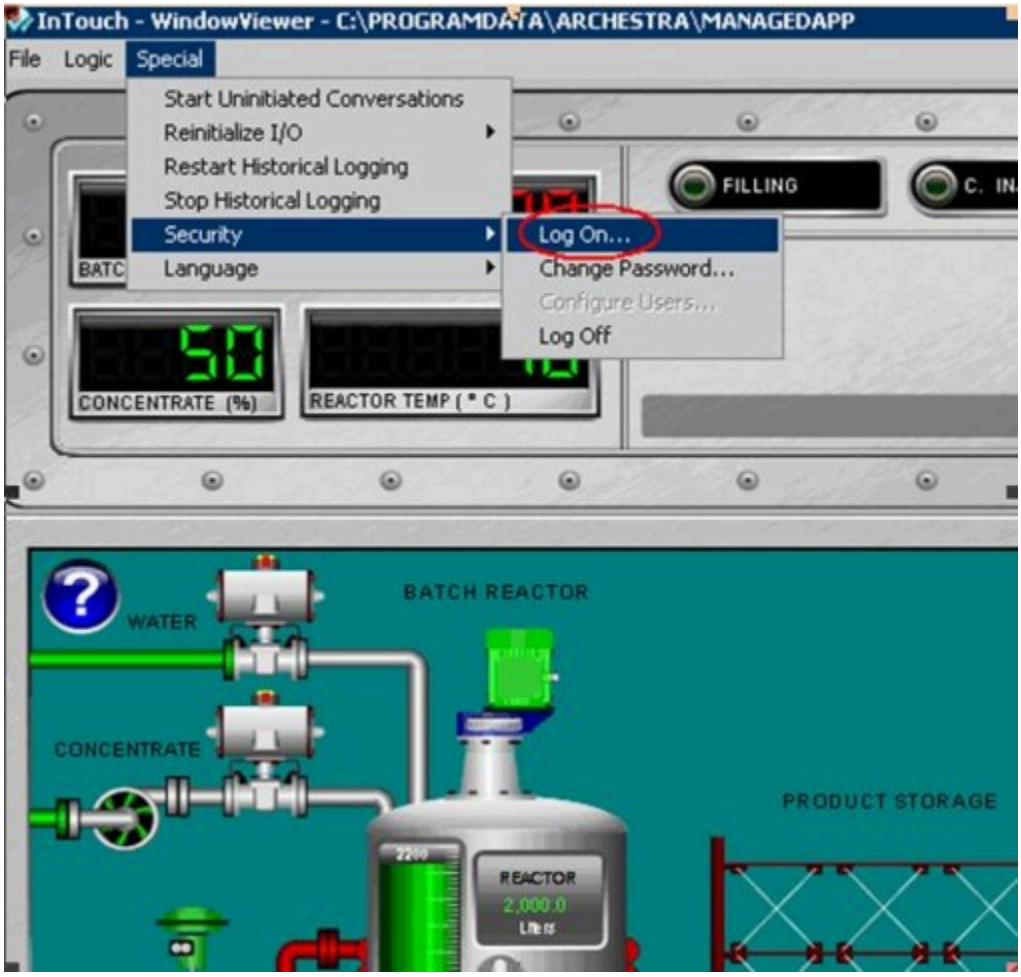
11. Enable security for the Lists window if required.

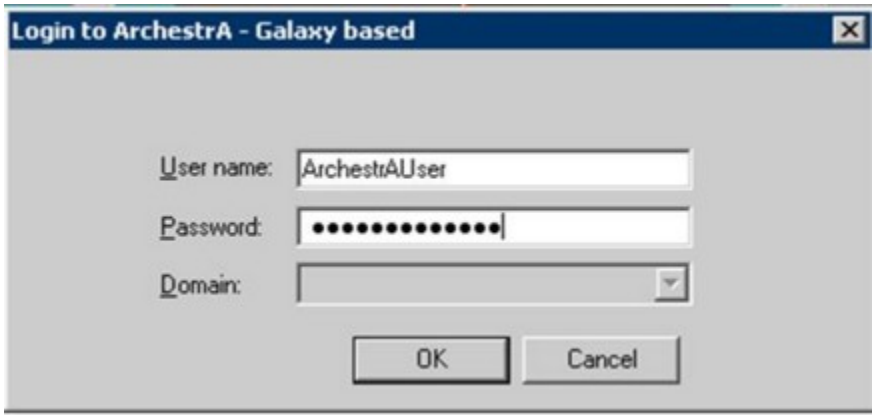


12. Save the window and exit InTouch WindowMaker.
13. Save and close the object in the System Platform IDE.
14. Deploy the $InTouchViewApp instance.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

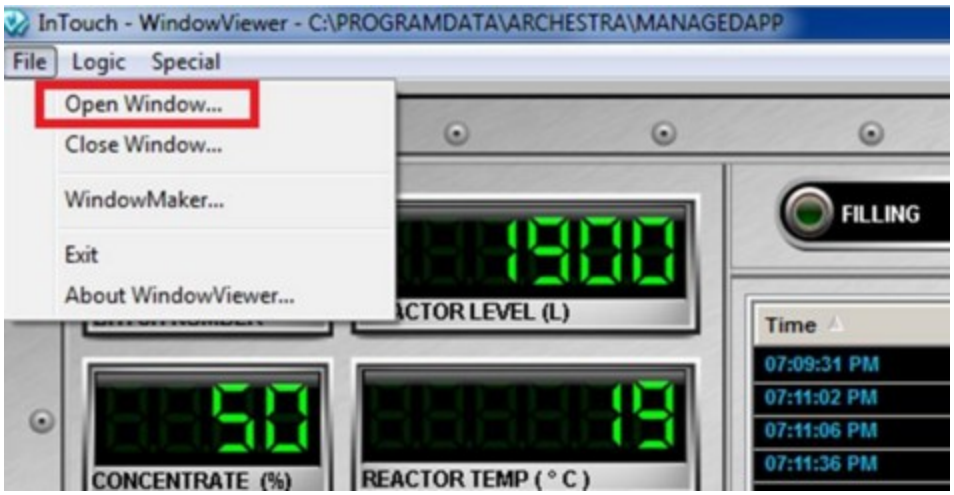
15. Open the InTouch Viewer and navigate to Special-> Security->Log On.
16. Enter the credentials and click OK.



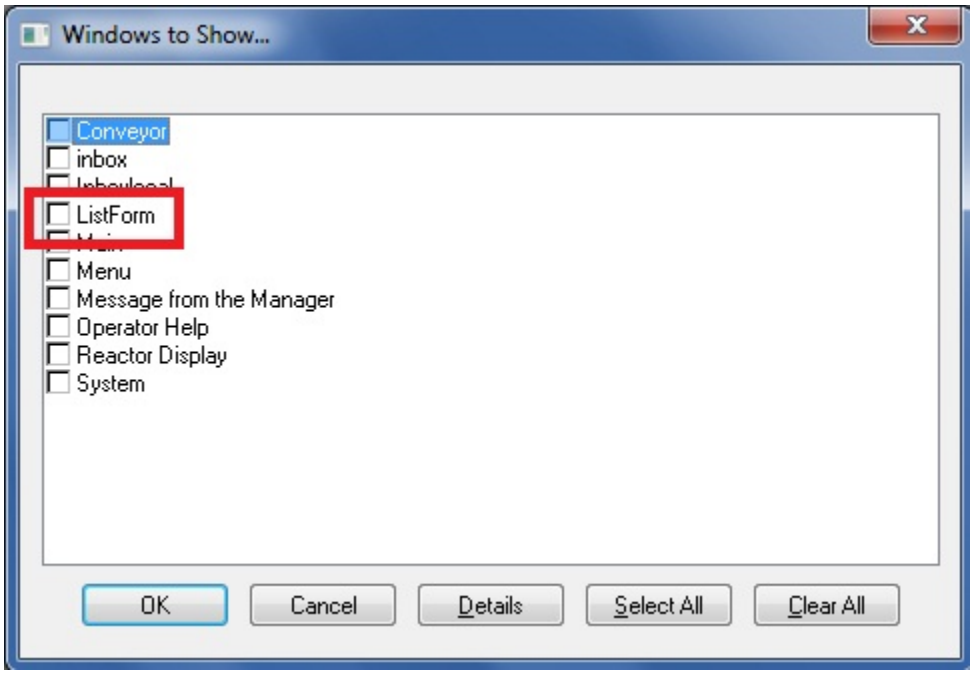


17. Open the created List Form window from the File menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The Lists Control loads in the window.



WorkItemInboxCount in InTouch

The WorkItemInboxCount control in InTouch displays the count of work items in the Inbox that have not been acknowledged or acted upon by the user. Also, the work items that have been assigned to Queues will be considered only when they are available in the user's Inbox. The count being displayed can be refreshed automatically or manually based on the configuration settings.

To use WorkItemInboxCount In InTouch

Follow Step 1 to Step 7 listed in the Initial Steps and continue:

1. Select the control in the editor and double-click.
2. Select Configuration under Animations to configure the control.

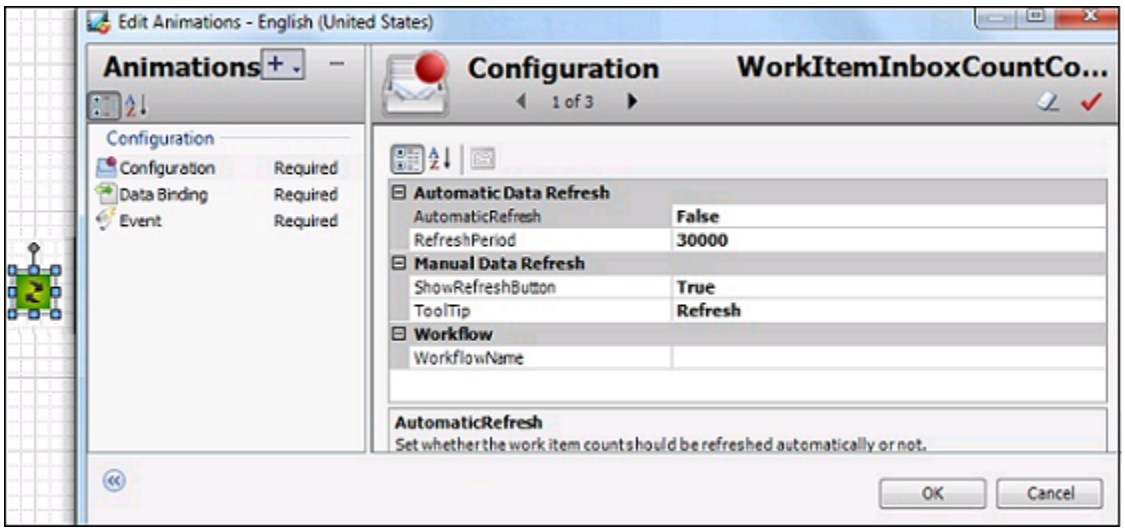
   Following are the properties of WorkItemInboxCount control:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- AutomaticRefresh: This property is used to set whether the work item count should be refreshed automatically or not.
- RefreshPeriod: This property is used to set the refresh period in milliseconds. The minimum value for Refresh Period is 30000.

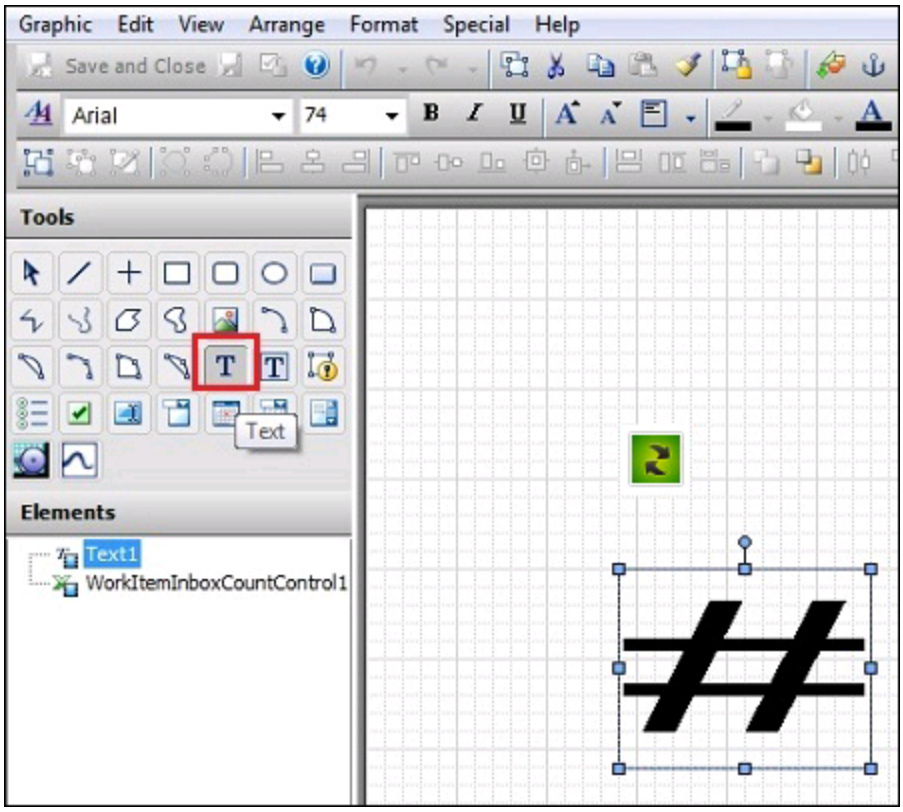
  If value of the RefreshPeriod property is set less than 30000, the value is automatically refreshed to the minimum value of 30000.
- ShowRefreshButton: This property is used to set whether manual data refresh button is visible or not.
- ToolTip: This property is used to set the tool tip for the Refresh button.
- WorkflowName: This property is used to set the workflow name for which the work item count will be visible. It has to be left blank to get the work item count of all the workflows.

3. After configuring the control, click OK.



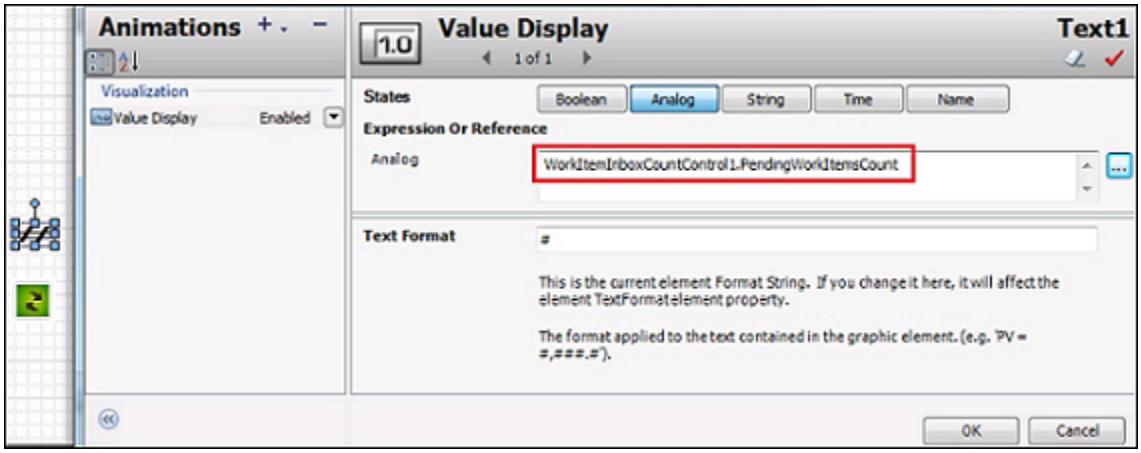
4. Drag and drop a Text control and set its Text property value as '#'.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



This hash (#) feature of InTouch will be used to display the pending work item count. The public property "PendingWorkItemsCount" of the WorkItemInboxCount control will be updated with the pending work item count.

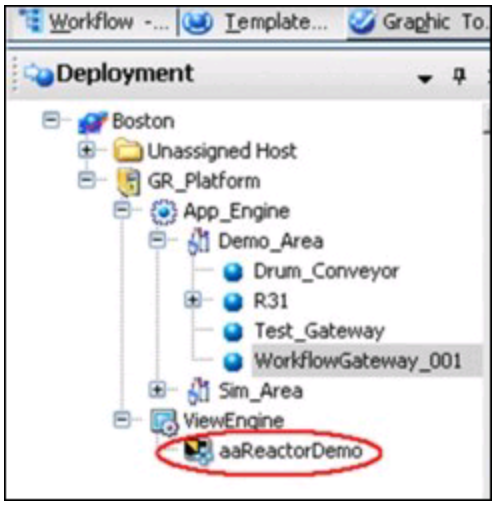
will be replaced by the value of the PendingWorkItemsCount property.



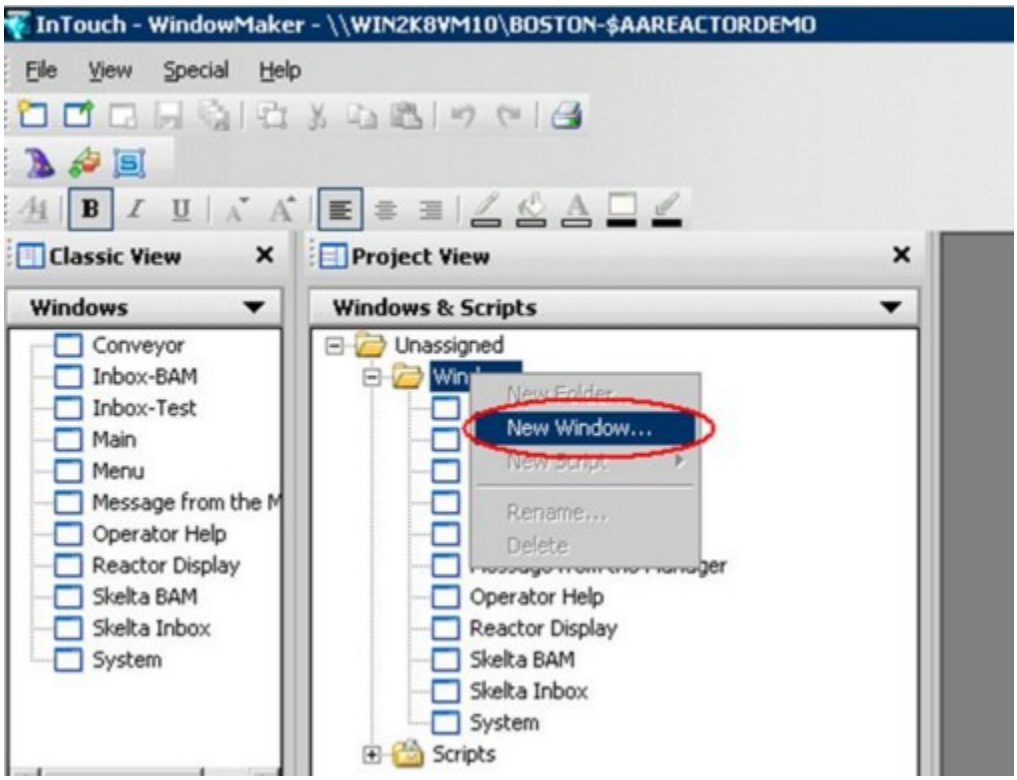
5. Select the WorkflowGateway object instance and deploy it in the System Platform IDE.
6. Double-click the $InTouchViewApp instance (aaReactorDemo) under ViewEngine from the System Platform IDE.

   The InTouch WindowMaker appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

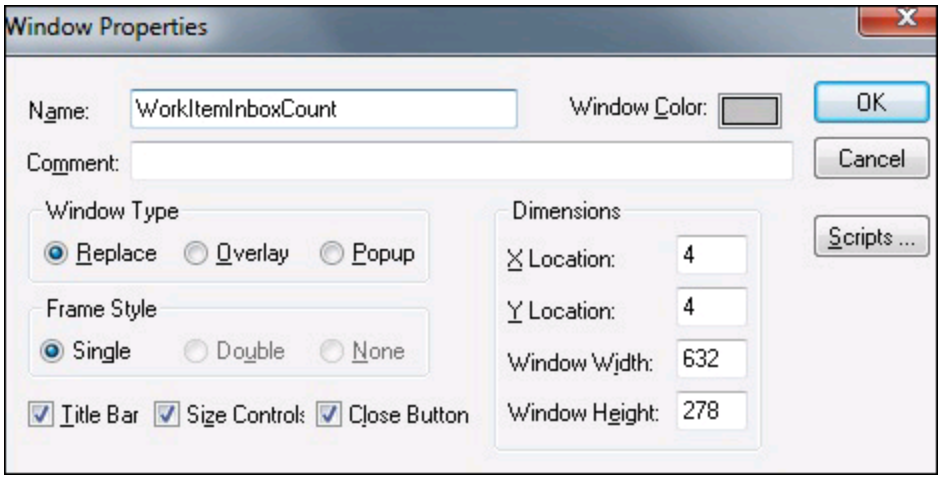


7. Create a new window for the WorkItemInboxCount control.



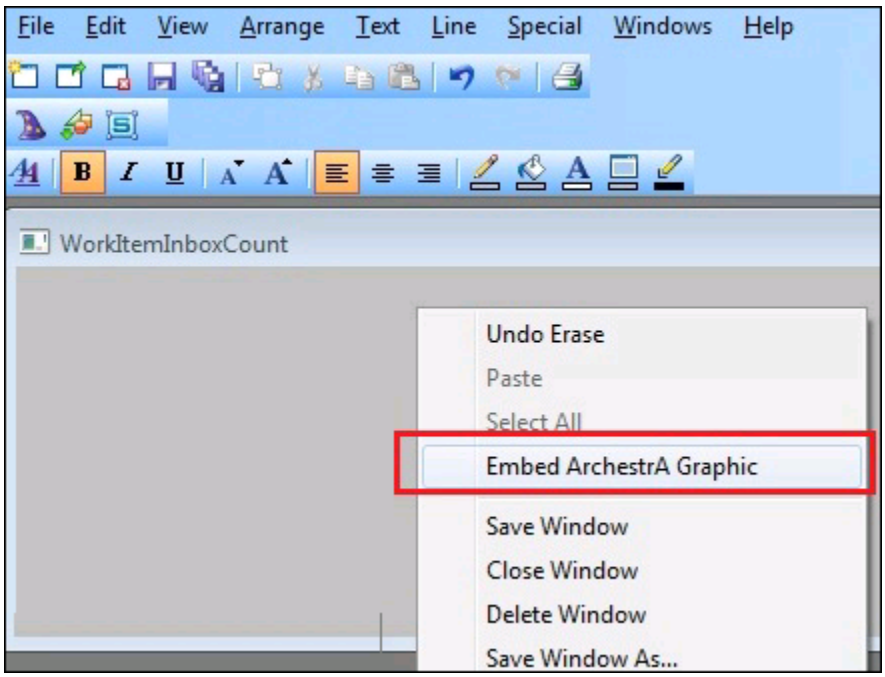
8. Provide the name and size for the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



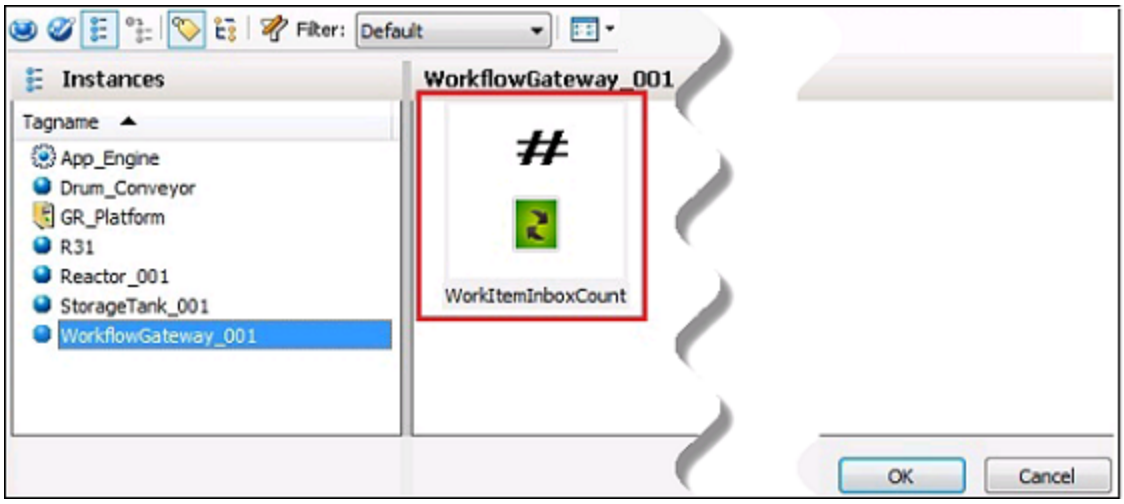
9. Right-click in the WorkItemInboxCount window and select **Embed ArchestrA Graphic** option.

   The ArchestrA Galaxy Browser appears.

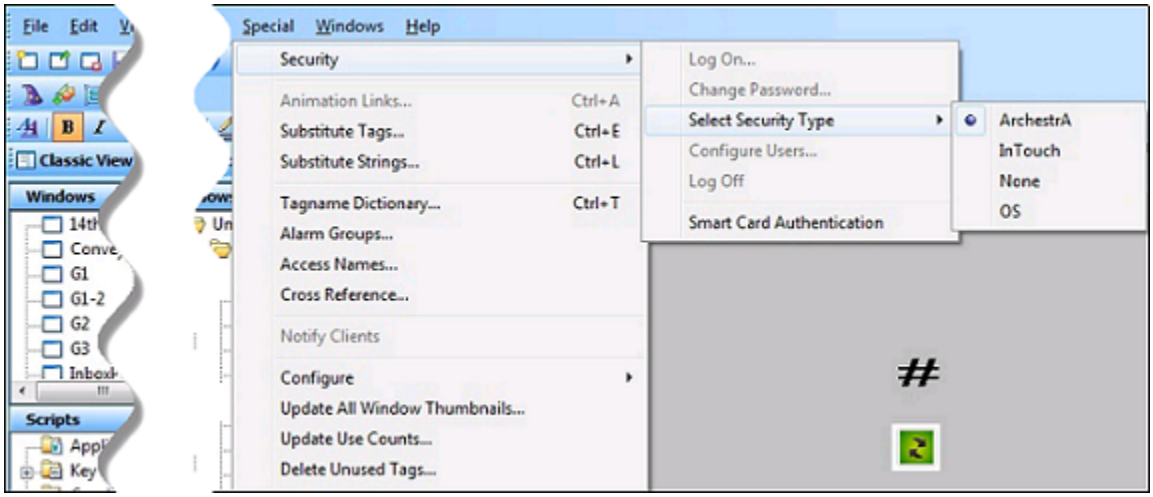


10. Select the Instances tab and select the WorkflowGateway instance to which the graphic symbol has been added.
11. Select the required graphic object ('WorkItemInboxCount').

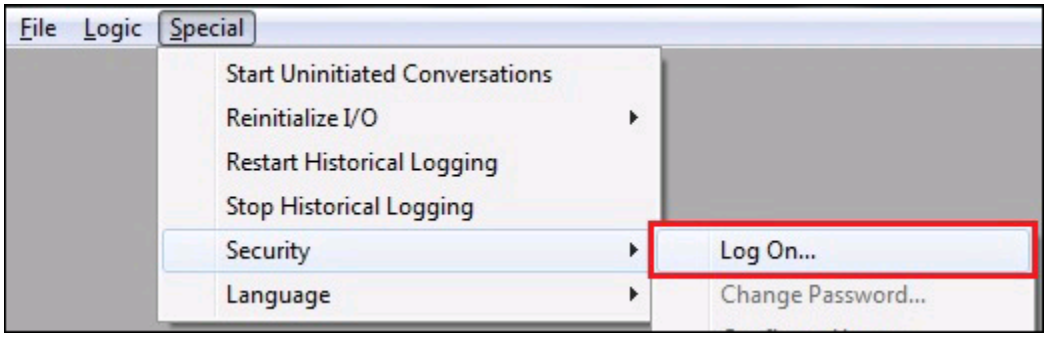
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



12. Enable security for the 'WorkItemInboxCount' window.
13. Select Security Type as "ArchestrA".

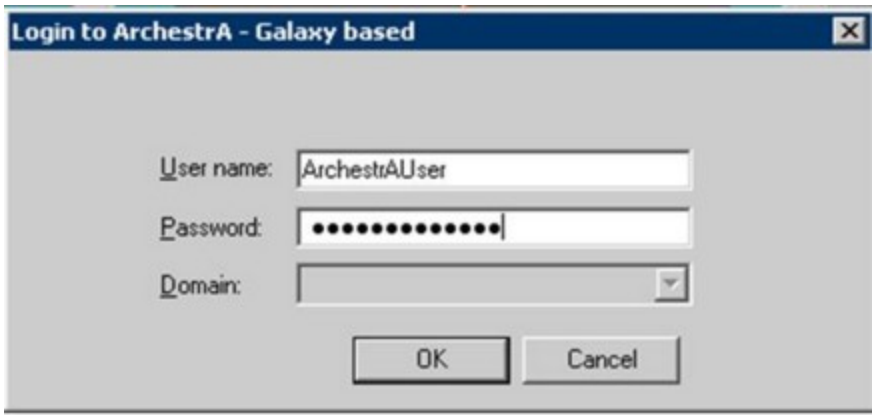


14. Save the window and exit InTouch WindowMaker.
15. Save and close the object in the System Platform IDE.
16. Deploy the $InTouchViewApp instance.
17. Click the Runtime button on the right hand side.
18. Open the InTouch Viewer and navigate to Special > Security > Log On.



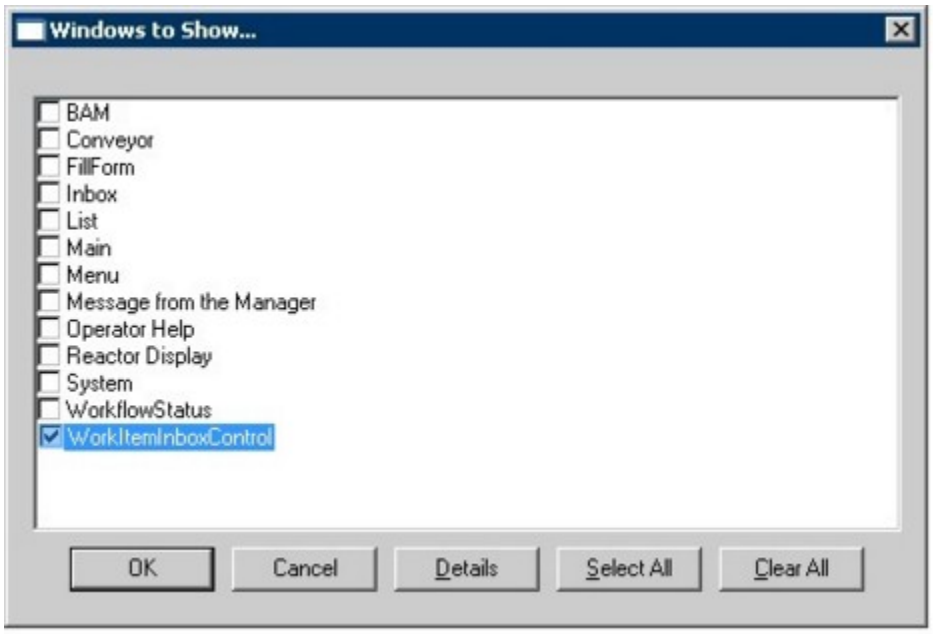
19. Enter the Credentials and click OK.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



20. Open the Work Item Inbox Count (the one created) window from the file menu.

    The WorkItemInboxCount control loads in the window.



Note: To use new controls after upgrading AVEVA Work Tasks, re-import ArchestrA.InTouch.SkeltaControls.dll. In the Import Preference dialog select 'Skip: Do not import' to ensure the existing controls are not overwritten. For more details about how to import ArchestrA.InTouch.SkeltaControls.dll, see Initial Steps.

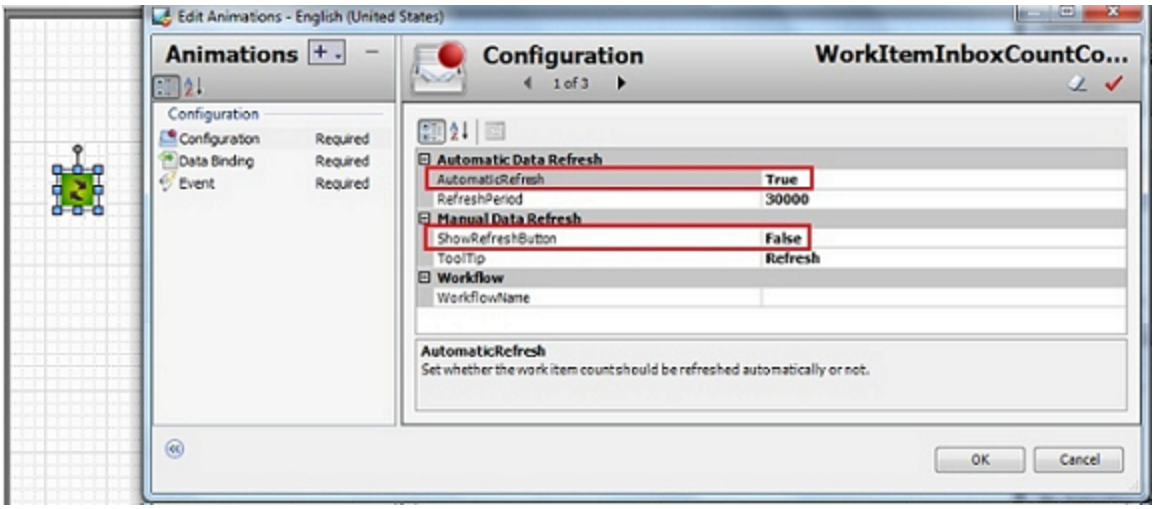
Open Inbox using WorkItemInboxCount

This section gives you step by step description on how to use the WorkItemInboxCount control to open Inbox:

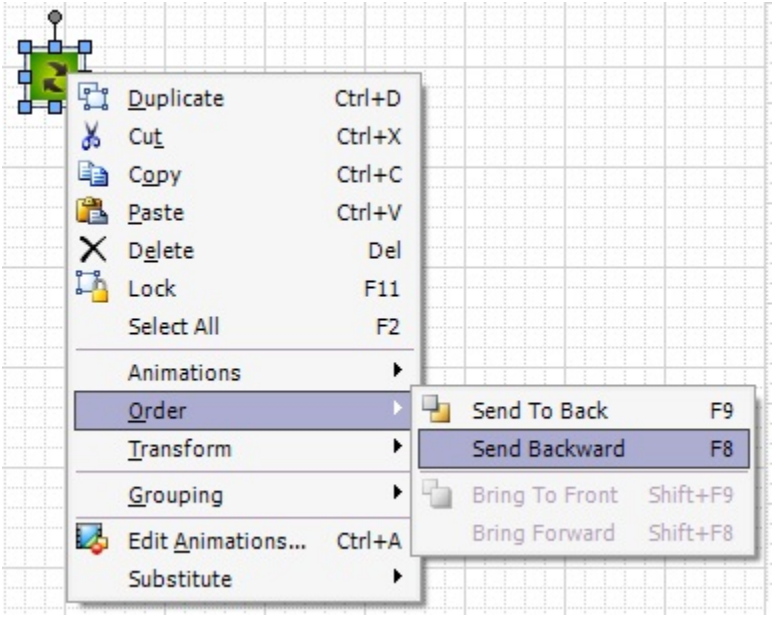
Follow steps 1-7 listed in the Initial Steps and continue:

1. Select the control in the editor and double-click. Select Configuration under Animations to configure the control. Set the properties as shown below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

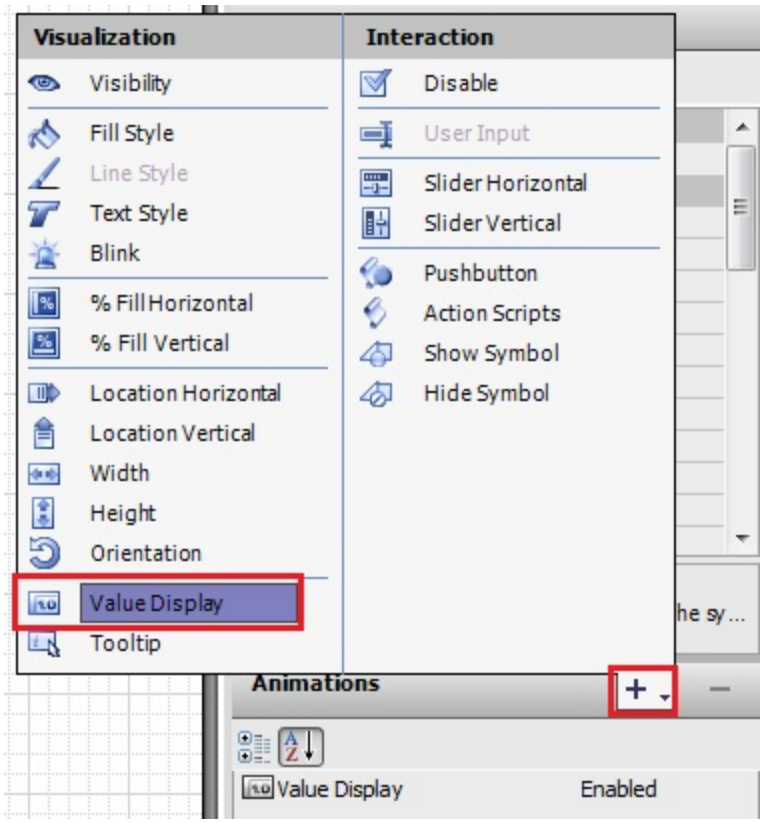


2. Right click the control and select Order -> Send Backward.



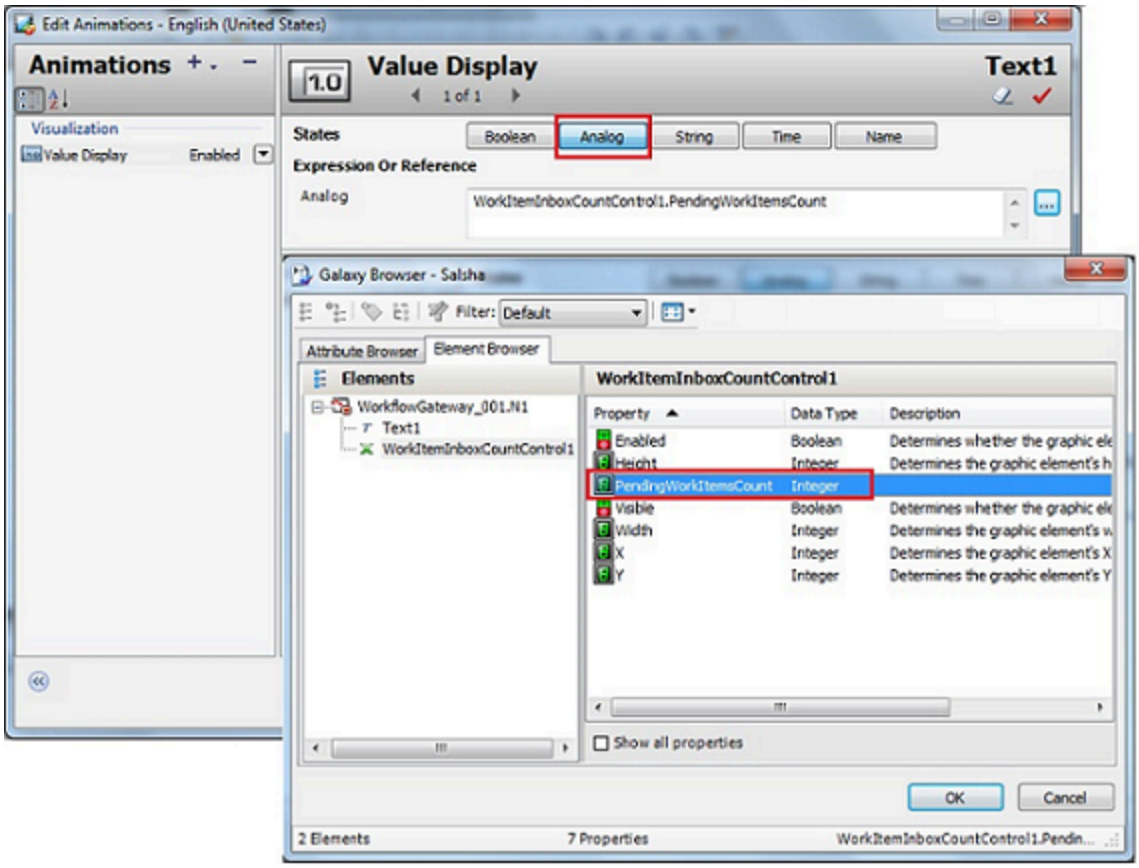
3. Add a button and place it above the WorkItemInboxCount control. Select Value Display as highlighted in the image given below. The Text property of the button should be set as #. # would be replaced by the actual work item count during run-time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

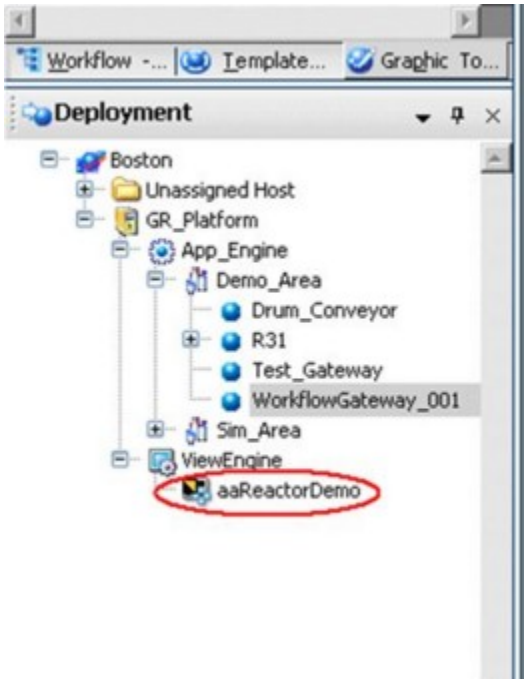


4. In the Value Display screen select Analog.
   The Galaxy browser window appears.
5. Select PendingWorkItemsCount from the Element Browser tab.
6. Click OK.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

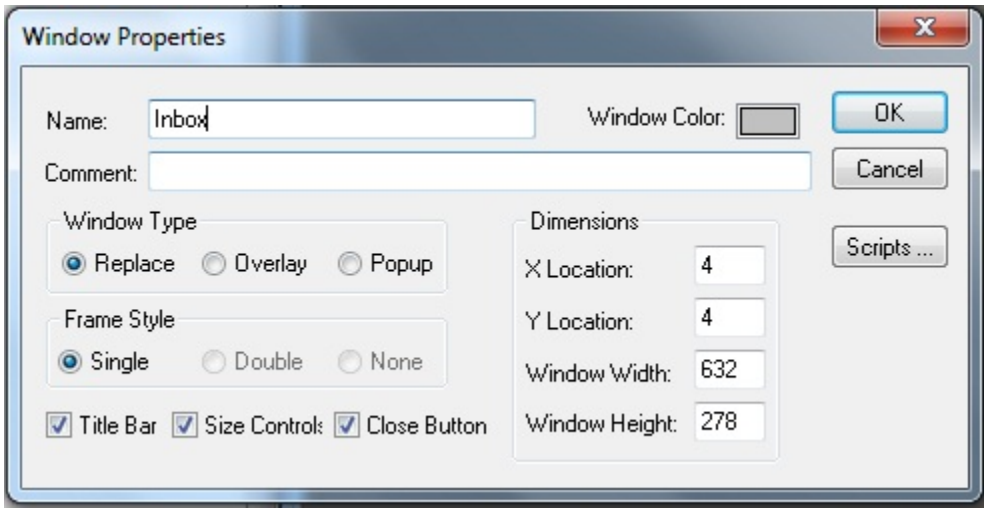


7. Create a graphic symbol for Inbox.
8. Deploy the respective Workflow Gateway Object.
9. Double-click the $InTouchViewApp instance (aaReactorDemo) under ViewEngine from the System Platform IDE.

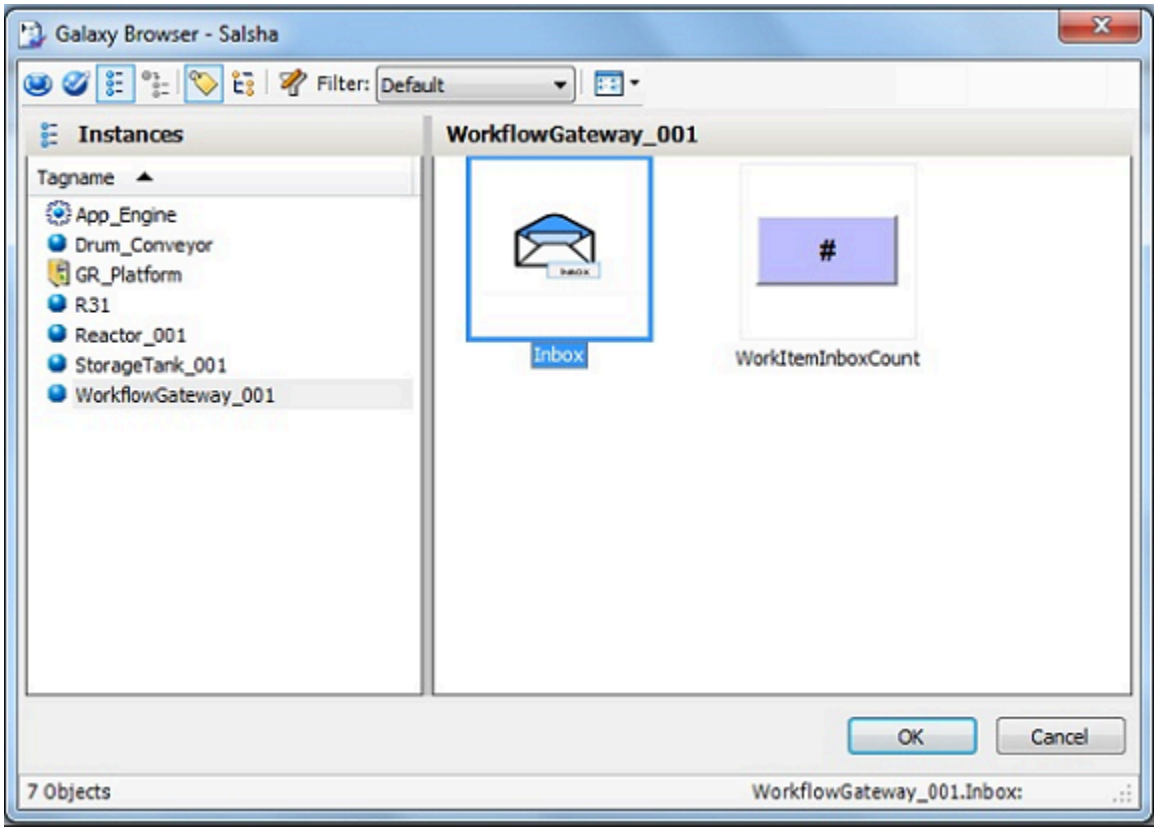


© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

10. Click File -> New Window. Add a new window named Inbox.

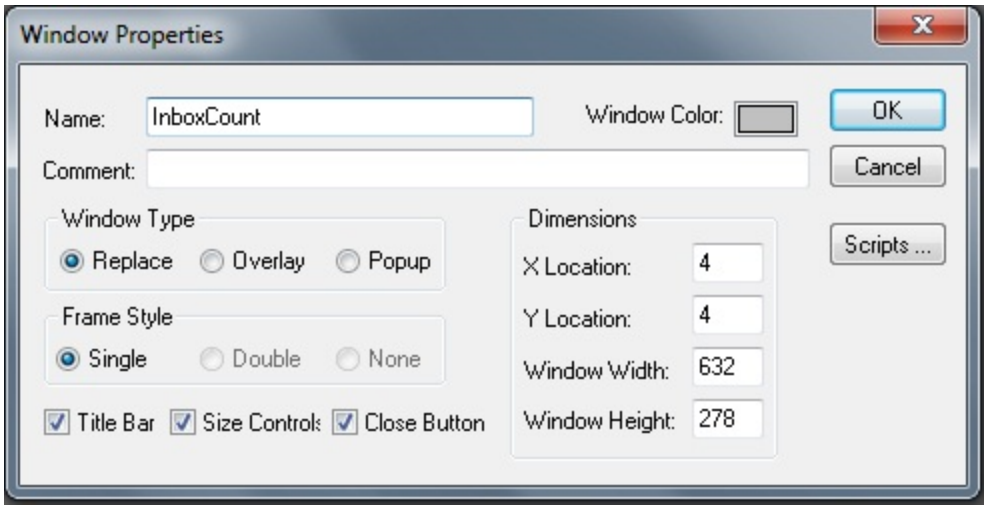


11. Open the Galaxy Browser.
12. Select the respective Workflow Gateway Object.
13. Select the Inbox graphic symbol and then click OK.

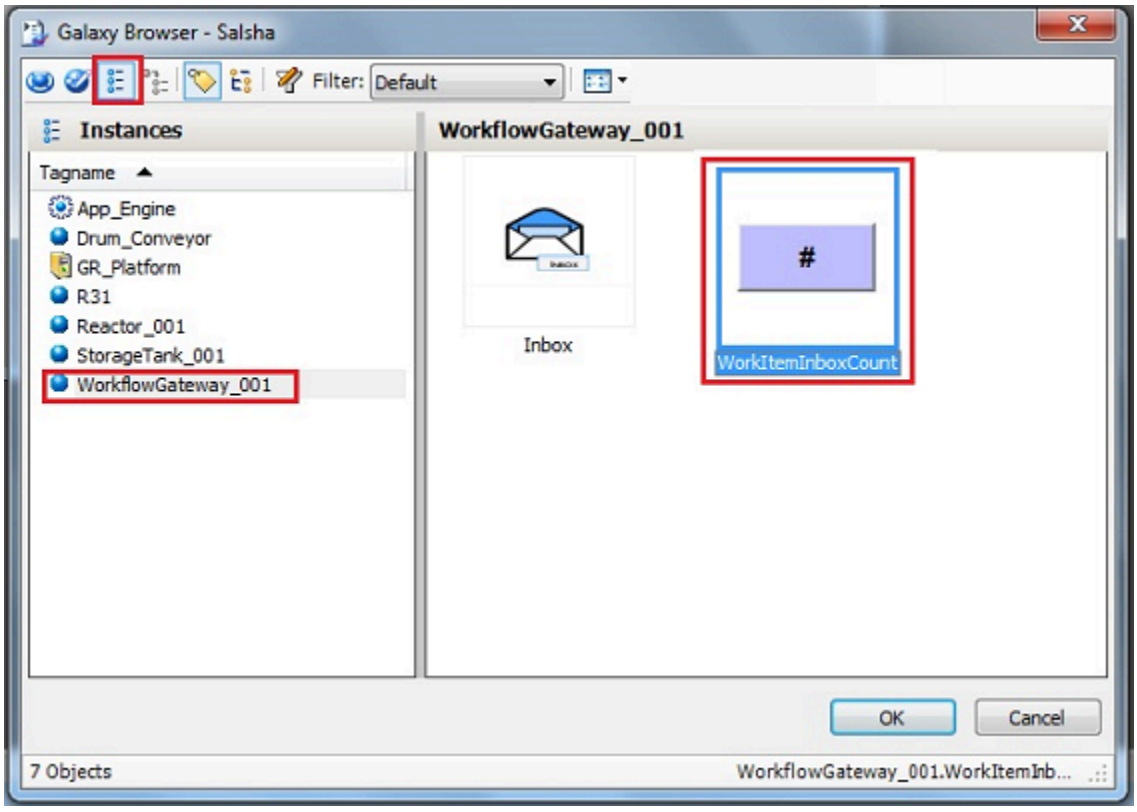


14. Save and close the Inbox window.
15. Click File -> New Window and add a new window named InboxCount.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

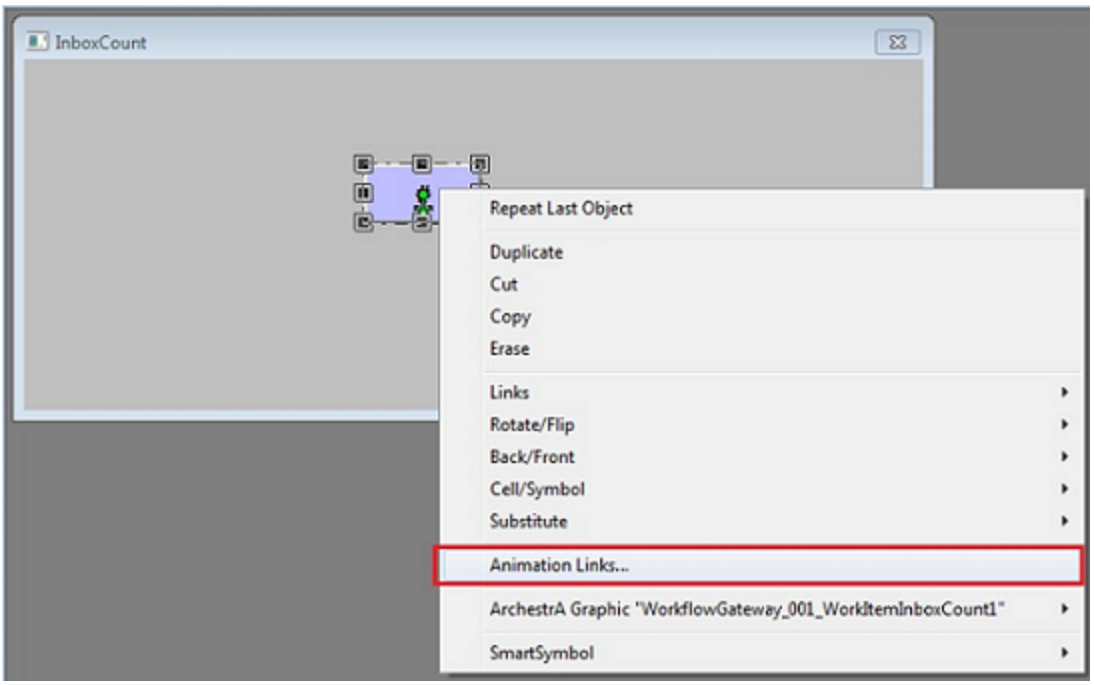


16. Open the Galaxy Browser.
17. Select the respective Workflow Gateway Object.
18. Select the WorkItemInboxCount graphic symbol and then click OK.

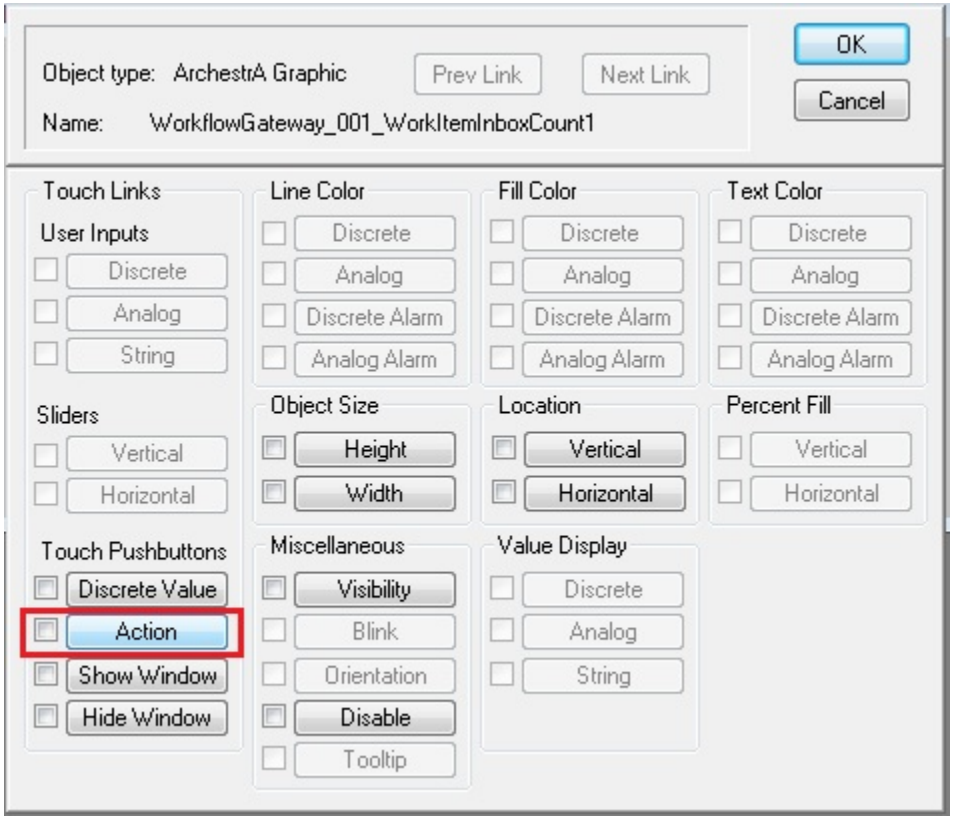


19. InboxCount screen appears. Right click the button and select Animation Links.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

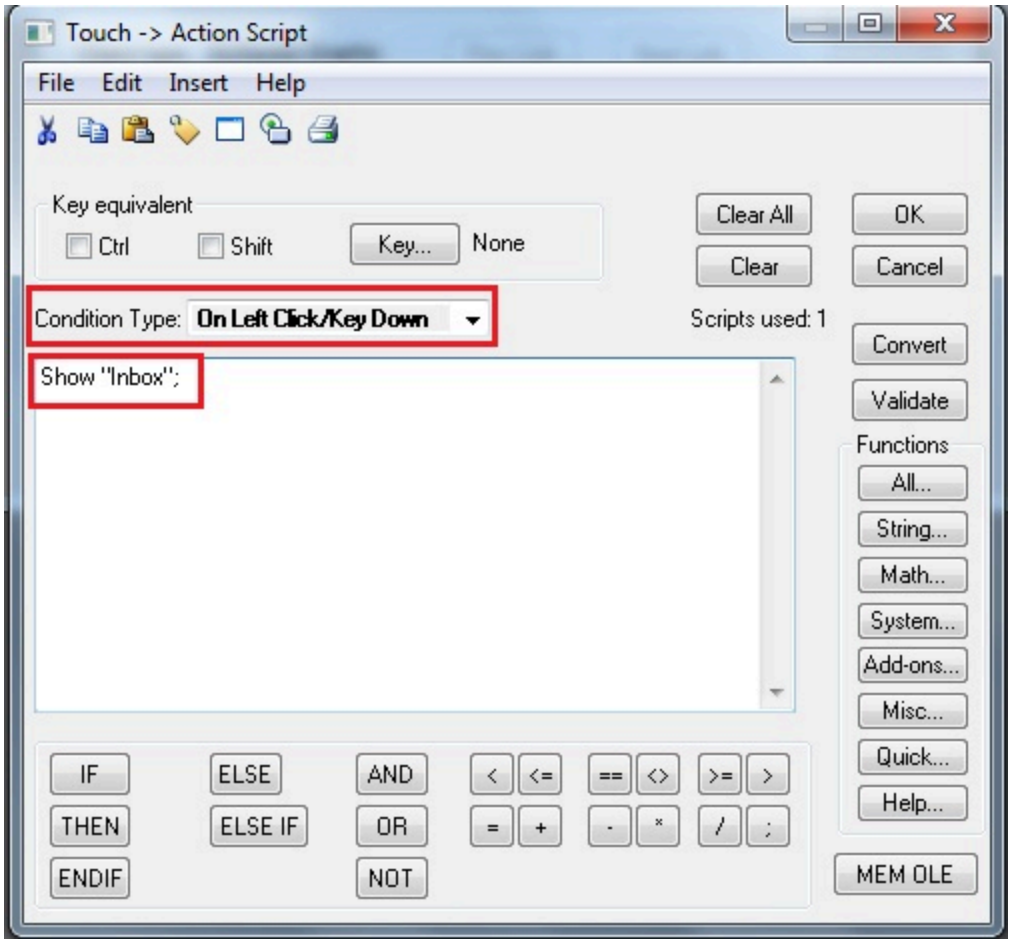


20. Click Action in the screen which appears.

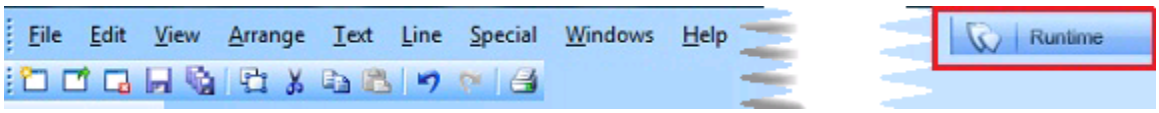


21. Select the Condition Type as On Left Click/Key Down and write the script as Show "Inbox".
22. Click OK.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

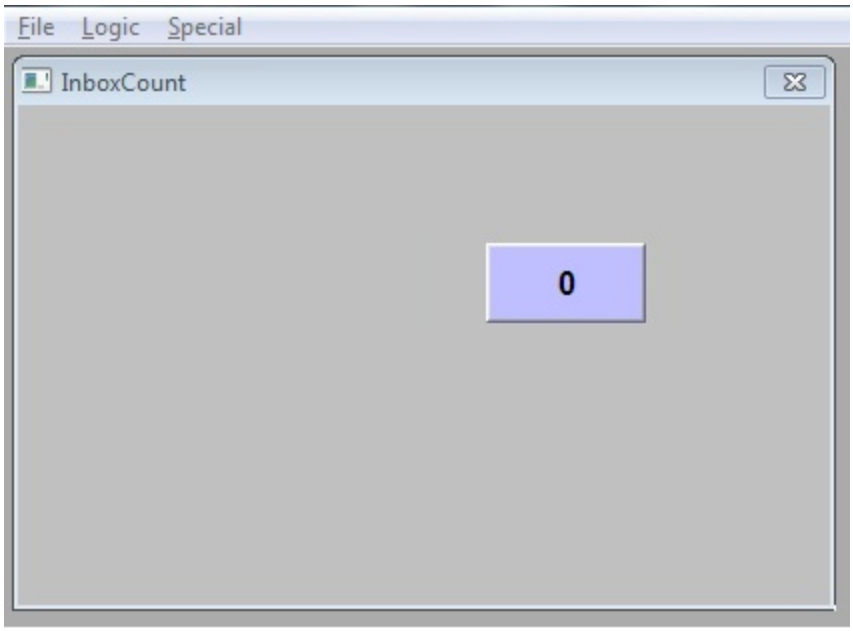


23. Click the Runtime button on the right hand side.

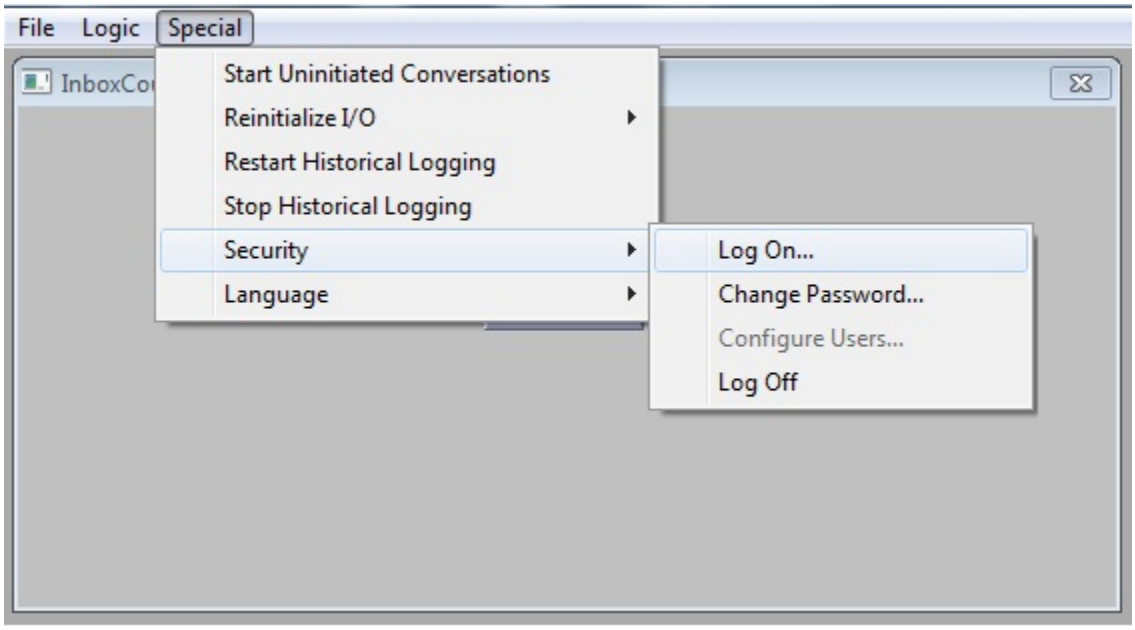


The InboxCount window appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

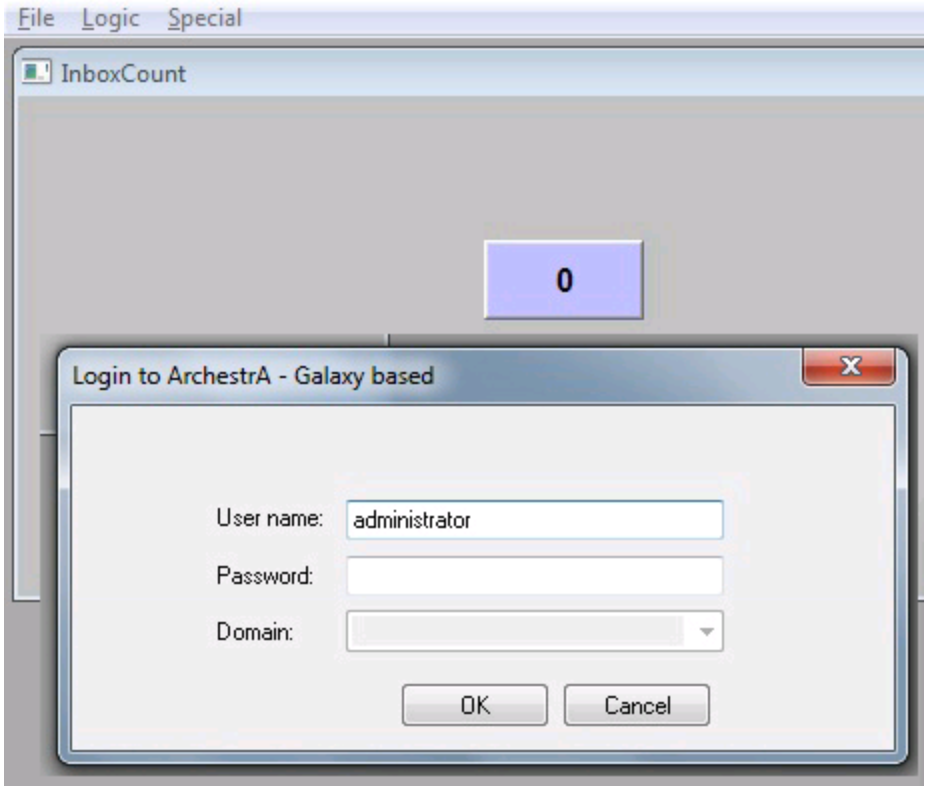


24. Navigate to Special -> Security -> Log On.

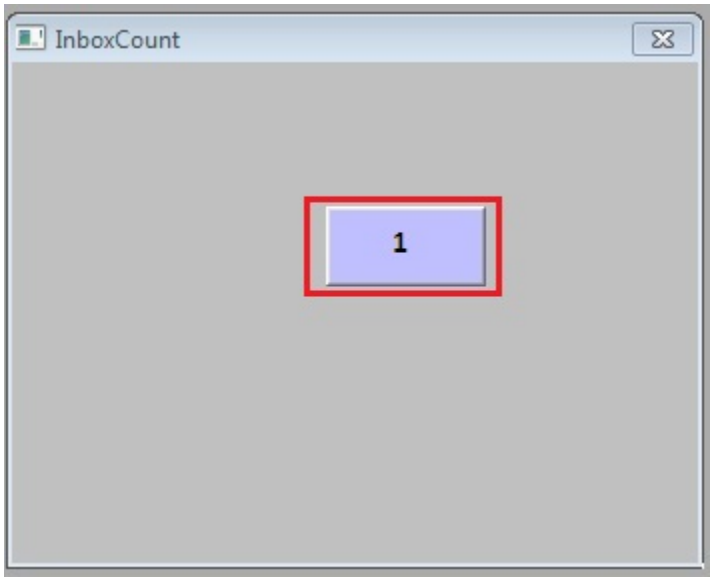


25. The login screen appears. Enter User Name and Password. Click OK.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

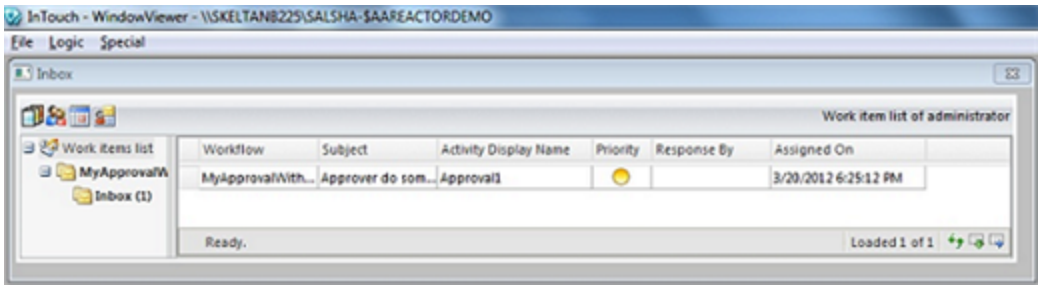


26. The actual work item count gets updated. Click the button to open the Inbox.



27. The Inbox screen appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Workflow Features Inside InTouch OMI

This topic provides information on how to use the Workflow Features inside InTouch OMI.

- Initial Steps
- AIM Authentication for Controls in InTouch OMI
- Forms Inside InTouch OMI
- Inbox Inside InTouch OMI
- List View Inside InTouch OMI
- List Form Inside InTouch OMI
- BAM Inside InTouch OMI
- Workflow Status Inside InTouch OMI
- Inbox Work Items Count Inside InTouch OMI
- Cryptography Settings Inside InTouch OMI

Initial Steps

You can use workflow features from inside **InTouch OMI**.

Note: The following steps are applicable when using System Platform 2023.

1. Setup the Initial Gateway Setup.
2. In the **System Platform IDE**, select **Import** > **Visualization** > **OMI App** from the **Galaxy** menu.
   The **Import OMI App** dialog box appears.
3. Select the **<ArchestrA Installation Path>\ArchestrA\Workflow\InTouchOMI\AVEVAWorkTasksApp** folder.
   The imported app appears as **WorkTaskApp00X** in the **Graphics**.
4. Select **Home** menu, and then select **Screen** from the ribbon bar.
   A new screen profile is created with the default settings, and appears in the **Graphics**.
5. Select **Home** menu, and then select **Layout** from the ribbon bar.
   A new layout is created with the default settings, and appears in the **Graphics**.
6. In the **Template**, select **<Galaxy Name>** > **System** > **$ViewEngine**.
7. Right-click **$ViewEngine**, select **New**, and then select **Instance**.
   A new instance of the view engine is created, and appears in the **Templates**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

8. In the **Templates**, select **<Galaxy Name>** > **ViewApps** > **$ViewApp**.
9. Right-click **$ViewApp**, select **New**, and then select **Derived Template**.

   A new derived template is created, and appears in the **Templates**.

   Double-click and configure (screen profile and layout) the derived template, if required.
10. Right-click the derived template, select **New**, and then select **Instance**.

    A new instance of the derived template is created and will appear either in the **Unassigned Area** or in the designated container set as default.
11. Drag and drop the instance of the ViewEngine to the required platform.
12. Drag and drop the instance of the ViewApp to the view engine.

If you are using a version earlier than System Platform 2023, follow these steps:

1. Setup the Initial Gateway Setup.
2. In the **System Platform IDE**, on the **Galaxy** menu, click **Import**, and then click **ArchestrA Apps**.

   The **Import ArchestrA App** dialog box appears.
3. Select the **<Installation Path>\Workflow\WorkTaskApp** folder.

   The imported app appears as **WorkTaskApp00X** in the **Graphic Toolbox**.
4. On the **Galaxy** menu, click **New**, and then click **Screen Profile**.

   A new screen profile is created with the default settings, and appears in the **Graphic Toolbox**.
5. On the **Galaxy** menu, click **New**, and then click **Layout**.

   A new layout is created with the default settings, and appears in the **Graphic Toolbox**.
6. In the **Template Toolbox**, select **<Galaxy Name> > System > $ViewEngine**.
7. Right-click **$ViewEngine**, click **New**, and then click **Instance**.

   A new instance of the view engine is created, and appears in the **Template Toolbox**.
8. In the **Template Toolbox**, select **<Galaxy Name> > System > $ViewApp**.
9. Right-click **$ViewApp**, click **New**, and then click **Derived Template**.

   A new derived template is created, and appears in the **Template Toolbox**.

   Double-click and configure (screen profile and layout) the derived template, if required.
10. Right-click the derived template, click **New**, and then click **Instance**.

    A new instance of the derived template is created, and appears in the **Template Toolbox**.
11. Drag and drop the instance of the view engine to the required platform.
12. Drag and drop the instance of the view app to the view engine.

Note: If the screen profile and layout are not configured when the derived template is created, then configure the same by double-clicking the instance of the view app.

Localizing AVEVA Work Tasks Controls in InTouch OMI

By default, locales are provided for **English (United States)** at **<Installation Path>\Workflow\InTouchOMI\WorkTaskApp\Locales\en-US** folder. For other locales, create locale files in the respective locales folder, and change the regional settings of the operating system to match the locale.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For example, to change the locale to **English (United Kingdom)**, create the locale files in the **en-GB** folder of the **Locales** folder, and change the regional settings of the operating system to **English (United Kingdom)**.

By default, the locale of the object is set to **English (United States)**.

AIM Authentication for Controls in InTouch OMI

You can use AVEVA Identity Manager (AIM) authentication for AVEVA Work Tasks Controls in **InTouch OMI**. To enable AIM authentication, ensure that in **Central Configuration** when the **Resource Provider** is set to **Active Directory**, the **Use Token Service** is set to the appropriate server. Otherwise, the control will be loaded within the context of the default user, if specified. If the default user is not specified, then the control will be loaded within the context of the user who configured the workflow gateway.

Note:
- Ensure to import the ArchestrA app before configuring AIM for InTouch OMI.
- When using AIM authentication for controls in InTouch OMI, ensure all controls are configured for AIM authentication.

Configuring application to point to AIM

Preview View App

1. Go to **<ArchestrA Installed Path>\Framework\Bin\ModernEditors\Preview\ViewAppFramework**.
2. Edit **View.exe.config** file to include the following key value pairs:

```
<!-- Configure the AIM by adding the following key value pairs.
Replace "your_token_server" and "your_client_id" with proper values. -->
<add key="TokenServiceBase" value="https://<AIM Server Name>/identitymanager" />
<add key="ClientId" value="SE_WSP_NG" />
<add key="Scope" value="openid profile system offline_access" />
```

Deployed View App

1. Go to **<ArchestrA Installed Path>\Framework\Bin\ViewAppFramework**.
2. Edit **View.exe.config** file to include the following key value pairs:

```
<!-- Configure the AIM by adding the following key value pairs.
Replace "your_token_server" and "your_client_id" with proper values. -->
<add key="TokenServiceBase" value="https://<AIM Server Name>/identitymanager" />
<add key="ClientId" value="SE_WSP_NG" />
<add key="Scope" value="openid profile system offline_access" />
```

Configuring System Platform IDE Security

1. In the **System Platform IDE**, on the **Galaxy** menu, click **Configure**, and then click **Security**.
   The **Security** dialog box appears,
2. On the **Authentication Mode** tab, select **OS User based** or **OS Role based**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. On the **Users** tab, under **Authorized Users Available**, add users, and then under **Associated Roles for Administrator**, map **Administrator** role.
4. Click **OK**.

   The **System Platform IDE** will close and reopen.

Using AIM authentication for controls

1. **Complete the** Initial Steps.
2. In the **System Platform IDE**, double-click on the instance of the view app.

   The instance of the view app is opened.
3. Open the layout and add a panel on the top.
4. On the **Toolbox** tab, click **TitleBarApp**, and then drag and drop **TitleBarControl** in the **Layout**.
5. On the **Toolbox** tab, click **WorkTaskApp**, and then drag and drop a control to the default panel in the **Layout**.
6. On the **Properties** tab, configure other settings for the control as required.
7. Click **Save**.

   The control, along with the set properties, is saved.
8. Click **Preview** to preview the control in **InTouch OMI**, if required.
9. In the **System Platform IDE**, right-click on the instance of the view app, and then click **Deploy**.

   The **Deploy** dialog box appears, and shows the progress of the deployment.
10. Click **Close**.

    The control can now be used inside **InTouch OMI**.
11. Run **AVEVA OMI Application Manager**, and then click **Launch** for the configured instance of the view app to use the control.

Note:
- When running **AVEVA OMI Application Manager**, if you get an error that “You should be part of either the Administrators, WorkTasksAdmins, or WorkTasksRemoteUsers role.”, then run **AVEVA OMI Application Manager** as an **Administrator**.
- For customization purposes, you can manually create a symbol to log on instead of using the **TitleBarApp** as follows:

1. In the **System Platform IDE**, on the **Galaxy** menu, click **New**, and then click **Symbol**.
A new symbol is created, and appears in the **Graphic Toolbox**.
2. Double-click on the symbol, add two buttons, and then label them as **Login** and **Logout**.
3. Right-click **Login** button, click **Edit Animations**, and then add **Action Scripts** as follows:
```
ShowLoginDialog();
```
4. Right-click **Logout** button, click **Edit Animations**, and then add **Action Scripts** as follows:
```
Logout();
```
5. Open the layout and add a panel on the top.
6. On the **Toolbox** tab, click the symbol, and then drag it to the top panel in the **Layout**.
7. On the **Toolbox** tab, click the symbol, and then drag it to the top panel in the **Layout**.

Forms Inside InTouch OMI

You can use forms inside **InTouch OMI**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. **Complete the** Initial Steps.
2. In the **System Platform IDE**, double-click on the instance of the view app.

   The instance of the view app is opened.
3. On the **Toolbox** tab, click **WorkTaskApp**, and then drag and drop the **Fillform** control to a panel in the **Layout**.
4. On the **Properties** tab, map or enter the **Workflow Gateway**.
5. On the **Properties** tab, map or enter the **Form Name**.
6. On the **Properties** tab, map or enter the **Form Version**, if required.

   Note: If the version of the form is not specified, then the system makes use of the latest published version of the form.
7. On the **Properties** tab, map or enter the **Input XML**, if required.
8. On the **Properties** tab, set the **Display Alert Message** to **True**, if you want to display an alert message when the form is submitted.
9. On the **Properties** tab, set the **Default User Always** to **True**, if you want to specify the default user. Otherwise, the user who configured the **Workflow Gateway** object will be considered as the default user.
10. On the **Properties** tab, map or enter the **Default User**, if the **Default User Always** is set to **True**. The user set here will be considered as the user for submission of the form.
11. On the **Properties** tab, configure other settings as required.
12. Click **Save**.

    The control, along with the set properties, is saved.
13. Click **Preview** to preview the **Fill Form** in **InTouch OMI**, if required.
14. In the **System Platform IDE**, right-click on the instance of the view app, and then click **Deploy**.

    The **Deploy** dialog box appears, and shows the progress of the deployment.
15. Click **Close**.

    The **Fill Form** can now be used inside **InTouch OMI**.
16. Run **Wonderware Application Manager**, and then click **Launch** for the configured instance of the view app to use the **Fill Form**.

Note:
- The **Fill Form** control shows published forms including those not associated to workflows. It even shows forms disabled in the **Enterprise Console**.
- If **Security Level** changes are made with respect to the rights given to the user, then the view app should be redeployed before previewing the **Fill Form**.
- When running **Wonderware Application Manager**, if you get an error that “You should be part of either the Administrators, SkeltaAdmins, or SkeltaRemoteUsers role.”, then run **Wonderware Application Manager** as an **Administrator**.

If **InTouch OMI** is setup on a different machine, then launching the instance of the view app will lead to an error about unauthorized user. In such cases, configure the **Internet Options** as follows:

1. On the **Security** tab, under **Select a zone to view or change security settings**, click **Trusted sites**.
2. Click **Sites**, in the **Add this website to the zone** box, type the URL of the **Enterprise Console** set for the **Workflow Gateway** object, click **Add**, and then click **Close**.
3. Under **Security level for this zone**, click **Custom level**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Under **Settings**, browse to **User Authentication > Logon** and click **Automatic logon with current username and password**, and then click **OK**.
5. Click **OK** to close **Internet Options**.

Passing Contextual Information in Forms from InTouch OMI

You can pass real-time value from InTouch OMI property to Fill Form control. Passing contextual information through attributes or hard-coded value is supported only for String data type.

The XML that is passed populates values of the controls on the form when loading the form. You can pass only the XML for the required controls. However, ensure the XML matches the syntax of the instance XML and does not exceed 1023 characters.

Example

To populate only the text controls **TEXTINPUT1** and **TEXTINPUT2**:

Script the XML as follows:

```
<SKRootDefinition><TEXTINPUT1>Value1
</TEXTINPUT1><TEXTINPUT2>Value1</TEXTINPUT2></SKRootDefinition>
```

OR

Bind the attribute as follows:

```
InputXmlAttribute="<SKRootDefinition><TEXTINPUT1>"+Galaxy:Drum_Conveyor.DrumCount+"</TEXTIN
PUT1><TEXTINPUT2>"+Galaxy:Drum_Conveyor.Speed+"</TEXTINPUT2></SKRootDefinition>";
```

Here **InputXmlAttribute**, **Galaxy:Drum_Conveyor.DrumCount**, and **Galaxy:Drum_Conveyor.Speed** are attributes, and the **InputXmlAttribute** attribute is of string data type.

Passing Contextual Information in Forms using Navigation App

You can pass the contextual information from the selected Asset using Navigation app to Fill Form control. When you use the Navigation app, you need to enter the attribute names in braces {}.

Example:

Form Name: {Attribute001}

Form Version: {Attribute002}

Input XML: {Attribute003}

Form Name is read from the {Attribute001} of the selected Asset.

Form Version is read from the {Attribute002} of the selected Asset.

Input XML is read from the {Attribute003} of the selected Asset.

Inbox Inside InTouch OMI

You can use inbox inside **InTouch OMI**.

1. **Complete the** Initial Steps.
2. In the **System Platform IDE**, double-click on the instance of the view app.

   The instance of the view app is opened.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. On the **Toolbox** tab, click **WorkTaskApp**, and then drag and drop the **Inbox** control to a panel in the **Layout**.
4. On the **Properties** tab, map or enter the **Workflow Gateway**.
5. On the **Properties** tab, set the **Default User Always** to **True**, if you want to specify the default user. Otherwise, the user who configured the **Workflow Gateway** object will be considered as the default user.
6. On the **Properties** tab, map or enter the **Default User**, if the **Default User Always** is set to **True**. The user set here will be considered as the user for submission of the work item.
7. On the **Properties** tab, configure other settings as required.
8. Click **Save**.

   The control, along with the set properties, is saved.
9. Click **Preview** to preview the **Inbox** in **InTouch OMI**, if required.
10. In the **System Platform IDE**, right-click on the instance of the view app, and then click **Deploy**.

    The **Deploy** dialog box appears, and shows the progress of the deployment.
11. Click **Close**.

    The **Inbox** can now be used inside **InTouch OMI**.
12. Run **Wonderware Application Manager**, and then click **Launch** for the configured instance of the view app to use the **Inbox**.

Note: When running **Wonderware Application Manager**, if you get an error that "You should be part of either the Administrators, SkeltaAdmins, or SkeltaRemoteUsers role.", then run **Wonderware Application Manager** as an **Administrator**.

If **InTouch OMI** is setup on a different machine, then launching the instance of the view app will lead to an error about unauthorized user. In such cases, configure the **Internet Options** as follows:

1. On the **Security** tab, under **Select a zone to view or change security settings**, click **Trusted sites**.
2. Click **Sites**, in the **Add this website to the zone** box, type the URL of the **Enterprise Console** set for the **Workflow Gateway** object, click **Add**, and then click **Close**.
3. Under **Security level for this zone**, click **Custom level**.
4. Under **Settings**, browse to **User Authentication > Logon** and click **Automatic logon with current username and password**, and then click **OK**.
5. Click **OK** to close **Internet Options**.

List View Inside InTouch OMI

You can use list view inside **InTouch OMI**.

1. **Complete the** Initial Steps.
2. In the **System Platform IDE**, double-click on the instance of the view app.

   The instance of the view app is opened.
3. On the **Toolbox** tab, click **WorkTaskApp**, and then drag and drop the **ListView** control to a panel in the **Layout**.
4. On the **Properties** tab, map or enter the **Workflow Gateway**.
5. On the **Properties** tab, map or enter the **List Name**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. On the **Properties** tab, map or enter the **List View Name**.
7. On the **Properties** tab, map or enter the **Show Ribbon Bar**, if required.
8. On the **Properties** tab, map or enter the **Show Header**, if required.
9. On the **Properties** tab, map or enter the **Show Views**, if required.
10. On the **Properties** tab, set the **Default User Always** to **True**, if you want to specify the default user. Otherwise, the user who configured the **Workflow Gateway** object will be considered as the default user.
11. On the **Properties** tab, map or enter the **Default User**, if the **Default User Always** is set to **True**. The user set here will be considered as the user for submission of the work item.
12. Click **Save**.

    The control, along with the set properties, is saved.
13. Click **Preview** to preview the **List View** in **InTouch OMI**, if required.
14. In the **System Platform IDE**, right-click on the instance of the view app, and then click **Deploy**.

    The **Deploy** dialog box appears, and shows the progress of the deployment.
15. Click **Close**.

    The **List View** can now be used inside **InTouch OMI**.
16. Run **Wonderware Application Manager**, and then click **Launch** for the configured instance of the view app to use the **List View**.

Note:
- If **Security Level** changes are made with respect to the rights given to the user, then the view app should be redeployed before previewing the **List View**.
- When running **Wonderware Application Manager**, if you get an error that “You should be part of either the Administrators, SkeltaAdmins, or SkeltaRemoteUsers role.”, then run **Wonderware Application Manager** as an **Administrator**.

If **InTouch OMI** is setup on a different machine, then launching the instance of the view app will lead to an error about unauthorized user. In such cases, configure the **Internet Options** as follows:

1. On the **Security** tab, under **Select a zone to view or change security settings**, click **Trusted sites**.
2. Click **Sites**, in the **Add this website to the zone** box, type the URL of the **Enterprise Console** set for the **Workflow Gateway** object, click **Add**, and then click **Close**.
3. Under **Security level for this zone**, click **Custom level**.
4. Under **Settings**, browse to **User Authentication > Logon** and click **Automatic logon with current username and password**, and then click **OK**.
5. Click **OK** to close **Internet Options**.

List Form Inside InTouch OMI

You can use list form inside **InTouch OMI**.

1. **Complete the** Initial Steps.
2. In the **System Platform IDE**, double-click on the instance of the view app.

   The instance of the view app is opened.
3. On the **Toolbox** tab, click **WorkTaskApp**, and then drag and drop the **ListForm** control to a panel in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Layout.

4. On the **Properties** tab, map or enter the **Workflow Gateway**.
5. On the **Properties** tab, map or enter the **List Name**.
6. On the **Properties** tab, map or enter the **List Item ID**.
7. On the **Properties** tab, select **Enable Edit**,if you want to enable editing of list item.
8. On the **Properties** tab, set the **Display Alert Message** to **True**, if you want to display an alert message when the form is submitted.
9. On the **Properties** tab, set the **Default User Always** to **True**, if you want to specify the default user. Otherwise, the user who configured the **Workflow Gateway** object will be considered as the default user.
10. On the **Properties** tab, map or enter the **Default User**, if the **Default User Always** is set to **True**. The user set here will be considered as the user for submission of the work item.
11. Click **Save**.

    The control, along with the set properties, is saved.
12. Click **Preview** to preview the **List Form** in **InTouch OMI**, if required.
13. In the **System Platform IDE**, right-click on the instance of the view app, and then click **Deploy**.

    The **Deploy** dialog box appears, and shows the progress of the deployment.
14. Click **Close**.

    The **List Form** can now be used inside **InTouch OMI**.
15. Run **Wonderware Application Manager**, and then click **Launch** for the configured instance of the view app to use the **List Form**.

Note:
- If **Security Level** changes are made with respect to the rights given to the user, then the view app should be redeployed before previewing the **List Form**.
- When running **Wonderware Application Manager**, if you get an error that “You should be part of either the Administrators, SkeltaAdmins, or SkeltaRemoteUsers role.”, then run **Wonderware Application Manager** as an **Administrator**.

If **InTouch OMI** is setup on a different machine, then launching the instance of the view app will lead to an error about unauthorized user. In such cases, configure the **Internet Options** as follows:

1. On the **Security** tab, under **Select a zone to view or change security settings**, click **Trusted sites**.
2. Click **Sites**, in the **Add this website to the zone** box, type the URL of the **Enterprise Console** set for the **Workflow Gateway** object, click **Add**, and then click **Close**.
3. Under **Security level for this zone**, click **Custom level**.
4. Under **Settings**, browse to **User Authentication > Logon** and click **Automatic logon with current username and password**, and then click **OK**.
5. Click **OK** to close **Internet Options**.

BAM Inside InTouch OMI

You can use BAM (Business Activity Monitoring) inside **InTouch OMI**.

1. **Complete the** Initial Steps.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. In the **System Platform IDE**, double-click on the instance of the view app.

   The instance of the view app is opened.
3. On the **Toolbox** tab, click **WorkTaskApp**, and then drag and drop the **BAM** control to a panel in the **Layout**.
4. On the **Properties** tab, map or enter the **Workflow Gateway**.
5. On the **Properties** tab, map or enter the **Report ID**.
6. On the **Properties** tab, map or enter the **Report Name**.
7. On the **Properties** tab, map or enter the **Report Title**.
8. On the **Properties** tab, map or enter the **Report Description**, if required.
9. On the **Properties** tab, map or enter the **Workflow Names**.
10. On the **Properties** tab, select **Display Individual Report**,if you want to show individual report.
11. On the **Properties** tab, set the **Default User Always** to **True**, if you want to specify the default user. Otherwise, the user who configured the **Workflow Gateway** object will be considered as the default user.
12. On the **Properties** tab, map or enter the **Default User**, if the **Default User Always** is set to **True**. The user set here will be considered as the user for submission of the work item.
13. Click **Save**.

    The control, along with the set properties, is saved.
14. Click **Preview** to preview the **BAM** in **InTouch OMI**, if required.
15. In the **System Platform IDE**, right-click on the instance of the view app, and then click **Deploy**.

    The **Deploy** dialog box appears, and shows the progress of the deployment.
16. Click **Close**.

    The **BAM** can now be used inside **InTouch OMI**.
17. Run **Wonderware Application Manager**, and then click **Launch** for the configured instance of the view app to use the **BAM**.

Note:
- If **Security Level** changes are made with respect to the rights given to the user, then the view app should be redeployed before previewing the **BAM**.
- When running **Wonderware Application Manager**, if you get an error that “You should be part of either the Administrators, SkeltaAdmins, or SkeltaRemoteUsers role.”, then run **Wonderware Application Manager** as an **Administrator**.

If **InTouch OMI** is setup on a different machine, then launching the instance of the view app will lead to an error about unauthorized user. In such cases, configure the **Internet Options** as follows:

1. On the **Security** tab, under **Select a zone to view or change security settings**, click **Trusted sites**.
2. Click **Sites**, in the **Add this website to the zone** box, type the URL of the **Enterprise Console** set for the **Workflow Gateway** object, click **Add**, and then click **Close**.
3. Under **Security level for this zone**, click **Custom level**.
4. Under **Settings**, browse to **User Authentication > Logon** and click **Automatic logon with current username and password**, and then click **OK**.
5. Click **OK** to close **Internet Options**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Workflow Status Inside InTouch OMI

You can use Workflow Status inside **InTouch OMI**.

1. **Complete the** Initial Steps.
2. In the **System Platform IDE**, double-click on the instance of the view app.

   The instance of the view app is opened.
3. On the **Toolbox** tab, click **WorkTaskApp**, and then drag and drop the **WorkflowStatus** control to a panel in the **Layout**.
4. On the **Properties** tab, map or enter the **Workflow Gateway**.
5. On the **Properties** tab, map or enter the **Workflow Execution ID**.
6. On the **Properties** tab, select **Enable Workflow Controls**, if you want to enable controlling of the workflow, if required.
7. On the **Properties** tab, select **Display Analyst View Only**, if you want to show only the Analyst's view, if required.
8. On the **Properties** tab, set the **Default User Always** to **True**, if you want to specify the default user. Otherwise, the user who configured the **Workflow Gateway** object will be considered as the default user.
9. On the **Properties** tab, map or enter the **Default User**, if the **Default User Always** is set to **True**. The user set here will be considered as the user for submission of the work item.
10. Click **Save**.

    The control, along with the set properties, is saved.
11. Click **Preview** to preview the **Workflow Status** in **InTouch OMI**, if required.
12. In the **System Platform IDE**, right-click on the instance of the view app, and then click **Deploy**.

    The **Deploy** dialog box appears, and shows the progress of the deployment.
13. Click **Close**.

    The **Workflow Status** can now be used inside **InTouch OMI**.
14. Run **Wonderware Application Manager**, and then click **Launch** for the configured instance of the view app to use the **Workflow Status**.

Note:
- If **Security Level** changes are made with respect to the rights given to the user, then the view app should be redeployed before previewing the **Workflow Status**.
- When running **Wonderware Application Manager**, if you get an error that “You should be part of either the Administrators, SkeltaAdmins, or SkeltaRemoteUsers role.”, then run **Wonderware Application Manager** as an **Administrator**.

If **InTouch OMI** is setup on a different machine, then launching the instance of the view app will lead to an error about unauthorized user. In such cases, configure the **Internet Options** as follows:

1. On the **Security** tab, under **Select a zone to view or change security settings**, click **Trusted sites**.
2. Click **Sites**, in the **Add this website to the zone** box, type the URL of the **Enterprise Console** set for the **Workflow Gateway** object, click **Add**, and then click **Close**.
3. Under **Security level for this zone**, click **Custom level**.
4. Under **Settings**, browse to **User Authentication > Logon** and click **Automatic logon with current username**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

and password, and then click **OK**.

5. Click **OK** to close **Internet Options**.

Inbox Work Items Count Inside InTouch OMI

You can use Inbox Work Items Count inside **InTouch OMI**.

1. **Complete the** Initial Steps.
2. In the **System Platform IDE**, double-click on the instance of the view app.

   The instance of the view app is opened.
3. On the **Toolbox** tab, click **WorkTaskApp**, and then drag and drop the **InboxWorkItemsCount** control to a panel in the **Layout**.
4. On the **Properties** tab, map or enter the **Workflow Gateway**.
5. On the **Properties** tab, map or enter the **Workflow Name**.
6. On the **Properties** tab, select **Enable Automatic Refresh**, if you want to enable refreshing of work items count automatically, if required.
7. On the **Properties** tab, map or enter the **Refresh Period**.

   Note: If **Refresh Period** is set to less than **30000** ms, then the value is automatically set to the minimum value of **30000** ms.
8. On the **Properties** tab, select **Display Refresh Button**, if you want to show the Refresh button.
9. On the **Properties** tab, map or enter the **Tooltip** for the Refresh button.
10. On the **Properties** tab, set the **Default User Always** to **True**, if you want to specify the default user. Otherwise, the user who configured the **Workflow Gateway** object will be considered as the default user.
11. On the **Properties** tab, map or enter the **Default User**, if the **Default User Always** is set to **True**. The user set here will be considered as the user for submission of the work item.
12. Click **Save**.

    The control, along with the set properties, is saved.
13. Click **Preview** to preview the **Inbox Work Items Count** in **InTouch OMI**, if required.
14. In the **System Platform IDE**, right-click on the instance of the view app, and then click **Deploy**.

    The **Deploy** dialog box appears, and shows the progress of the deployment.
15. Click **Close**.

    The **Inbox Work Items Count** can now be used inside **InTouch OMI**.
16. Run **Wonderware Application Manager**, and then click **Launch** for the configured instance of the view app to use the **Inbox Work Items Count**.

Note:
- If **Security Level** changes are made with respect to the rights given to the user, then the view app should be redeployed before previewing the **Inbox Work Items Count**.
- When running **Wonderware Application Manager**, if you get an error that “You should be part of either the Administrators, SkeltaAdmins, or SkeltaRemoteUsers role.”, then run **Wonderware Application Manager** as an **Administrator**.

If **InTouch OMI** is setup on a different machine, then launching the instance of the view app will lead to an error

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

about unauthorized user. In such cases, configure the **Internet Options** as follows:

1. On the **Security** tab, under **Select a zone to view or change security settings**, click **Trusted sites**.
2. Click **Sites**, in the **Add this website to the zone** box, type the URL of the **Enterprise Console** set for the **Workflow Gateway** object, click **Add**, and then click **Close**.
3. Under **Security level for this zone**, click **Custom level**.
4. Under **Settings**, browse to **User Authentication > Logon** and click **Automatic logon with current username and password**, and then click **OK**.
5. Click **OK** to close **Internet Options**.

Cryptography Settings Inside InTouch OMI

You can import the cryptography settings from a server where AVEVA Work Tasks is installed to this server.

To Import Cryptography Settings

1. Open InTouch OMI application, and then click **Launch**.

   The **AVEVA Work Tasks Cryptography Settings** screen appears.

   Note: The Cryptography Settings screen is available if there are AVEVA Work Tasks controls in InTouch OMI and you are importing the Cryptography settings for the first time. Later to update the settings, you need to place the **.settings** file in the InTouch OMI user's ViewApp Instance folder i.e., *C:\Users\<user name>\AppData\Local\ViewApps\<viewapp instance>\Controls\AVEVAWorkTasksApp001*
2. On the **Cryptography Settings** page, do the following:
   a. Enter the passphrase for the settings.

      Note: Enter the same passphrase that you configured in AVEVA Work Tasks for Cryptography Settings using Farm Configuration Wizard.
   b. Click **Import** to import the cryptography settings file in the server.
3. Click **Finish**.

   The cryptography settings get imported to the InTouch OMI server.

Historian Server

You can monitor multiple and specific Historian Event tags using the Historian server. You can also trigger the Workflow from System Platform based on an action in the Historian Server.

Trigger Workflow from the Historian Server

You can trigger the Workflow from System Platform based on an action in the Historian Server.

Prerequisites

- ArchestrA should be installed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Gateway object should be imported.
- AppEngine process owner should have Registry read permissions and file read-write permissions.

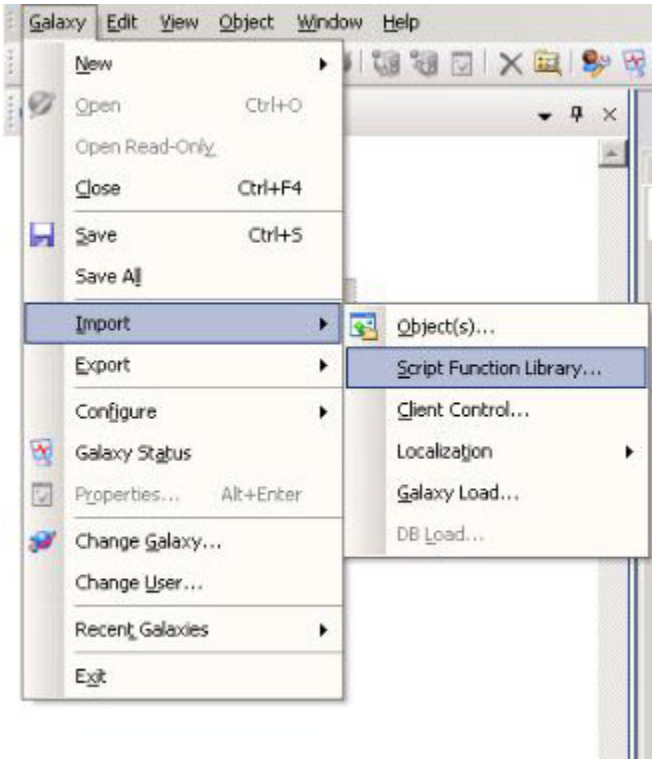
Historian

1. Creating Historian Event tag

   You can create a new Historian event tag or use an existing Historian event tag for triggering System Platform workflow. For creating the Historian event tag, see the Historian documentation available in Start > Programs > Wonderware Historian folder.

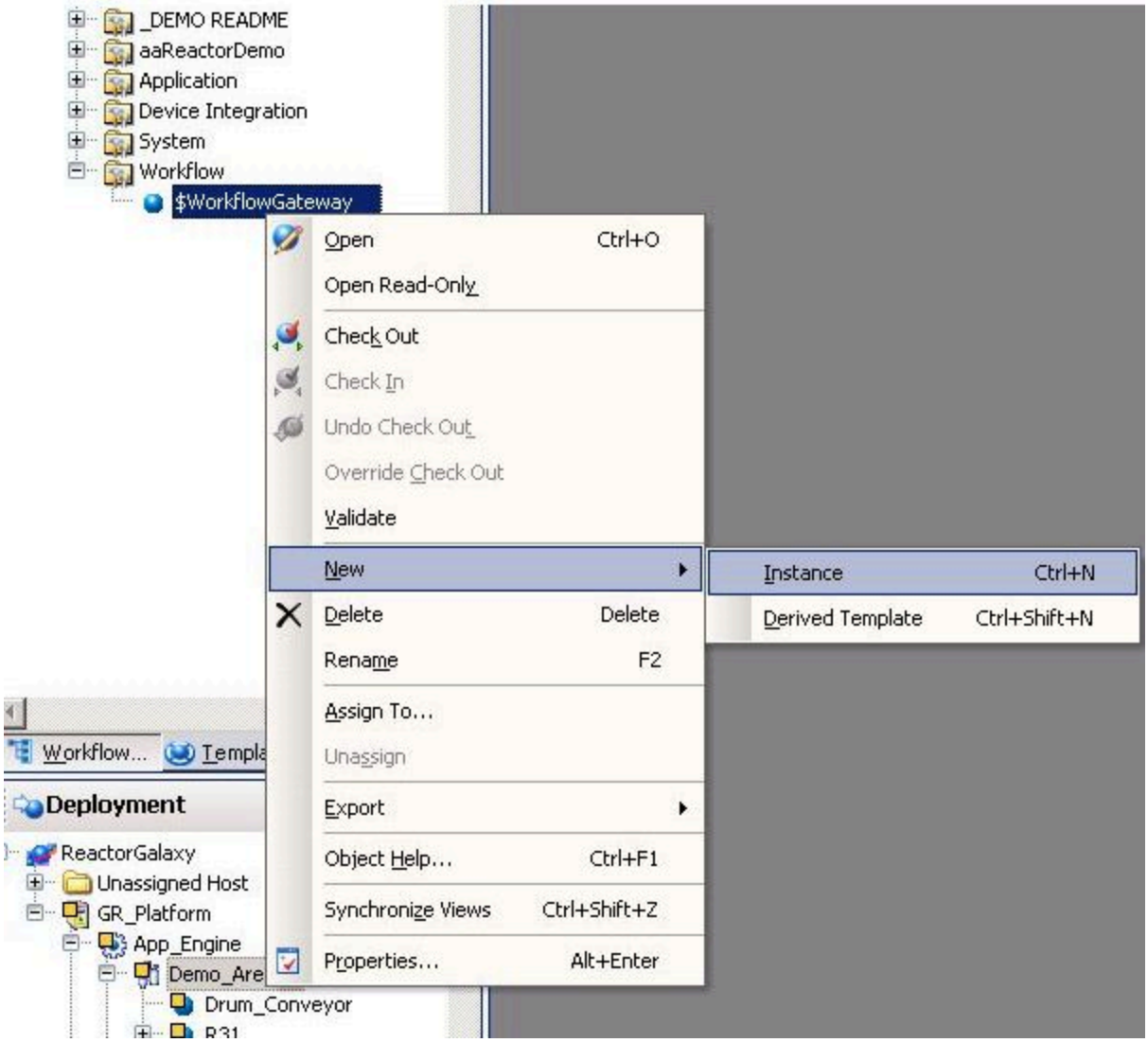
2. System Platform IDE: Importing Script Function Library

   Import the script library (Historian.EventScript.dll) from <A2 Install Path>\Workflow\ScriptLibrary



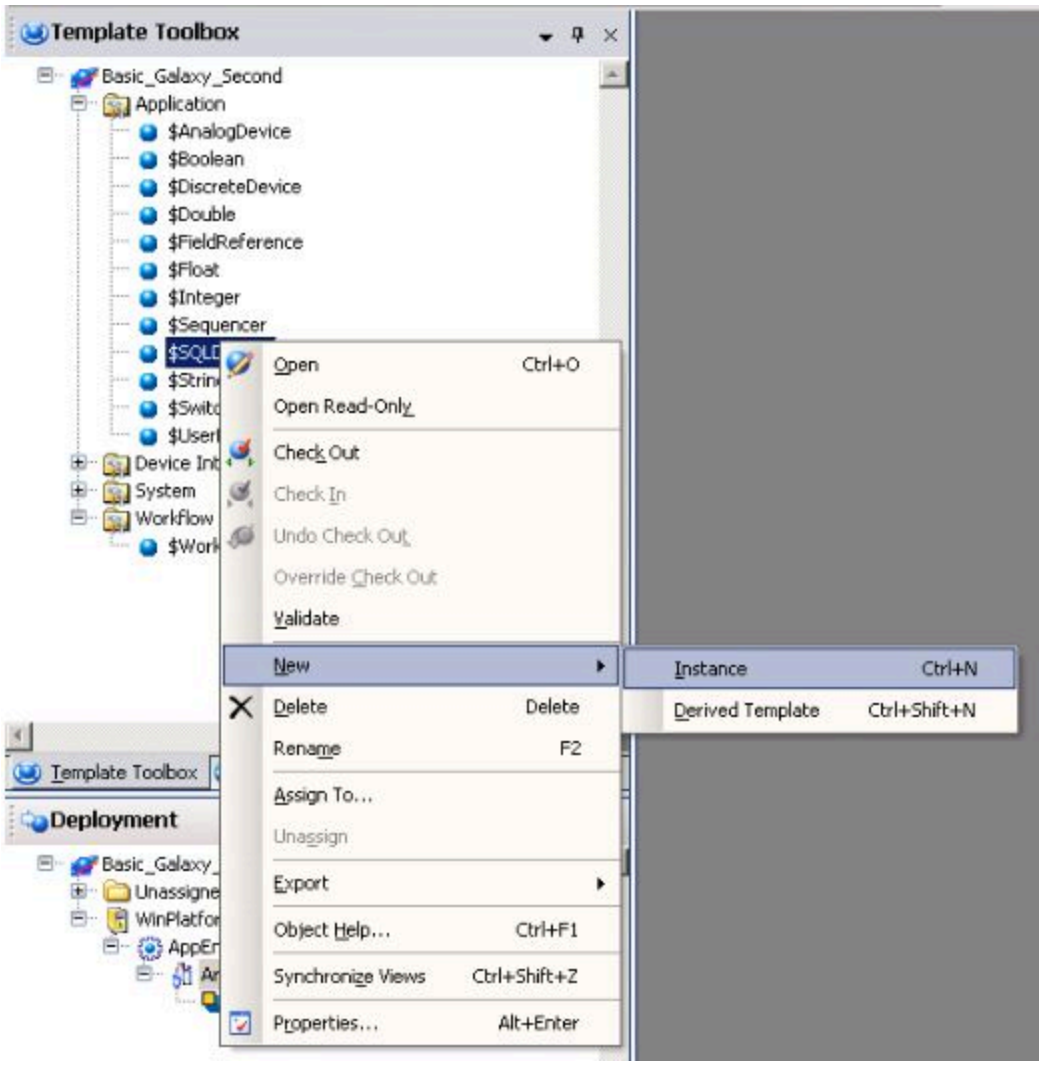
3. Creating Gateway object: Create a new instance. Right-click $WorkflowGateway, point to New, and select Instance. An instance with name WorkflowGateway_001 is created.

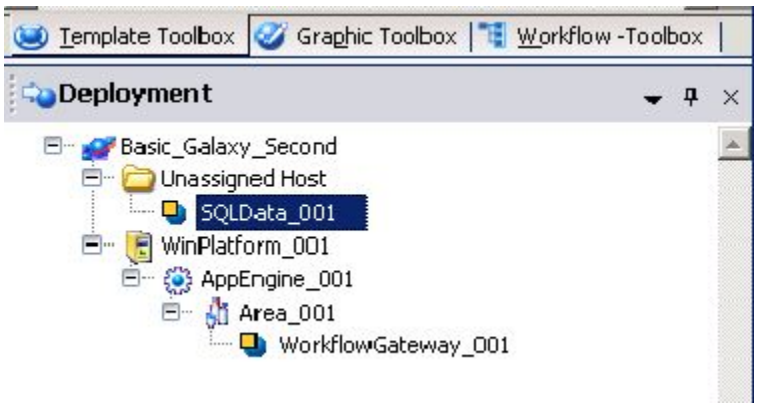
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



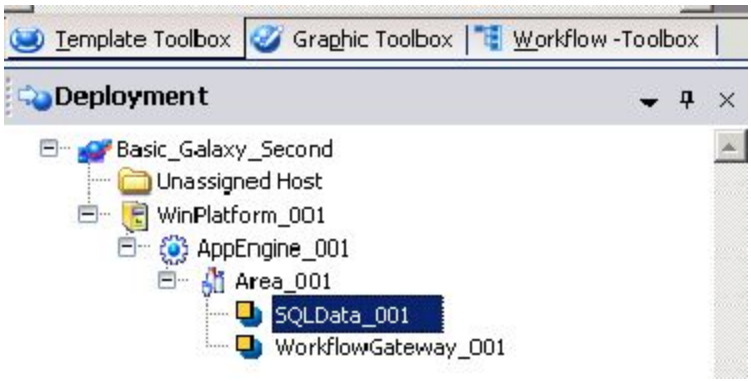
4. Assign this instance to the proper AppEngine and area.
5. Next, configure the Gateway object, as described in Initial Gateway Setup.
6. Next, you can create an instance of the app object from a template, such as the SQL Data template.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.





© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Monitor Multiple Historian Event Tag

Follow steps 1-7 listed in Trigger Workflow from the Historian Server and continue:

1. Next, you need to create a user-defined attribute (UDA) (by clicking the + sign) for each Historian event tag to trigger a workflow for a Historian event occurrence. The category for the UDA should be User Writeable.
2. Write a script to get the triggered event information from Historian.

   You need to run the script in asynchronous mode. Select the Runs Asynchronously check box. The sample script is given below:

```
dim dataTables as System.Data.DataTable;
'Add UDA for each Event Tag; Make all the values of attributes as false here
SQLData_001.On_SysStatusEvent_Trigger = false;
dataTables =
Historian.EventScript.HistorianEvent.GetTriggeredEvents("HistorianDatabaseServerName"
, "DatabaseName", true, "", "","GalaxyName", "ObjectName", me.Last_Lookup_Key);
If(dataTables <> null)Then
'Add if else to check each tag;
if(dataTables.Rows[0]["TagName"] == "SysStatusEvent")Then
'Make the respective attribute value as true if the event tag name matches;
SQLData_001.On_SysStatusEvent_Trigger = true;
EndIF;
if(dataTables.Rows[0]["TagName"] == "ExternalEvent")Then
'Make the respective attribute value as true if the event tag name matches;
SQLData_001.On_ExternalEvent_Trigger = true;
EndIF;
me.Last_Lookup_Key = dataTables.Rows[0]["EventLogKey"]
Endif;
dataTables = null;
```

Explanation of the script:

- The Historian.EventScript.dll has one public static method named as GetTriggeredEvents. The script above shows how to invoke the GetTriggeredEvents method.
- The GetTriggeredEvents method return type is System.Data.DataTable. The data type will have two columns, EventLogKey and TagName.
- Last_Lookup_Key is the value of the last EventLog fetched from the Historian database. If the primary server fails for any reason, the backup server will automatically start monitoring the Historian server using the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Last_Lookup_Key value.

- The GetTriggeredEvents method has seven parameters, as described below:

Parameter Name	Datatype
serverName	String
databaseName	String
isWindowsAuthentication	Boolean
userId	String
password	String
galaxyName	String
objectName	String
lastProcessedNumberRead	Integer

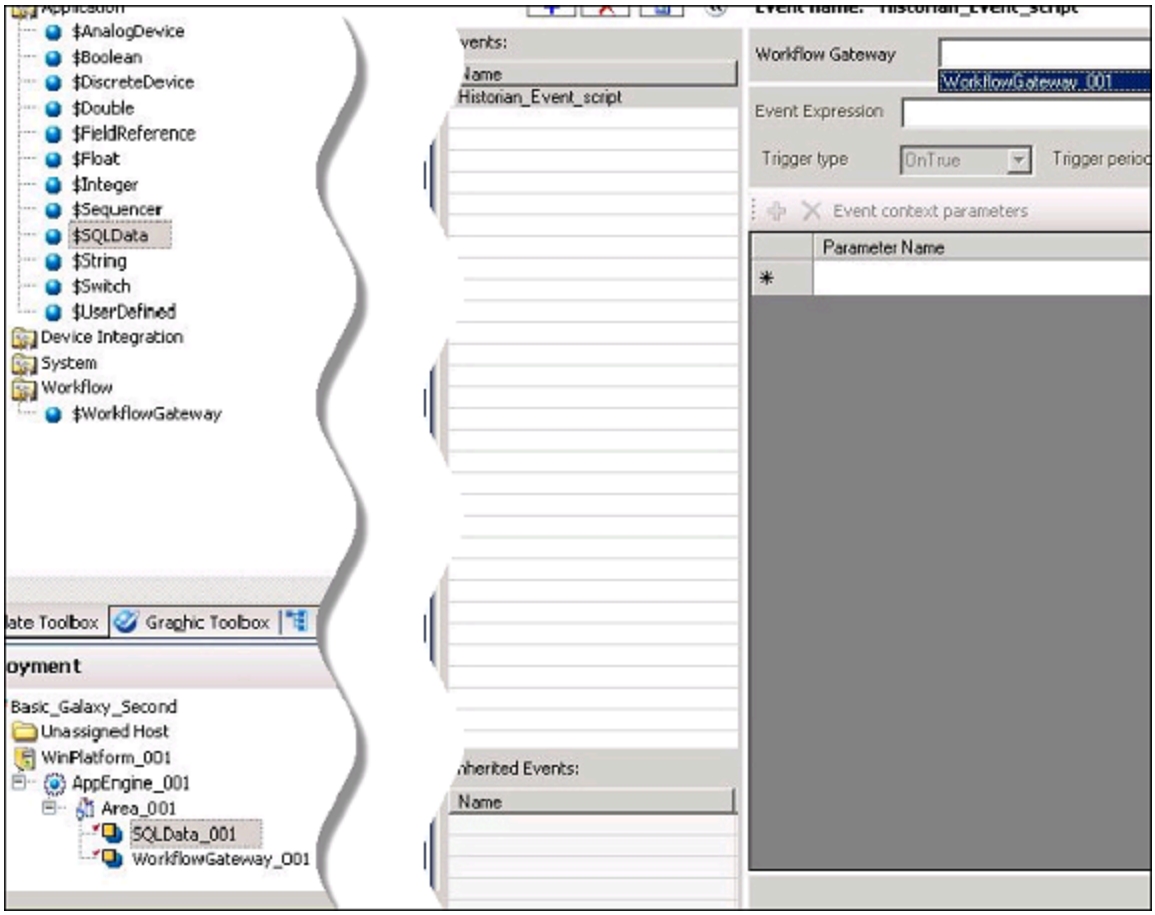
Ensure to set the SQLData_001.On_SysStatusEvent_Trigger to 'false' at the beginning of the script.

Associating Workflow Event with System Platform Workflow

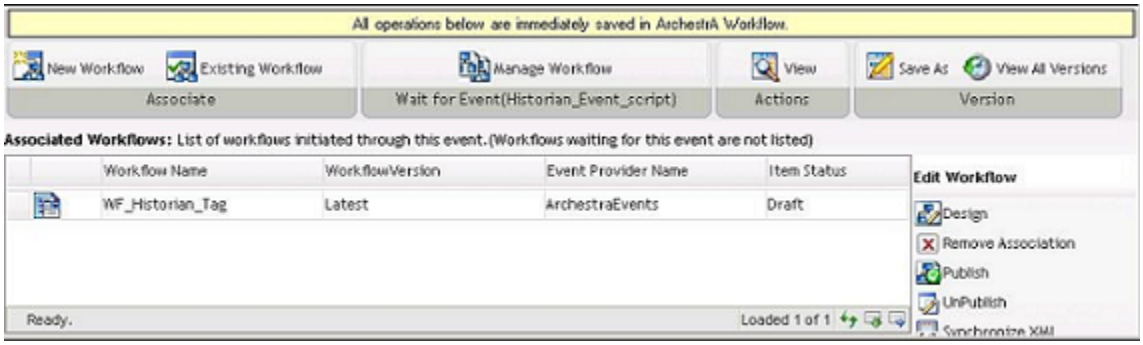
1. Create a workflow event to attach the workflow. Select the Workflows Tab of SQLData_001.

   For event association details, refer to Associate an Existing Workflow to a Workflow Event.
2. Click the + sign to add an event, Historian_Event_script. Select WorkflowGateway_001 from the Workflow Gateway drop-down list and click Select.

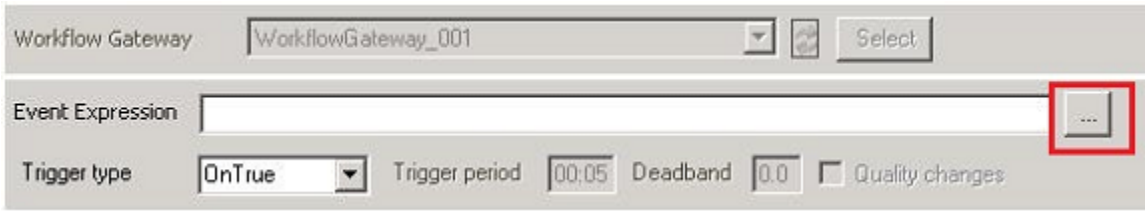
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. The Associated Workflows section appears. You can create a new workflow or use an existing workflow to associate with this event.
4. After you add the workflow, it appears in the Associated Workflows list.



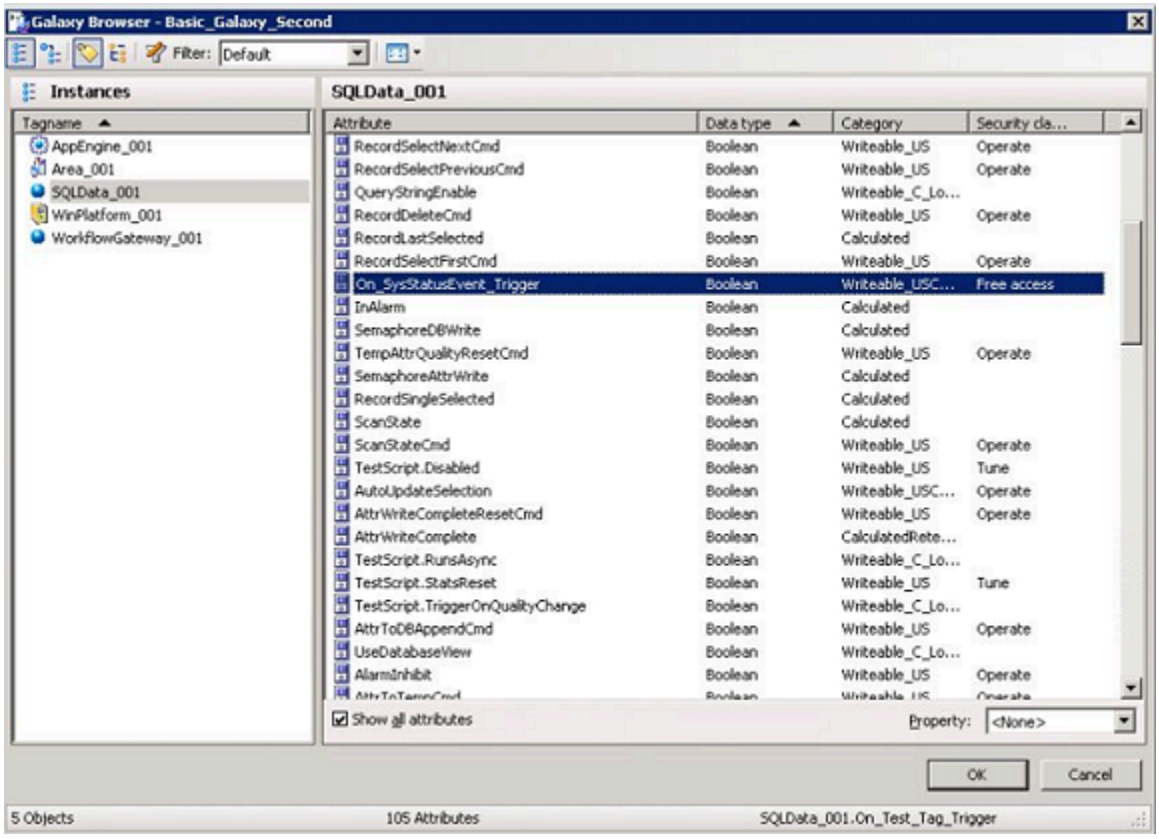
5. Next, you need to set the event expression. Click the Expression Editor button on the right side. The workflow gets triggered depending on the Trigger Type selected.



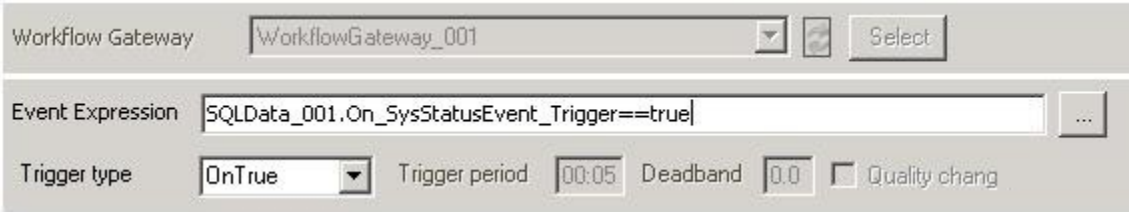
The Galaxy browser appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Select the SQLData_001, On_SysStatusEvent_Trigger attribute and click OK.

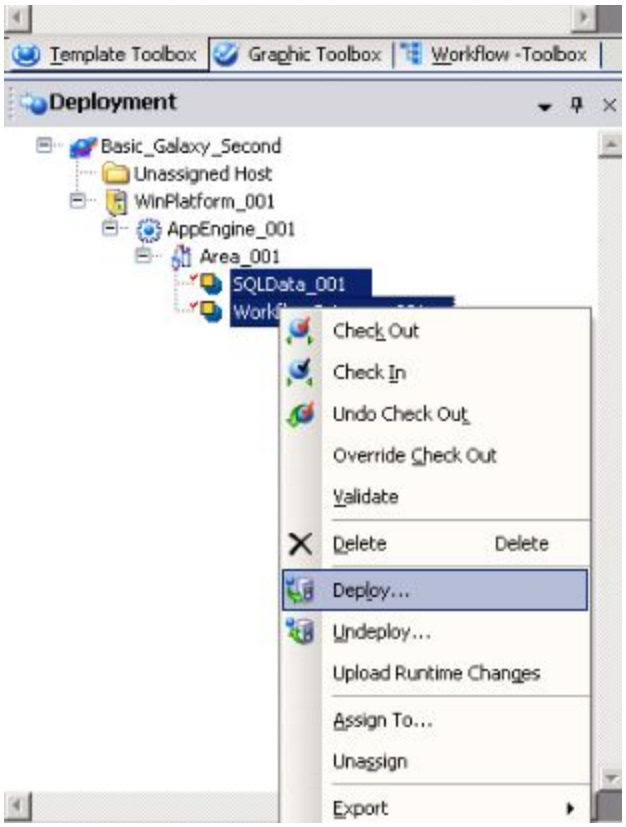


7. The SQLData_001.On_SysStatusEvent_Trigger attribute appears in the EventExpress (OnTrue) box. Add ?==true? after the attribute name and click the Save button.



8. Next, deploy the objects by right-clicking on the SQLData_001 and WorkflowGateway_001 objects and selecting Deploy.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Continue to monitor the EventHistory table of the Historian database. If any matching tags are inserted, the WF_Historian_Tag workflow will be fired.

Monitor Specific Historian Event Tag

1. Follow the steps listed in Trigger Workflow from the Historian Server.
2. Next, you need to create a user-defined attribute (UDA) (by clicking the + sign) for each Historian event tag to trigger a workflow for a Historian event occurrence. The category for the UDA should be User Writeable.
3. Next, write a script to get the triggered event information from Historian.

   You need to run the script in asynchronous mode. Select the Runs Asynchronously check box. Sample script is given below:

```
dim dataTables as System.Data.DataTable;
'Add UDA for each Event Tag; Make all the values of attributes as false here
SQLData_001.On_SysStatusEvent_Trigger = false;
dataTables =
Historian.EventScript.HistorianEvent.GetTriggeredEventsForSpecificTag("HistorianDatab
aseServerName", "DatabaseName", true, "", "","GalaxyName", "ObjectName",
"SysStatusEvent", me.Last_Lookup_Key);
If(dataTables <> null)Then
SQLData_001.On_SysStatusEvent_Trigger = true;
me.Last_Lookup_Key = dataTables.Rows[0]["EventLogKey"]
Endif;
dataTables = null;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Explanation of the script:

- The Historian.EventScript.dll has one public static method named as GetTriggeredEventsForSpecificTag. The script above shows how to invoke the GetTriggeredEvents method.
- The GetTriggeredEvents method return type is System.Data.DataTable. The data type will have two columns, EventLogKey and TagName.
- Last_Lookup_Key is the value of the last EventLog fetched from the Historian database. If the primary server fails for any reason, the backup server will automatically start monitoring the Historian server using the Last_Lookup_Key value.
- The GetTriggeredEvents method has seven parameters, as described below:

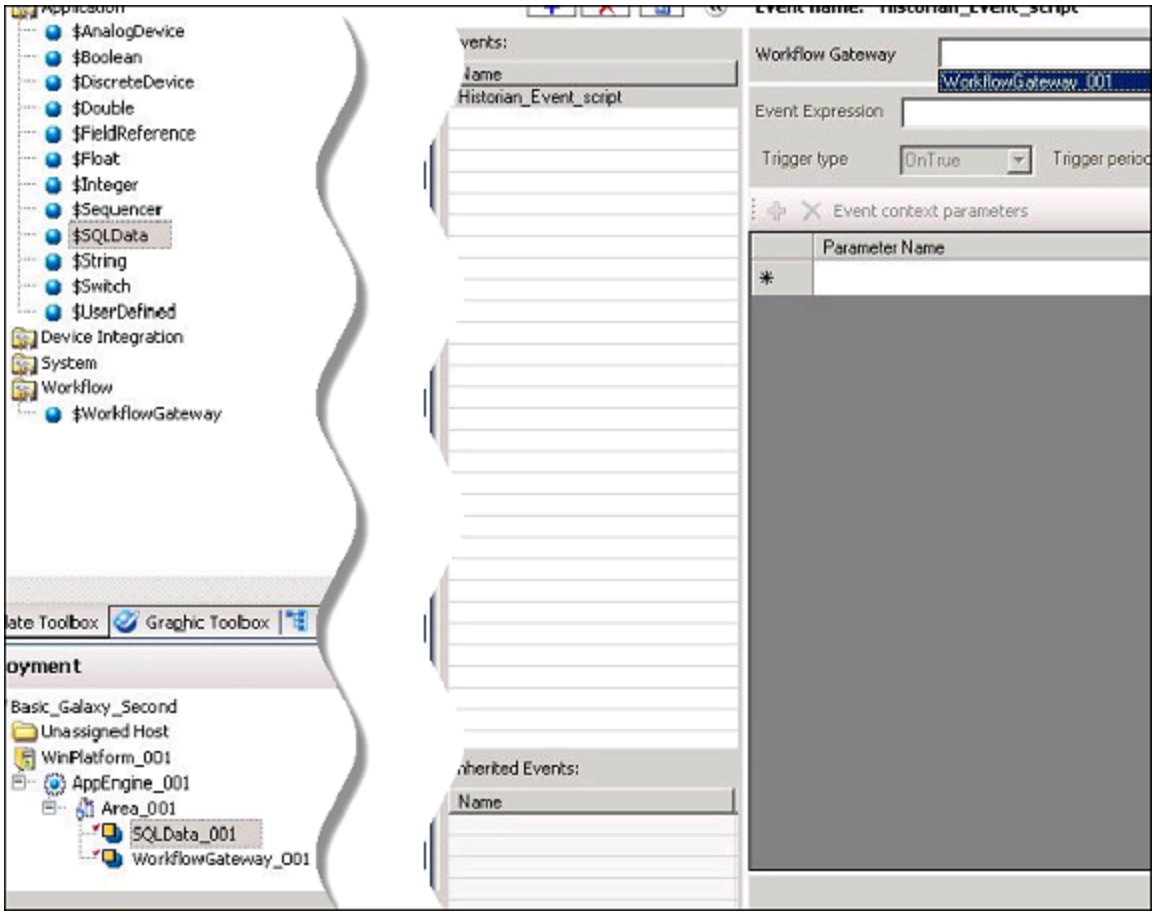
Parameter Name	Datatype
serverName	String
databaseName	String
isWindowsAuthentication	Boolean
userId	String
passWord	String
galaxyName	String
objectName	String
tagName	String
lastProcessedNumberRead	Integer

Ensure to set the SQLData_001.On_SysStatusEvent_Trigger to false at the beginning of the script.

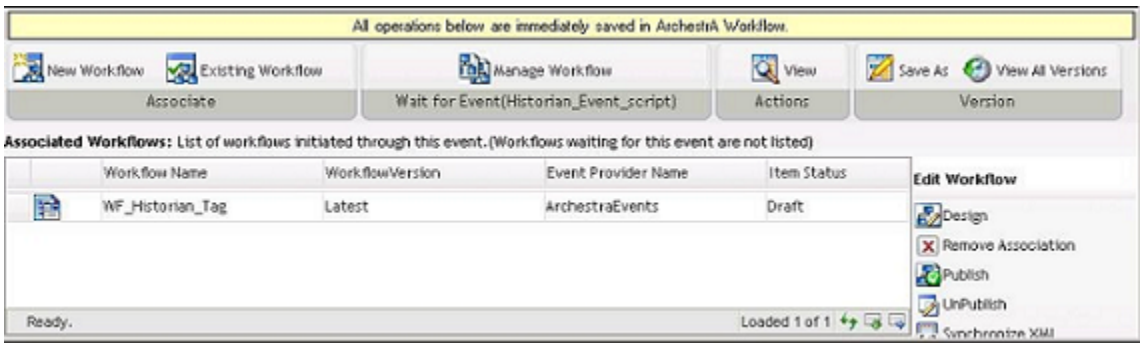
Associating Workflow Event with System Platform Workflow

1. Next, you create a workflow event to attach the workflow. Select the Workflows tab of SQLData_001.
   For event association details, refer to Associate an Existing Workflow to a Workflow Event.
2. Click the + sign to add an event, Historian_Event_script. Select WorkflowGateway_001 from the Workflow Gateway drop-down list and click Select.

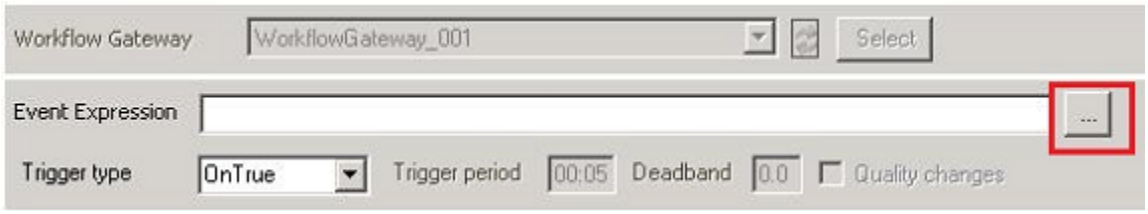
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



1. The Associated Workflows section appears. You can create a new workflow or use an existing workflow to associate with this event.
2. After you add the workflow, it appears in the Associated Workflows list.



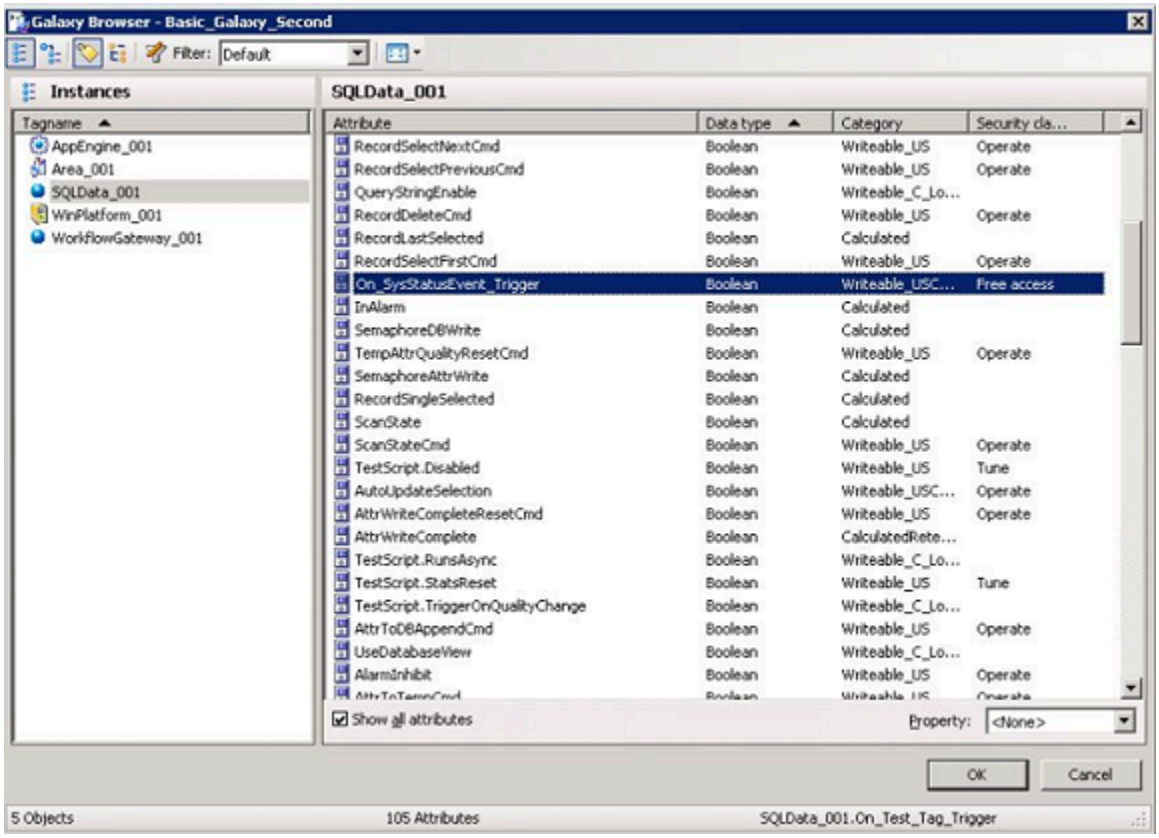
3. Next, you need to set the event expression. Click the Expression Editor button on the right side. The workflow gets triggered depending on the Trigger Type selected.



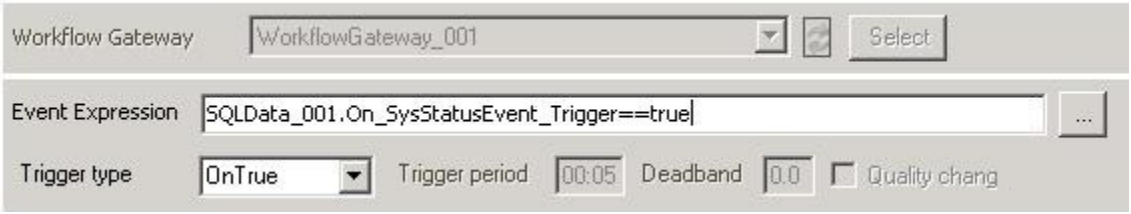
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Galaxy Browser appears.

4. Select SQLData_001, On_SysStatusEvent_Trigger attribute and click OK.

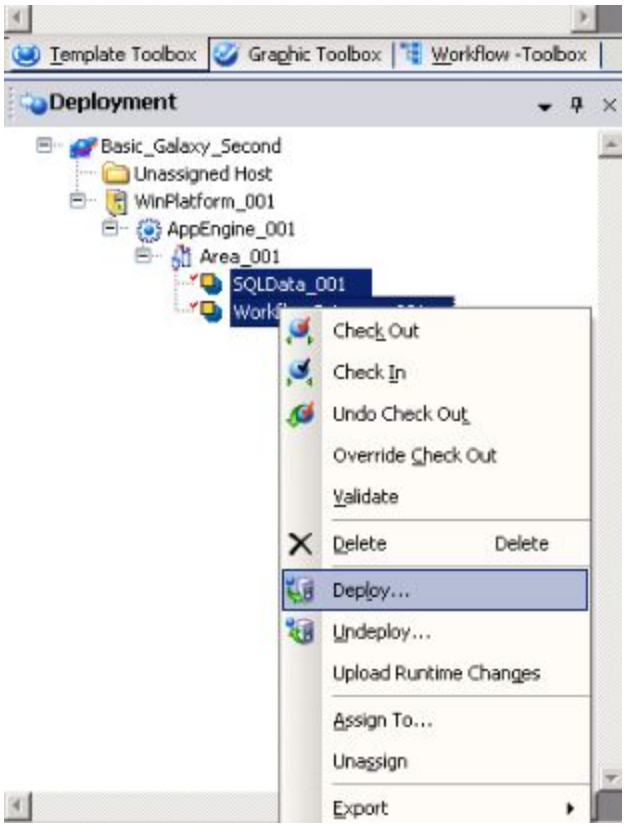


5. The SQLData_001.On_SysStatusEvent_Trigger attribute appears in the EventExpress (OnTrue) box. Add ?==true? after the attribute name and click the Save button.



6. Next, deploy the objects by right-clicking on the SQLData_001 and WorkflowGateway_001 objects and selecting Deploy.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Continue to monitor the EventHistory table of the Historian database. If any matching tags are inserted, the WF_Historian_Tag workflow will be fired.

Retrieve Historian Data Using Database Activity

AVEVA Work Tasks has a powerful feature where it can connect to and retrieve data from Wonderware Historian Server using an AVEVA Work Tasks Database Activity. You can use this functionality in many scenarios, a few of which are described below:

Your network administrator needs data on the number of system status event tags that have been fired in a particular time-frame, which can then be used to optimize system performance.

Using the Database Activity, you can fetch data from the Historian database on how many SysStatusEvent event tags have been triggered in the last one week.

How you go about it:

Use the queries shown below in the database activity to get the desired output.

Sample Script 1: This script gets the Event Trigger history for a specific EventTag for a specific time-span.

```
--Get Event Trigger history for a particular EventTag for the last 10 days
SELECT isnull((select HierarchicalName from
dbo.faaGetHierarchicalAttributeNames(Tag.TagName)),Tag.TagName)as HierarchicalName ,
Description, DateTime = convert(nvarchar, DateAdd(mi,60,DateTime), 21), DetectDateTime =
convert(nvarchar, DateAdd(mi,60,DetectDateTime), 21)
FROM EventHistory
INNER JOIN Tag ON Tag.TagName = EventHistory.TagName
WHERE Tag.TagName IN ('SysStatusEvent')
AND DateTime >= DateAdd(DAY,-10,GetDate())
AND DateTime <= GetDate()
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Samples Script 2: This script gets all the Event Trigger history for all EventTags for a specific time-span.

```
--Get All Event Trigger history for last 10 days
SELECT isnull((select HierarchicalName from
dbo.faaGetHierarchicalAttributeNames(Tag.TagName)),Tag.TagName)as HierarchicalName ,
Description, DateTime = convert(nvarchar, DateAdd(mi,60,DateTime), 21), DetectDateTime =
convert(nvarchar, DateAdd(mi,60,DetectDateTime), 21)
FROM EventHistory
INNER JOIN Tag ON Tag.TagName = EventHistory.TagName
WHERE DateTime >= DateAdd(DAY,-10,GetDate())
AND DateTime <= GetDate()
```

Samples Script 3: This script gets details of all Analog tags.

```
--Get detail of all Analog Tag
SELECT isnull((select HierarchicalName from
dbo.faaGetHierarchicalAttributeNames(Tag.TagName)) ,Tag.TagName)as HierarchicalName ,
Description = Tag.Description, MinRaw, MaxRaw, Unit, MinEU, MaxEU, StorageRate,
StorageType, DateCreated
FROM AnalogTag, Tag, EngineeringUnit
WHERE Tag.TagName = AnalogTag.TagName
AND AnalogTag.EUKey = EngineeringUnit.EUKey
```

To create Database connection object in the Workflow

1. Open the Database Connections object.
2. Click on New Connection.
3. Enter title, description and uncheck the "Connect to repository data source" check box
4. Database Type will be SQL server as the data store of Historian is Microsoft SQL server.
5. Keep the Connection Type as Connection string.
6. Mention the server name (Server name of Historian Database).
7. Select the Authentication type.

   If Authentication type is "SQL server", then mention User Id, Password and Advance Settings.
8. Click the Refresh buttons to get all the databases in the specified server.
9. Select runtime for historian data.
10. Click Test Connection.

    If the connection succeeds, the Save button will be enabled.
11. Click the Save button.

    The connection string is stored.

To add the Database Activity

Database Activity is an activity inside the workflow wherein the workflow actually establishes the connection to the database server and performs actions to retrieve or update data in the database. The activity helps to execute Queries, Stored Procedure and Functions in workflows with ease.

The database activity can execute your queries, stored procedures and functions and carry the result in an XML variable where the schema will load the columns of the table in your database and will contain the value as per the given query.

For Insert and Update statements, we need not give the XML Variable Name since it does not return any value.

1. Hover the cursor on Enterprise Console and click on Workflows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Workflow window will open.

2. Click on the Workflow ribbon in the ribbon bar.
3. Enter the workflow name and description.
4. Click on Save and Continue. A window appears.
5. Click on Finish.

   (Design the Workflow Now option should be checked before clicking on Finish: this is a default selection).

   Workflow Design Editor will appear.
6. Add the database activity to the Workflow design.
7. Go to the properties of the Database Activity.
8. Change the Activity Name, if required. Add the Description, if required.
9. Select the database connection from the drop down and click on Save button.
10. Click on the Query Builder icon.

Query Builder

1. Select Command Type Query
2. Enter Query, you can use the same query mentioned I the scenario section.
3. Click on the Load Parameters button. It will load all the parameters below.
4. Enter the value of the parameter.
5. Enter the output XML variable name.
6. Save the information by clicking on the Save button.
7. If you want you can click on View Schema button to view the schema of the HistorianDataLoad variable as displaying in the below screen.
8. Click on Close button to close the view schema window.

Note: Multiple queries are not supported. If multiple queries are used, XML variable will carry the result of the last query written.

To log data in Console logger:

Close the query window and back to workflow design editor.

1. Drag and Drop "Logger" from Engine Activity. Right click on the logger image and click on property.
2. In Activity property, modify name if need. Add description if required.
3. Select "i – Information" as LogType.
4. Click on the right-most button to open the LogMessage property page.
5. Select XML Variable as Field.
6. The drop-down list will be populated on selecting the drop-down field.
7. Select "HistorianDataLoad" in the drop-down list.
8. Select Table in the last drop-down list and click on Append button.
9. After append, click on the Update button of the page.
10. On clicking Update, this page will be closed and the workflow design page will be on focus. Now we are done with the Workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

11. Click TestRun. Before clicking "TestRun", open the Logger Console.
12. If it succeeds, then check the Logger console whether all expected rows have been logged or not.

Information Server

You can configure workflow Repository and Farm for Information server. You can also configure workflow server to add the Repository details.

Configure Repository Using AVEVA Work Tasks Connector for Information Server

Configure Repository Using AVEVA Work Tasks Connector for Information Server

- Provide connectivity from Information Server to the Workflow Repository.
- Enable Information Server users to access the Workflow Repository functionality from within Information Server.
- Enable access to Central Configuration where users can create or edit workflow repositories, and use its functionalities.

Configure Workflow Repository and Farm for Information Server

To configure Database Credentials for Information Server Database

1. Run the AVEVA Work Tasks Connector for Information Server application (Programs > AVEVA> AVEVA Work Tasks > Information Server > AVEVA Work Tasks Connector for Information Server).

   The Database Server Name and Database Name appears.

   - Database Server Name: Displays the Information Server Database Server name. This is auto-populated from the Registry and is Read-only.
   - Database Name: Displays the Information Server Database name. This is Read-only.
2. Select the Windows radio button to test the connection using the Windows authentication mode. You need not type User ID or Password, if you select this option.

   Or

   Select the SQL Server radio button to test the connection using the SQL Server authentication mode. Type "User Id" or "Password", if you select this option.

   Enter the "SQL Server" username in User Id.

   Enter the "SQL Server" password in Password.

   The credentials should have row insertion and deletion permission for the SuiteVoyager database.
3. Click the Test Connection button.
   - If the connection succeeds, the Next button will be enabled.
   - If the connection fails, you have to re-enter the details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure Workflow Server to Add Repository Details

To configure the Workflow Server to Add the Repository Details for AVEVA Work Tasks

1. Enter the **Unique Environment Name**.

   The Unique Environment Name, by default, is a combination of the machine name and Information Server, and is editable the first time. After repository configuration is done, it cannot be edited.
2. Enter the Workflow server address.
3. Enter the AVEVA Work Tasks user domain.
4. Enter the user ID of the AVEVA Work Tasks user.
5. Enter the AVEVA Work Tasks user password.
6. To check if authentication is successful, and to see the list of repositories available, click **Test Connection**.

   If the authentication is successful, the **Repository Name** field is populated with repositories available.

   If the connection fails, you need to re-enter the details.
7. Select the repository name from the **Repository Name** drop-down list.

   The Provisioning screen appears.
8. Select the provider from the drop-down list, username and password, depending on provider selection.
9. Click **Authorize** to enable provisioning.

   The Workflow Server and Repository Details screen appears.
10. Click **Connect** to configure the repository for WIS.

Manage Connected Workflow Repositories

The Manage Connected Workflow Repositories screen appears.

Next, you can add or delete workflow repositories by using the Add New or Delete buttons respectively.

When you connect, this repository is listed in WIS under Workflow Repositories node. Also, the Central Configuration link of the Repository is created under the Workflow Farm node.

Administrative Activity

The administrator can perform three activities to customize the workflow repositories and workflow farm links to the specific users:

- Add Custom Panel
- Add User Permissions
- Perform Workflow Repository Functionality Within Information Server

Add Custom Panel

1. Select **Administration > Panel Manager > New Panel**.
2. Add the Panel name and click anywhere in the right pane.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Edit**, **Delete**, and **Rename** options appear. You can edit, delete and rename the panel, using these options.

3. To configure the options, click **Edit**.
4. Drag and drop activities into the Custom Panel.
5. Click **Users** to add users.

   The updated Custom Panel shows the new custom panel configuration.
6. Click **Save Panel Content** to save the Custom Panel.

   A message appears indicating that the panel tree information is saved successfully.

Add User Permissions

1. Click **Users** under **Panel Content** to add users for the custom panel that you created.
2. Under **Panel User Access**, select the required domain from the **Domain** drop-down list.
3. Click **Retrieve Users from Selected Domain**.

   The list of user names is displayed under **Domain Users**.
4. To add a user, select the user name and click **Add**. You can add multiple users.

   You can also remove users by clicking the **Remove** button.
5. Click **Save Panel Content**.

   A message appears indicating that the panel tree information is saved successfully.

Perform Workflow Repository Functionality Within Information Server

Within Information Server, a user sees the Custom Panel as per details that the administrator has set for the user.

Workflow Farms

In the Workflow Farm, you can perform any workflow-related operation using the available links.

Workflow Repositories

- Click the **WISRepo** repository link.

  An automatic pop-up screen appears to display the Inbox, if the user authentication type is Active Directory.
  - If the user authentication type is AVEVA Work Tasks List (not Active Directory), you will be directed to a Login page for entering the credentials.
  - If the pop-up page does not open automatically, click the repository name in the right pane to open the repository.

Troubleshoot Issues

Issue:

In case of multiple providers, unable to browse the AVEVA Work Tasks website for URL mapped to Information

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Server.

Resolution:

1. In the **AVEVA Work Tasks Connector Configuration for Information Server** window, delete the provisioned repository.
2. Click **Add New** to add a new repository.

   The Server and Repository details window appears.
3. Enter the **Unique Environment Name, Server address, Authentication User Domain, Authentication User ID, Authentication,** and **User Password**.
4. Click **Test Connection**.
5. In Provision window, select the **Provider**.
6. Enter user name and password, and then click **Authorize**.

   The Repository name will be populated in the Server and Repository details window.
7. Go to **Advanced** tab.
8. Select the correct **User Provider Name** from the available list.
9. Click **Connect**.

Intelligence Data Inside Workflows

You can use the database activity to fetch the data from the AVEVA Work Tasks Intelligence (EMI) datastore. For details, see the subsequent topics.

Fetch Data from Intellience Datastore

You can use the database activity to fetch the data from the AVEVA Work Tasks Intelligence (EMI) datastore.

Database activity, a part of Engine activities, is an activity inside the workflow where the workflow establishes the connection to the database server and performs actions to retrieve data from the database. This activity executes user queries and the result is in created in an XML variable. This ml variable has the schema of columns of the database table and contains values based on the query made.

Before creating the database activity, you need to create a database connection object, which connects to the AVEVA Work Tasks Intelligence datastore.

Scenario: The AVEVA Work Tasks Intelligence datastore has the historical OEE data of the plant. This data can be sent to corporate management, plant management, and engineers, for them to know details of the performance, quality, and availability of the plant. For example, corporate management can be updated with the production information, such as total numbers of products, defective products, expenses of plant on a weekly basis.

For this feature to happen as part of the workflow, we need to create the database activity to fetch the data from the fact table and the dimension tables.

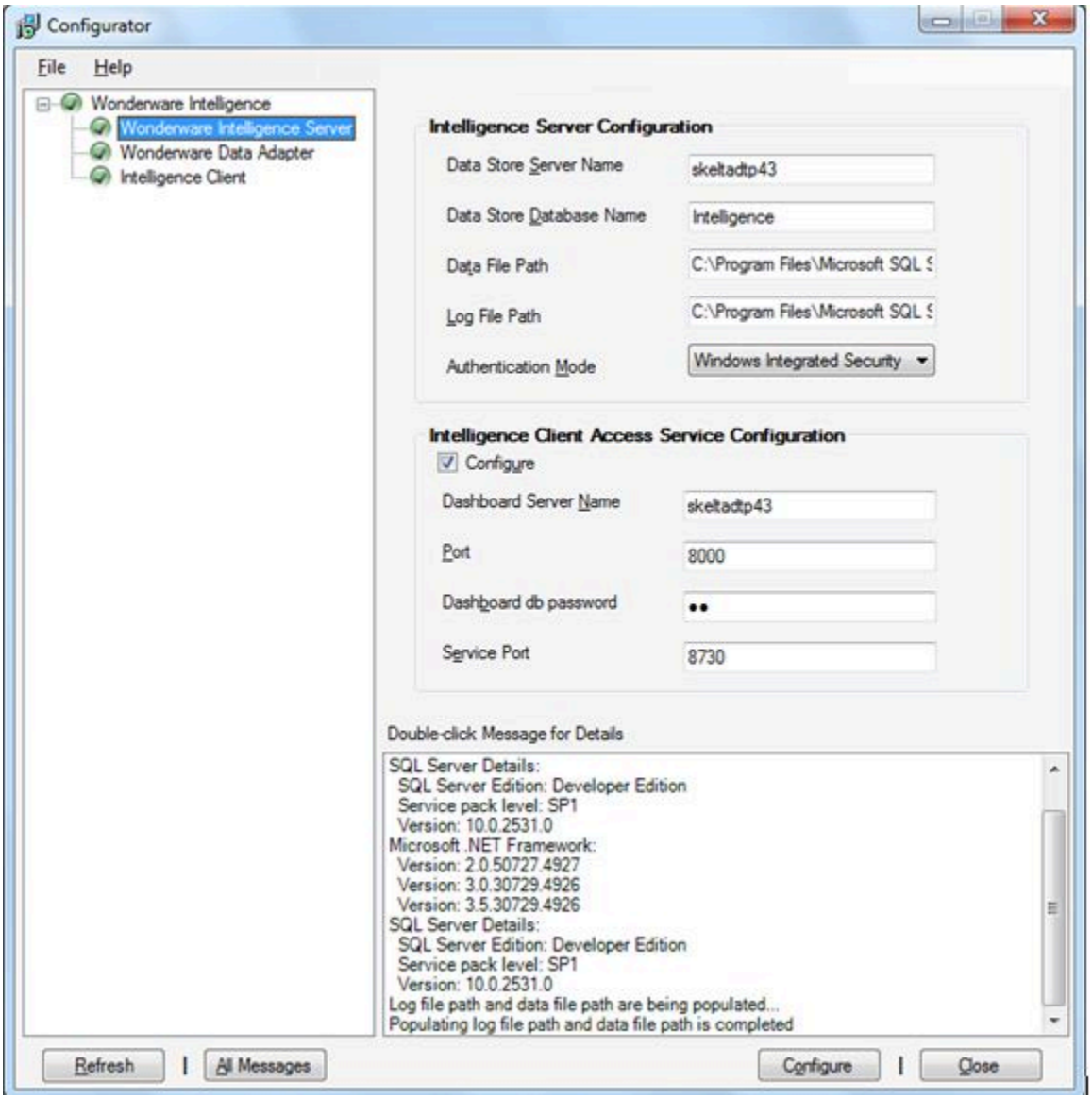
Configure Workflow

You need to first create the database connection to create the database activity.

To create the database connection for an activity, you need the server name, database name and database

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

authentication information of the datastore. You can get this information from the Wonderware Intelligence configuration.



To create a Database Connection for the Workflow:

1. Open the Database Connections object.
2. Click on New connection.
3. Enter title, description and uncheck the "Connect to repository datasource" check box.
4. Select SQL server as the database type.
   Keep the Connection string as Connection Type.
5. Enter the server name (Server name of datastore).
6. If the authentication type is SQL Server, then enter User Id, Password and Advance Settings.
7. Enter the Database name (datastore name).
8. Click on the refresh buttons to get all the databases on the specified server.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Click on Test connection.

   If the connection succeeds, the Save button is enabled.
10. Click Save.

    The Connection object is created and listed.

Add Database Activity

You can add the database activity to the existing workflow or to a new work flow. In the workflow, using the database activity, you can fetch the production information of the industrial plants and show them in the logger. This information can be mailed to users in real-time and can be scheduled using scheduling activity.

To add Database Activity

1. Select the Workflows option to create a new workflow.
2. To create a new workflow, click on Workflow.
3. Add the title for the workflow.
4. Click Save & Continue.
5. Select Design the Workflow now option.
6. Click Finish to design the workflow.
7. Go to the Engine activities to find Database activity.
8. Add the database activity to the workflow design.
9. Go to Properties of the database activity.
10. Change the Activity Name if needed.
11. Add the Description if needed.
12. Select the database connection from the drop-down list and click Save.
13. Click on the Query Builder icon to create the query to fetch data from Intelligence datastore.

Query Builder

1. To enter the query, click on the Query Builder property to open the property window.

   Set the following properties:

   - Command Type: Select the Query command.
   - Query: You can enter the query either in the text box or from Expression builder.
   - Output XML variable

   Example: The query can be "Select * from FactTable" and if the query is to be passed with parameters, then the query will be "Select * from FactTable where DefectiveProducts >@defectiveProducts". Here @ defectiveProducts is the parameter.
2. Click Load Parameters. It loads the parameters within the query. In the Value field you can pass the value for the parameter or assign the values from a workflow variable. Enter the Output XML variable to store the select result set in the XML format. Click Save.

   The query used in this example fetches the product-related information from the fact table and product dimension table. It fetches the total number of products, defective products and expenses based on the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

ProductionDate range and setting of defective products > 1000. The variable @defectiveProducts value is kept constant to 1000. The production date range values come for the work flow variables.

select p.Product, p.ProductType, f.TotalProducts, f.DefectiveProducts, f.TotalExpenses from FactTable f inner join Product p on f.productId = p.productId where f.DefectiveProducts > @defectiveProducts and ProductionDate between @currentDate1 and @currentDate2

3. Save the information. Click on 'View Schema' to view the schema of the output XML variable "outXmlInfo".

   The XML Variable "outXmlInfo" will be added in the Start Activity > XML Variables.

   Parameter values cannot update data types like ‘image’, ‘blob’ and ‘unique identifier’ from XML variables or variables.

4. Output XML Variable for return value: The Output XML variable will have the schema of the table as per the query entered. If the query is "Select * from Products", the XML variable will have all the columns as attributes in the schema. These can be looped through and used in the subsequent activities. The Output XML Variable name should be unique.

   To create an XML Iterator of the output XML Variable, the node selected should always be the "Table" node. XML Iterator can be created in the XMLVariables section of Start activity.

Activity outputs: The Database Activity returns the following values.

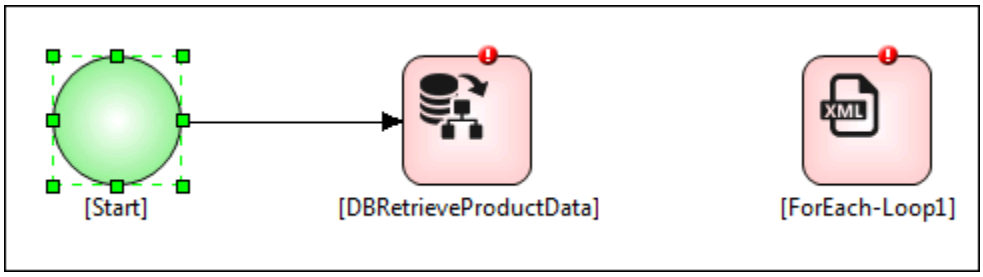
- Successful: Displayed when the query is executed.
- Unsuccessful: Displayed when the query execution is unsuccessful.

Add ForEach Loop

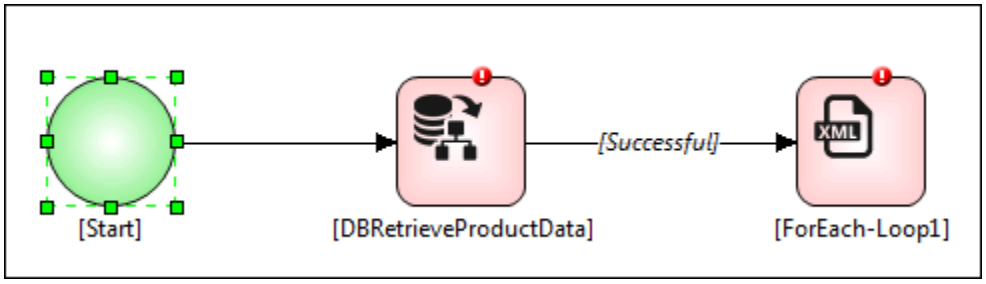
In case, if it is required to iterate through each node of the output XML variable, then we can use the foreach-loop.

In the following example, we will iterate through the output variable @outXmlInfo and log the values of each node.

1. Drag the ForEach-Loop activity from the Engine Activities section to the workflow.

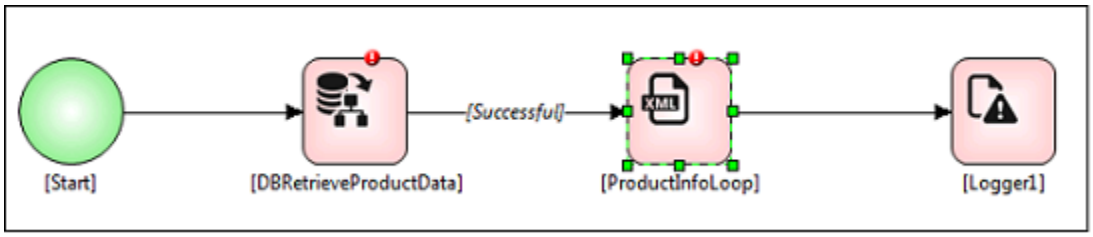


2. Add the connector and add the connector value as 'Success'.

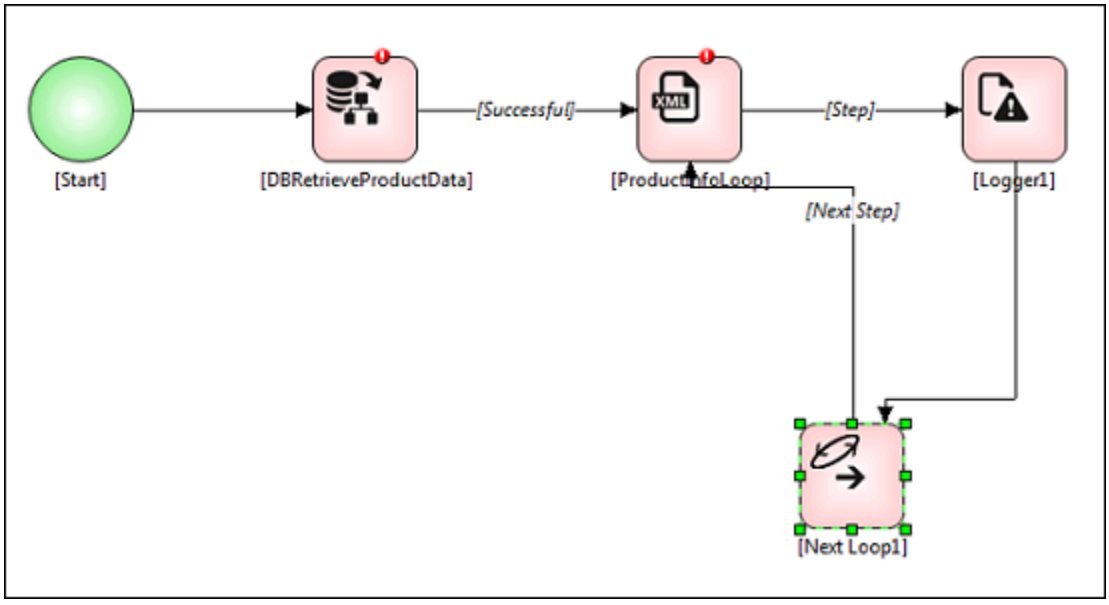


© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Go to the Property window of ForEachLoop. Change the Activity Name, if needed. Assign the XMLIterator which is created above in the last step of database activity. Click Save.
4. Add the logger to log the information of the iterated node values.



5. Add the step event to the connector.
6. Change the properties of the logger. Add Logtype as 'Information'. Click on the LogMessage icon to get the editor below.
7. Select XML variables, iterator and variable. Click on the Update button.
8. Save the Logger workflow.
9. For foreach loop workflow, we need to have a 'Next Loop' activity to go to the next node in the XML variable. Add the NextLoop workflow.
10. Add the connector between the Logger and Next Loop.
11. Add the connector between Next Loop and foreach loop.
12. Add the "Next Step" event to the connector of Next Loop and foreach loop.
13. Save the Next Loop activity.



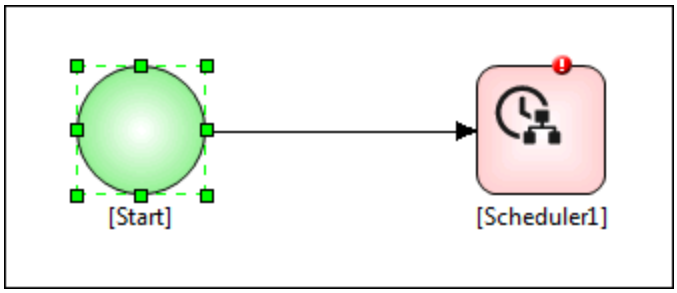
Edit Scheduler Activity

Database activity or complete workflow can be scheduled to run on weekly basis or at a specified date-time.

Output result of the Scheduler activity can be input to the next activity like the Database activity.

1. Select the Scheduler activity from the Scheduler activities group.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



2. Go to the properties of the Scheduler activity. Mention the Scheduler name, Scheduler details and the output path.
3. Configure the Scheduler details by clicking on the Scheduler details icon.
4. Let us configure the weekly scheduler as every Sunday, 1 AM recurring. Mention recurrence as 0 for having infinite recurrence. Click on the Update button.
5. Configure the output path. The Output path is the result of the Scheduler task after it is executed at the scheduled time. Let us keep its value as "executed".
6. Click on the **Save** button and connect it to the database activity. Select the "executed" event on the Scheduler connector.

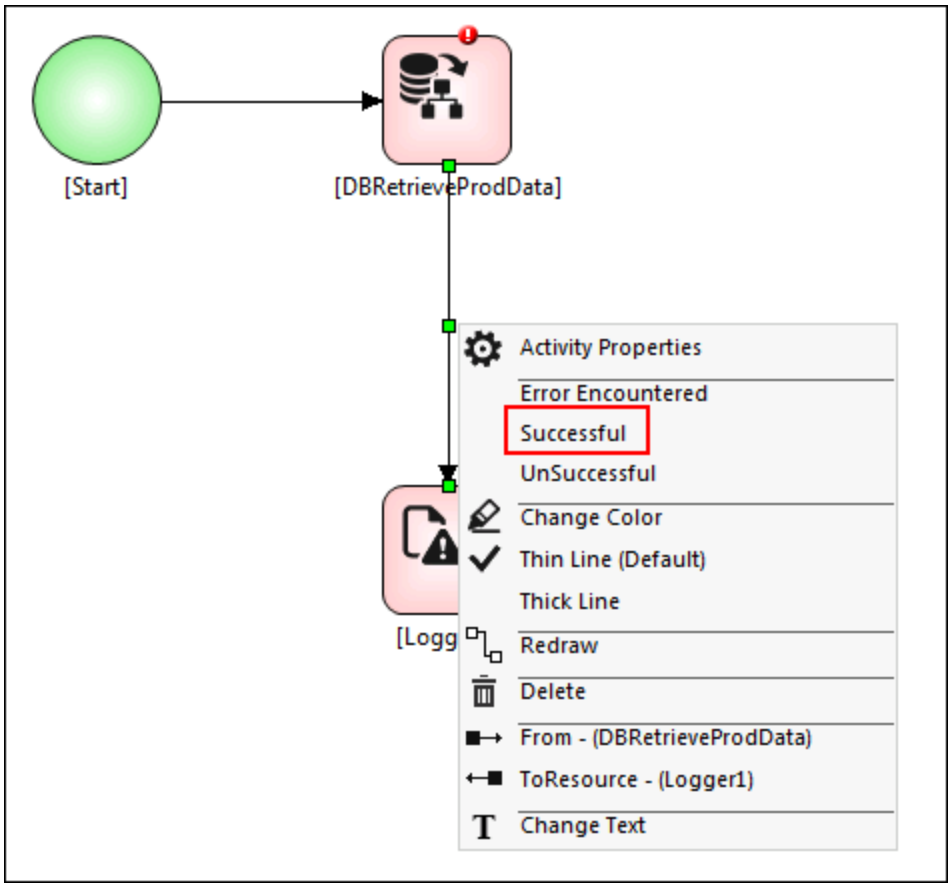
Diagnostics

To add Logger activity

We can add the logger activity to show the retrieved data in the logger. In the example, on successful execution of the Database activity, the retrieved data will be stored to the XML variable outXmlInfo. The value from this variable can be shown in the logger.

1. Add the Logger activity.
2. Add the connector between database activity and logger and ensure that Successful event is selected.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Go to the property window of **Logger** and select **Logtype** as Information.
4. Mention the logger message using the Expression Editor. Expression Editor is needed to show the XML variable values. Click the **Expression Editor** icon to open the Expression Editor. Add the text to editor and click the **OK** button.
5. Save the **Property** window of the logger.
6. Publish the activity and **Test Run** it.

   The results can be seen in the Logger Console.

Move System Platform Galaxy and AVEVA Work Tasks Repository from Engineering Environment to Production Environment

This document gives detailed instructions to move the System Platform Galaxy and AVEVA Work Tasks Repository from engineering environment to production environment. In addition, this document also gives the detailed steps to move only the workflow and some of the galaxy objects from engineering to production environment.

Following three scenarios have been explained in this document:

- Moving System Platform Galaxy and AVEVA Work Tasks Repository from Engineering Environment to Production Environment
- Moving only the Workflow from Engineering Environment to Production Environment
- Moving a Gateway Object, Workflows, and Associated Workflow Events

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Prerequisites

- Prior knowledge of AVEVA Work Tasks is required.
- AVEVA Work Tasks Repository should be provisioned from ArchestrA in the engineering environment.
- In the production environment, System Platform Workflow Connector should be installed on the GR Node and System Platform Workflow Extension should be installed on the AVEVA Work Tasks Server.

Move System Platform Galaxy And Repository

This section gives you step by step instructions on how to import ArchestrA Galaxy and Repository Database from engineering environment to production environment:

- Importing Repository Database from engineering environment to production environment
- Importing ArchestrA Galaxy from testing environment to production environment

Move Repository

We recommend to have the same domain and galaxy name on both the engineering and production environment.

The following steps also take care of domain and galaxy name change in the production environment.

Complete the following steps to import Repository Database from engineering environment to production environment:

1. Take a backup of the required Repository database present in engineering environment's Server.
2. Copy the database backup file in the production environment.
3. Restore the database in the production environment of the Workflow server.
   a. Open SQL Server Management Studio and connect to Database Engine.
   b. In the **Object Explorer**, go to the Database.
   c. Right-click the Database and select **Restore Database** from the context menu to restore "XXX" database.
4. Execute the SQL script given below on the restored Repository database in production environment.

   Before executing the script the following changes needs to be done:
   - Set or change the value of @OldRepositoryName with Workflow Repository name as in the engineering environment.
   - Set the value of @UniqueEnvironmentName with Unique Environment Name used in the engineering environment while provisioning the Repository from ArchestrA.

     -- Declaring variables

     DECLARE @OldRepositoryName AS NVARCHAR(150)

     DECLARE @OldUniqueEnvironmentName AS NVARCHAR(150)

     -- Set the Old Workflow repository name

     SET @OldRepositoryName = '<Old Repository Name>'

     -- Set the Unique Environment name - The Unique Environment Name used to initially while provisioning the workflow repository in engineering environment

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
SET @OldUniqueEnvironmentName = '<UniqueEnvironmentName>'

-- Deleting all ArchestrA external connections details entries from SKExternalConnectionDetails table

DELETE FROM SKExternalConnectionDetails WHERE RecordParentItemId IN (SELECT Id FROM
SKExternalConnections WHERE SourceType='ArchestrA' AND Application=@OldRepositoryName AND
Title=@OldUniqueEnvironmentName)

AND Application=@OldRepositoryName

-- Deleting all ArchestrA external connections information from SKExternalConnections table

DELETE FROM SKExternalConnections WHERE SourceType='ArchestrA' AND Application
=@OldRepositoryName AND Title=@OldUniqueEnvironmentName

-- Deleting Galaxy User resource mapping from SKESGResource Mapping table

DELETE FROM SKESGResourceMapping WHERE UserId In(SELECT Id FROM SKVirtualActor WHERE
Provider = @OldUniqueEnvironmentName AND ResourceHandler=@OldRepositoryName)

AND Application=@OldRepositoryName

-- Deleting galaxyuserprovider and galaxyrole provider entries from SKEntityDataSourceProvider table

DELETE FROM SKEntityDataSourceProvider WHERE Provider='galaxyuserprovider' OR
Provider='galaxyroleprovider' AND Application=@OldRepositoryName AND
InstanceName=@OldUniqueEnvironmentName

-- Deleting 'galaxyaduserprovider' and 'galaxyadroleprovider' entries from SKEntityDataSourceProvider
table

DELETE FROM SKEntityDataSourceProvider WHERE Provider='galaxyaduserprovider' OR
Provider='galaxyadroleprovider' AND Application=@OldRepositoryName AND
InstanceName=@OldUniqueEnvironmentName

-- Updating SKEEventAssociations table and Setting EventBindingXML with rootnodes where ever
EventBindingXML is empty

-- This step is required to avoid the 'Root Element is missing' message while attaching the datasource and
creating a repository

UPDATE SKEEventAssociations SET EventBindingXML = '<EventBind></EventBind>' WHERE
RTRIM(EventBindingXML) = '' AND Application =@OldRepositoryName AND
EventProviderName='ArchestraEvents'

-- Truncating Galaxy Role User Mappings table

DELETE FROM GalaxyRoleUserMappings WHERE UserId IN (SELECT Id FROM GalaxyUsers where
UniqueGalaxyName=@OldUniqueEnvironmentName)

-- Deleting Galaxy Role Security Group Mappings table

DELETE FROM GalaxyRoleSecurityGroupMappings WHERE Id IN ( SELECT Id FROM GalaxyRoles WHERE
GalaxyName = @OldUniqueEnvironmentName)

-- Deleting Galaxy Roles from Galaxy Roles table

DELETE FROM GalaxyRoles WHERE GalaxyName = @OldUniqueEnvironmentName

-- Deleting Galaxy Users from Galaxy Users table

DELETE FROM GalaxyUsers WHERE UniqueGalaxyName = @OldUniqueEnvironmentName
```

5. If the Repository in the engineering environment is also provisioned from MES, then the SQL script (given below) needs to be executed on the restored Repository Database in the production environment.

   Before executing the script the following changes needs to be done:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Set or change the value of @OldRepositoryName with Workflow Repository name as in the engineering environment.
- Set the value of @UniqueEnvironmentName with Unique Environment Name used in the engineering environment while provisioning the Repository from MES.

  -- Declaring variables

  DECLARE @OldRepositoryName AS NVARCHAR(150)

  DECLARE @OldUniqueEnvironmentName AS NVARCHAR(150)

  -- Set the Old Workflow repository name

  SET @OldRepositoryName = '<Old Repository Name>'

  -- Set the Unique Environment name - The Unique Environment Name used to initially while provisioning the workflow repository from MES in engineering environment

  SET @OldUniqueEnvironmentName = '<UniqueEnvironmentName>'

  -- Deleting all MES Connection details entries from MESConnectionDetails table

  DELETE FROM MESConnectionDetails WHERE RecordParentItemId IN

  (

  SELECT Id FROM SKExternalConnectionDetails where RecordParentItemId IN

  (

  SELECT Id FROM SKExternalConnections WHERE SourceType='MES' AND Application=@OldRepositoryName AND Title=@OldUniqueEnvironmentName

)

  AND Application =@OldRepositoryName

)

  -- Deleting all MES external connections details entries from SKExternalConnectionDetails table

  DELETE FROM SKExternalConnectionDetails WHERE RecordParentItemId IN (SELECT Id FROM SKExternalConnections WHERE SourceType='MES' AND Application=@OldRepositoryName AND Title=@OldUniqueEnvironmentName)

  AND Application=@OldRepositoryName

  -- Deleting all MES external connections information from SKExternalConnections table

  DELETE FROM SKExternalConnections WHERE SourceType='MES' AND Application =@OldRepositoryName AND Title=@OldUniqueEnvironmentName

  -- Deleting MES User resource mapping from SKESGResourceMapping table

  DELETE FROM SKESGResourceMapping WHERE UserId In(SELECT Id FROM SKVirtualActor WHERE Provider = @OldUniqueEnvironmentName AND ResourceHandler=@OldRepositoryName)

  AND Application=@OldRepositoryName

  -- Deleting mesuserprovider and mesroleprovider entries from SKEntityDataSourceProvider table

  DELETE FROM SKEntityDataSourceProvider WHERE Provider='mesuserprovider' OR Provider='mesroleprovider' AND Application=@OldRepositoryName AND InstanceName=@OldUniqueEnvironmentName

  -- Updating SKEEventAssociations table and Setting EventBindingXML with rootnodes where ever EventBindingXML is empty

  -- This step is required to avoid the 'Root Element is missing' message while attaching the datasource &

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

creating a repository

UPDATE SKEEventAssociations SET EventBindingXML = EventBindingXML + '</EventBind>' WHERE SUBSTRING(EventBindingXML,LEN(EventBindingXML)-11,12) != '</EventBind>' AND Application = @OldRepositoryName AND EventProviderName='MESEvents'

6. Repository Data Cleaning: If you do not wish to retain the workflows runtime data, then execute the following script on restored Repository Database in the Production environment.

   -- Script for removing workflows runtime data

   TRUNCATE TABLE SKWorkItem

   TRUNCATE TABLE SKHWSActivity

   TRUNCATE TABLE SKWorkItemTransaction

   TRUNCATE TABLE SWTaskList

   TRUNCATE TABLE SKMultipleTimeoutWarning

   TRUNCATE TABLE SWActionEventLog

   TRUNCATE TABLE SWActionEventLong

   TRUNCATE TABLE SKWorkItemCounter

   TRUNCATE TABLE SKChannelWorkItem

   TRUNCATE TABLE SKWorkItemData

   TRUNCATE TABLE SKWorkItemFields

   TRUNCATE TABLE SWExecutionDetails

   TRUNCATE TABLE SWExecute

   TRUNCATE TABLE SWVariables

   TRUNCATE TABLE SKWorkflowEvents

   TRUNCATE TABLE SKChildWorkflows

   TRUNCATE TABLE SWCustomTimerDetails

7. In the production environment, navigate to Central Configuration to access the workflow server.
8. Click the **Datasource Settings** link.

   The **Datasource List** window is displayed.
9. Click the **Create Datasource** link.

   The **Manage Datasource** window is displayed.
10. Enter all the required values. Ensure that the restored database is selected for the Database value.
11. Click **Test Connection**. After the Test Connection is successful, a message is displayed.
12. Click **Ok**.
13. Navigate to **Central Configuration > Repository Settings** to view the **Repository List**.
14. Click **Add**.

    The **Manage Repository** page appears.
15. Select the appropriate data source (DevDS1) and check the **Attach Datasource** check box.
16. Enter the repository name, repository description, server name, and select the required Resource Provider from the drop-down list in the **Manage Repository** window.
17. Click **OK** to create the Repository.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Config Administrator** window is displayed.

18. Select "admin [AVEVA Work Tasks List]" in the **User Lookup** field and click **OK**.

    In this page, configure the administrator for the Repository.

    - In case the provider is "WorkflowManagementList", select the default user admin as administrator.
    - In case of other providers, use the user lookup and select the required user to configuring as an administrator.

19. Navigate to the Repository List and click the Site Path (url) to open the Enterprise Console Page.
20. Login to the Enterprise Console site and check whether all the workflows are displayed in the Workflow list.

Move System Platform Galaxy

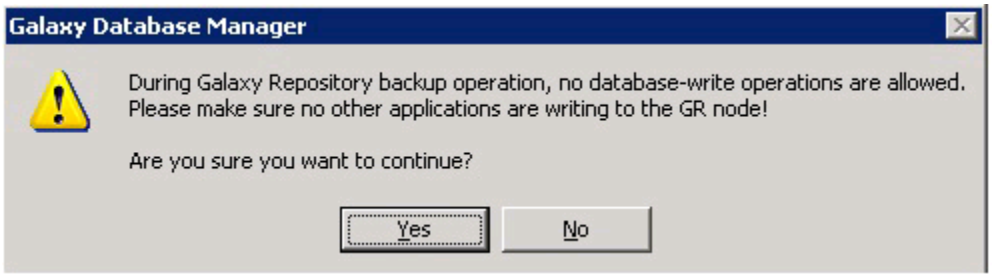
Steps to import ArchestrA Galaxy from testing environment to production environment:

1. Access the engineering environment in your machine.
2. Go to **Start > Run > aaSMC** and click **Enter**.

   The **ArchestrA System Management Console** is displayed.
3. Expand the Galaxy Database Manager.
4. Select the required galaxy instance which has to be moved to production environment.
5. Right-click the Database and select the **Backup** option.

   The Galaxy should be in Un-deploy state before taking the galaxy backup in the engineering environment.

   A warning message as shown in the following image appears.



6. Click **Yes**.
7. Select a path where the *backup.cab* file has to be saved and click **Save**.

   The **Galaxy Database Manager** screen appears.
8. Click **Close** after the backup is saved and copy the .cab file from the folder in which it was saved.
9. Access the production machine and create a new galaxy with the same name as in the engineering environment.
10. Go to the **Start > Run > aaSMC** and click **Enter**.

    The **ArchestrA System Management Console** appears.
11. Expand the Galaxy Database Manager.
12. Right-click the galaxy DevToProd and click the **Restore** option.
13. Select the ArchestraWorkflowSoftware.cab backup file and click on **Open**.
14. After restoring successfully, access System Platform IDE and change the GR_Platform network address to the actual production GR address. Save and close the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

15. Open WorkflowGateway (WGO) derived template/instance and re-provision with restored Repository in production environment.

    While re-provisioning let's keep Unique Environment Name same as existing (specified in the engineering environment)

    If multiple Workflow Gateway objects are present, then each of the Workflow Gateway object needs to be re-provisioned to the Repository.

    In case of derived template instances, only derived template need to be re-provisioned.
16. After re-provisioning the WGO, deploy the entire platform.
17. After successful deployment, you can see the all associated workflows under the workflow tab of the respective instances. Now trigger the workflow to test whether it is working or not.

    Ensure you use the same user in engineering and production environment, to see the associated Workflows.
18. If the Workflow is not triggering, then go to **Start > Run >** type **Services.msc** and click the **Enter** button.
19. Verify if SkeltaConnectorClientArchestrA service is running or not. If it is not running, then right-click on the service and click the **Start** option.
20. Verify again whether the workflow is triggered or not.

See *Using Workflow Features Inside System Platform IDE and Runtime* to provision a WorkflowGatewayObject and trigger a workflow.

Move only the Workflow

In this scenario:

- A workflow is exported from the engineering environment.
- A new workflow is created through the Workflow Management tab in the System Platform IDE.
- Workflow exported from the engineering environment is imported to the production environment and published.

1. Access the System Platform IDE in the engineering server.
2. Open the instance where workflow has been associated with the events.
3. Select the workflow, and then click **Unpublish**.

   A confirmation message about successful unpublish appears.
4. Click **Close**.
5. Select the workflow again, and then click the **Design**.

   The **Process Designer** window appears.
6. On the menu bar, click **Tool**, and then click **Export**.

   The File Download dialog box appears.
7. Click **Save** to save it as an XML file.
8. Move the Workflow XML file to production machine.
9. Open System Platform IDE in the production machine.
10. Open the specific instance to which you want to associate this workflow.
11. Create a new Workflow event. For example, *WF_Event4*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

12. Select **Workflow Gateway** instance from the drop down list.
13. Click **Select**.
14. Click **New Workflow**.

    The **New ArchestrA_Workflow** window appears.
15. Enter the **Title** and **Description**.
16. Click **Save & Continue**.

    The **Process Designer** window appears.
17. On the menu bar, click **Tool**, and then click **Import**.

    The **Workflow Management – Import – Process Definition** dialog box appears.
18. Click **Import** to import the XML file you copied from the engineering machine.

    It imports the actual workflow which you have designed in engineering machine.

    Assigned actor(s) can be changed using the activity property, if required.
19. Create all the associations required in the workflow.
20. On the menu bar, click **Publish**.

    A confirmation message about successful publish appears.
21. Click **Close,** and then go back to the System Platform IDE.
22. In the System Platform IDE, add the **Event Expression** and then click **Save** to save the instance.
23. Deploy the object.

    When an event occurs, the associated workflow will be triggered.

Move a Gateway Object, Workflows, and Associated Workflow Events

In this scenario:

A Workflow Gateway Object (WorkflowGateway_002) is created and a Drum_Conveyor instance is created. The Drum_Conveyor instance uses the WorkflowGateway_002 to associate workflow event. The steps followed for this are given below:

1. Create a Workflow Gateway Object in the engineering environment and provision it to AVEVA Work Tasks Repository.
2. Create a Drum_Conveyor instance. Add Workflow Event and associate a workflow to it.
3. Save and deploy the Workflow Gateway Object and Drum_Conveyor instance. Trigger the workflow.
4. If everything is working as expected, access System Platform IDE in the engineering machine.
5. Select the WorkflowGateway_002 and Drum_Conveyor_002.
6. Select **Galaxy > Export > Object(s)**.
7. Save the exported object in a disk. In addition, follow the steps 1-8 to export the workflow given in the above section Move only the Workflow.
8. Move exported objects (.aaPKG) and workflow (.xml) to the Production machine. Follow the steps given below.
9. Open the System Platform IDE of Production machine.
10. Go to **Galaxy > Import > Object(s)**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

11. Select the .aaPKG file that was moved from the engineering to production machine.
12. After successfully importing, re-provision the WorkflowGateway_002 with the Repository in production server.
13. Save and deploy the **WorkflowGateway_002**.
14. Open the Drum_Conveyor_002 and click the **Workflows** tab.
15. Select the events you have created in the engineering environment. Click the **Refresh** button of the **Workflow Gateway** drop-down list.
16. Select **WorkflowGateway_002** from the drop-down and click the **Select** button.
17. Click **New Workflow** and follow the steps given in "Move only the Workflow" (Steps 11 to 19).

Diagnostics: Logger Integration with ArchestrA Logger

In the AVEVA Work Tasks Integrated ArchestrA environment, troubleshooting is made easy by having a single tool for debugging messages from WTLogger Console and Wonderware Logger.

The Logger is able to log messages to the Logger, thereby leveraging the rich UI features of System Management Console for viewing messages. The Logger has rich UI features such as filtering and search option. Using the Logger Integration, you will be able to debug messages more effectively. The following screenshot of the Log Viewer show details of logged messages.

Available Log Options in AVEVA Work Tasks

AVEVA Work Tasks has the following logging options available:

- aaLogViewer
- AWS Logger
- Event Viewer
- Mail Handler
- Log file

If you install the AVEVA Work Tasks in the same environment as System Platform then the following default options are available:

- aaLogViewer
- AWS Logger
- Mail Handler

To modify the above-mentioned options, you must add the following registry entry of type string:

- Path: HKEY_LOCAL_MACHINE\SOFTWARE\AVEVA\WorkTasks
- Key: logToOptions
- Value: Refer to the following table:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Value	Description
ALL	AVEVA Work Tasks will log to aaLogViewer (If Wonderware environment), AWS Logger, Event Viewer, Mail Handler, Log file.
WL	Only AVEVA Work Tasks will log to aaLogViewer.
SLC	Only AVEVA Work Tasks will log to AWS Logger.
ML	Only AVEVA Work Tasks will log to Mail Handler.
EV	Only AVEVA Work Tasks will log to Event Viewer.
LF	Only AVEVA Work Tasks will log to file.

Note: If you have modified any values, restart the AVEVA Work Tasks services for the change to take effect.

Known Issues

- **"Could not create class ArchestrA.Toolkit.ArchestrAWorkflowGateway.ArchestrAWorkflowGatewayConfigtime. Class not registered" gets logged in the ArchestrA logger during Gateway Object configuration.**

  Possible Cause:

  This error is seen by Beta users. There is already a Gateway Object instance for a non-active Galaxy. If you are using the release AppObject, ensure that all the Galaxies are cleaned by removing the Gateway Object instance from the non-active as well as the active Galaxy. This error is due to a Gateway Object version change (from Beta to Release candidate).

- **Wonderware Information Server (WIS)**
  - The theme functionality will not work in WIS.
  - Before uninstalling Workflow Server WIS extension, you should stop the Workflow Engine service.
  - If you are configuring multiple server repositories for WIS, you will lose the Workflow Repository Node and Workflow Farms Node links if you uninstall the servers - WorkflowserverWIS.

- **Workflow Association**
  The Synchronize XML UI pop-up window displays invalid Help text. To synchronize, the user has to only click the **Synchronize** button.

- **JavaScript Error**: A JavaScript error occurs if you right-click on a workflow item.

- **Gateway Object:**

  Exception occurs if two Gateway objects mapping to different repositories are provisioned to the same data source. Ensure to have a separate data source for each repository.
  Renaming of gateway object and event is not supported.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Enterprise Console repository:**

  The EC repository list displays "No Providers" after provisioning the Galaxy for a repository. A workaround is to perform an IISRESET.

- **Windows opened are not closed if Change user or Change Galaxy is done from IDE.**

  While creating or editing a workflow event from the workflow tab of an object, you can open different windows as popup. Without closing these windows if you try to change the galaxy or user, these windows will not get closed automatically. Similar behavior can happen for the popup windows opened from the tabs shown using workflow toolbox.

- **Clicking multiple times on 'New workflow' or 'Existing Workflows' or 'Manage workflows' icon in Event association page is opening multiple windows.**

  While creating or editing a workflow event from the workflow tab of an object, you can click multiple time on the ribbon bar('New workflow' or 'Existing Workflows' or 'Manage workflows') of workflow association section, which will result opening new separate windows. Similar behavior can happen from the tabs shown using workflow toolbox.

- **Reference warning on importing a template or object which has multiple workflow events associated with it.**

  The user exports an object or template which is associated with multiple workflow events. The same object or template is imported to a different Galaxy, which does not have workflow gateway object with the same name as in the parent Galaxy. Warning messages are shown in the Instance Property window.
  We recommend to follow the steps given below to avoid the reference warnings on importing a template or object:

  1. Close the System Platform IDE.
  2. Go to the <ArchestrA Install Path>\ArchestrA\Framework\FileRepository\CustomCategoryUIFiles folder. Remove ArchestrA.Workflow.IDEExtension.dll from this folder. Take a backup of the dll prior to removing it from the folder.
  3. Go to the <ArchestrA Install Path>\ArchestrA\Framework\Bin\IDEExtensions folder. Remove ArchestrA.Workflow.IDEExtension.dll from this folder.
  4. Open the System Platform IDE.
  5. Open the respective template instance.
  6. Go to the Script tab and open the scripts with names starting with _aaWFE. Find and replace the workflow gateway object name with the workflow gateway object name from the parent Galaxy.
  7. Save the details and close the System Platform IDE.
  8. Place ArchestrA.Workflow.IDEExtension.dll in the below mentioned path:

     <ArchestrA Install Path>\ArchestrA\Framework\FileRepository\CustomCategoryUIFiles
  9. Open System Platform IDE and verify.

- **Mouse-over options not available for Form or Control properties**

  When the Form or Control properties are opened in the Forms Designer window and saved, you do not get the mouse-over options against that control. It is recommended to refresh the Form Designer window to get the mouse-over options.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Limitations

- The Workflow-Toolbox provides a list of GatewayObject instances configured in Galaxy. Selecting a GatewayObject instance will list the EC menu for the mapped repository in Tree view. Clicking on the link will open the respective tab window on the right side of the IDE. Using the Workflow-Toolbox users can work with multiple repositories. However, there is no support for working with multiple repositories (GatewayObject instance) at the same time. Close all the open windows for a repository before working on another repository. Switching over to one repository while having one or more windows open for different Repository could lead to termination of session. Multiple presses on the Back button on these windows is not supported.
- Authentication or Verifier user credential used for the 'Write' Attribute should be a user-created before deploying the GatewayObject. Any user used for authentication or verification after GatewayObject deployment may not work and may require a redeploy of GatewayObject.

Troubleshoot Issues

Problem

Error while trying to start **ArchestrA Connector Client** service, which was paused or stopped. The following error is logged while redeploying the **Gateway** object:

```
Error while trying to start ArchestrA connector client service, which was paused  or
stopped. Error :System.InvalidOperationException: Cannot start service
SkeltaConnectorClientArchestrA on computer '.'. ---> System.ComponentModel.Win32Exception:
An instance of the service is already running
   --- End of inner exception stack trace ---
   at System.ServiceProcess.ServiceController.Start(String[] args)
   at System.ServiceProcess.ServiceController.Start()
   at Skelta.Runtime.Gateway.GatewayRuntime.CheckServiceStatus(Object nullParam)
```

Resolution

Undeploy the **Gateway** object and ensure that no errors are getting logged into the logger. Redeploy the **Gateway** object.

Problem

Inbound authentication user could not be validated in platform. The following error is logged while provisioning and deploying the **Gateway** object:

```
Inbound authentication user (SKELTANB26\John)  could not be validated in platform, this
could affect the inbound connection from workflow server.Please provide valid user
information and re deploy Gatewayobject.
```

Resolution

1. Ensure that the inbound information entered is correct.
2. If the user belongs to the local machine, enter **Authentication User Domain** as **localhost** instead of giving the machine name.
3. Click **Save**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Problem

The given key was not present in the dictionary. The following error is logged while deploying the gateway object:

```
System.Collections.Generic.KeyNotFoundException: The given key was not present in the
dictionary.
at System.ThrowHelper.ThrowKeyNotFoundException()
at System.Collections.Generic.Dictionary`2.get_Item(TKey key)
at Skelta.Connector.Proxy.SingletonProxy`1.ReCreateConnection(String
multiHostSingleProxyIdentifier)
at Skelta.Connector.Proxy.SingletonProxy`1.GetConnection(String
multiHostSingleProxyIdentifier, Boolean& isOldConnectionExpired)
at Skelta.Connector.Proxy.EngineProxy.GetChannel()
at Skelta.Connector.Proxy.EngineProxy.ConnectorRuntimeShutdown(String connectorPluginName,
String uniqueEnvironmentName, String[] connectorContext, String repositoryName, String[]&
error)
```

Possible Cause

The gateway object is initially deployed with machine name as the **Server Address**. The user redeploys the gateway object by changing the **Server Address** to either **localhost** or IP address.

Resolution

1. Undeploy the **Gateway** object.
2. Edit the **Server Address** as required and click **Save**.
3. Reauthorize and redeploy the **Gateway** object.

Problem

Authorization fails with the following error after entering the credentials when the **Provider** selected is **Galaxy Users** and none of the **Gateway** objects are deployed from that **Galaxy** to the same repository.

Resolution

Ensure that at least one **Gateway** object is deployed from this Galaxy to the same repository.

Problem

Page Not Found error or blank page in the **System Platform IDE**.

Possible Cause

‘**--**‘ prefix in the in the **Enterprise Console** site URL

Resolution

Remove the ‘--’ prefix while editing the **Enterprise Console** site URL in the workflow **Gateway** object.

Problem

"**Could not connect to the Workflow Server...**" error on **Test Connection** while connecting **Galaxy** to the **Workflow Server**.

Resolution

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Turn off the **Windows Firewall**.
2. Ensure that the **Workflow Engine** service is running.
3. If the **Workflow Engine** service is not running, start the service. Check the message logged in the logger and note down the **Port Number** for further verification.
4. Enter the required information for **Gateway Configuration**.
5. In the **Advanced** tab of the **Gateway Configuration**, ensure that the **Outbound Connector Port Number** is same as the **Port Number** logged in the logger.
6. Click **Test Connection** and verify.

Problem

Slow response time from **AttributeReadWrite** object.

Possible Cause

Failed to acquire any type of license feature **MXAccess_Runtime**.

Resolution

Acquire a valid **MXAccess_Runtime** license on the node where the **Workflow Gateway** object is hosted.

Problem

The Enterprise Console menu does not load when you try to connect to the GR node through IDE, from a different server.

Possible Cause

The required files are not copied on the IDE node,therefore the Enterprise Console menu does not load.

Resolution

1. Open **System Platform IDE**, specify the **GR node name** and **Galaxy name** ,and connect to the galaxy.
2. Open **Workflow Gateway Object** and close it without saving.
3. Restart IDE and connect to the galaxy.
4. Go to **Workflow - Toolbox**
5. Select **Workflow Gateway object** from the dropdown.
6. The Enterprise Console menu is displayed.

Problem

The Enterprise Console page crashes in IDE with the message: Error decrypting the request (The input is not a valid Base-64 string as it contains a non-base 64 character, more than two padding characters, or an illegal character among the padding characters).

Possible Cause

The DLL files are mismatched as they are not copied from the object location to the **CustomCategoryUIFiles** and **IDEExtensions** folders.

Resolution

Manually copy the connector DLL files from **C:\Program Files (x86)\ArchestrA\Workflow\IDEExtension** to:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **C:\Program Files (x86)\ArchestrA\Framework\FileRepository\CustomCategoryUIFiles**
- **C:\Program Files (x86)\ArchestrA\Framework\Bin\IDEExtensions**

Problem

The page does not get displayed for InTouch OMI when the controls does not load.

Possible Cause

- The Workflow Gateway Object might be configured with a user that might not be available in the current provider.
- Browser caching issue.

Resolution

- Deselect all the checkboxes for **Compatibility View Settings**.
- Clear browser cache.

  To clear Chrome browser cache, go to **Settings** --> **Clear browsing data** --> **Advanced** tab, and then select **Password and other sign-in data** checkbox.
- Set User Authentication for Security Settings - Internet Options
  - For Internet

    Navigate to **Custom Level** --> **User Authentication** (scroll down towards end) --> **Logon**. Select **Automatic logon only in Intranet zone** option, and then click **Ok**.
  - For Local Intranet

    Navigate to **Custom Level** --> **User Authentication** (scroll down towards end) --> **Logon**. Select **Automatic logon only in Intranet zone** option, and then click **Ok**.
  - For Trusted Sites

    Navigate to **Custom Level** --> **User Authentication** (scroll down towards end) --> **Logon**. Select **Automatic logon only in Intranet zone** option, and then click **Ok**.
  - For Advanced Tab

    Select **Enable Integrated Windows Authentication** option.

Problem

On importing the Workflow Gateway Object .pkg file in the Upgrade environment, below errors are logged in aalogviewer:

```
System.MissingMethodException: Method not found: 'Void
Skelta.WinForms.ExtendedWebBrowser.UIHostBase.set_OverrideTabAndDeleteKey(Boolean)'.
at ArchestrA.InTouch.SkeltaControls.Controls.BAM.BAMControl.InitializeComponent()
at ArchestrA.InTouch.SkeltaControls.Controls.BAM.BAMControl..ctor() Error in constructor
BAM control
System.MissingMethodException: Method not found: 'Void
Skelta.WinForms.ExtendedWebBrowser.UIHostBase.set_OverrideTabAndDeleteKey(Boolean)'.
at ArchestrA.InTouch.SkeltaControls.Controls.FillForm.FillFormControl.InitializeComponent()
at ArchestrA.InTouch.SkeltaControls.Controls.FillForm.FillFormControl..ctor()Error in
constructor FillForm control
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Probable cause

The IE Web Browser control is replaced by Microsoft Edge WebView2 control. Hence, the errors are logged because of caching after the upgrade.

Resolution

Clear the cache and restart System Platform IDE.

Problem

In System Platform IDE, if configured with AIM and provisioned with AIM repository, sometimes it prompts for AIM credentials on browsing Enterprise Console pages, and sometimes not.

Probable cause

There is no logout mechanism available in it, so it will take from cache and not prompt for AIM credentials.

Resolution

To access Enterprise Control pages, you must provide AIM credentials if prompted.

Work with Extension for OPC UA

OPC or OLE Process Control is a series of standard specifications resulted from the collaboration of a number of leading worldwide automation suppliers working in cooperation with Microsoft. It was originally based on Microsoft's OLE COM (component object model) and DCOM (distributed component object model) technologies. The specification defined a standard set of objects, interfaces and methods for use in process control and manufacturing automation applications to facilitate interoperability. The COM/DCOM technologies provided the framework for software products to be developed. There are now hundreds of OPC Data Access servers and clients.

OPC UA or OPC Unified Architecture is a new set of specifications that are not based on Microsoft COM. The Extension for OPC UA helps you to read or write tag values from the OPC UA server.

Capabilities:

- Connect AVEVA Work Tasks to one or more existing OPC UA servers on the network.
- Read or Write Tag values from the OPC UA server which is connected to the AVEVA Work Tasks Server.

Provision Repository for OPC UA

The AVEVA Work Tasks OPC UA Repository Provisioning Tool is used to provision a workflow repository for OPC UA. Only the provisioned repositories can be connected to OPC UA. Workflow repository provisioning for OPC UA should be done only after installing AVEVA Work Tasks.

OPC UA Repository Provisioning Tool

You can use the **AVEVA Work Tasks OPC UA Repository Provisioning Tool** to provision a workflow repository for OPC UA. Only the provisioned repositories can be connected to OPC UA. Workflow repository provisioning for OPC UA should be done only after installing AVEVA Work Tasks.

The **AVEVA Work Tasks OPC UA Repository Provisioning Tool** is installed during the installation of AVEVA Work

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Tasks Extension for OPC UA.

To Provision a Repository for OPC UA:

1. Run the **AVEVA Work Tasks OPC UA Repository Provisioning Tool** from the **Start** menu > **All Programs** > **AVEVA** > **Work Tasks** folder.

   The AVEVA Work Tasks Extension for OPC UA Configuration window appears. This window displays all the repositories from the system.
2. Right-click on any cell column of the particular repository, and then select the **Provision** option.

   Alternatively, you can click the row arrow corresponding to the particular repository. When the repository row gets selected, click **Provision**.

   The following message appears:

   Provisioning will modify the repository database by adding new tables and adding or updating of records in the existing tables. It is highly recommended to take a backup of the repository database before provisioning, as there is no option to revert the provisioned changes. To revert back the repository database to its previous state, use the Database Restore option. Do you want to Continue?
3. Select **Yes** to provision the workflow repository.

   The repository provision is successful.
4. Click **Close**.

To UnProvision a Provisioned Repository:

To unprovision an already provisioned repository, right-click on any cell column of the particular repository, and then select the **Unprovision** option.

OPC UA Connections

To create or establish a new connection to OPC UA

The OPC UA External Connection can be created through the OPC UA Connection screen. You can use this screen to create or establish new OPC UA connection in AVEVA Work Tasks which will communicate to the OPC UA installation.

Prerequisites:

Provision a repository to create OPC UA connections.

To create an OPC UA Connection

1. Log on to the **Enterprise Console**.
2. Access the repository which is provisioned for OPC UA.
3. In the Enterprise Console, click menu icon, click **Tools**, and then click **External Connections**.

   The **External Connections** page appears.
4. On the **Action bar**, click **New**, and then **New OPC UA Connection**.

   The **OPC UA Server Connection** screen appears.
5. Enter the required details.
   - Title: Unique connection name.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Server URL: .OPC UA server URL.
- Authentication Settings: Select the Authentication Settings required for establishing the connections. The Authentication Settings can be set as Anonymous or User Name.
  - Anonymous - Anonymous tokens are associated with servers which do not require user authentication.
  - User Name/Password tokens are associated with servers with any password based system, such as Windows
- User ID: User ID.
- Password: Password.

6. Click **Test Connection**.

   If successful, click **Save** to save the connection.

Activity: OPC UA Tag Read Write Activity

Use the **OPC UA Tag Read Write** activity to read or write the tag values from and to the OPC UA server which is connected to the AVEVA Work Tasks Server.

Activity Properties:

Specify appropriate values for the activity properties to configure the **OPC UA Tag Read Write** activity.

Right-click the activity, and click **Properties** to open the **Activity Properties** window. Alternatively, select the activity, and then click **Activity Properties** in the tool bar of the **Process Designer** window.

The activity properties of the **OPC UA Tag Read Write** activity are described below.

Name & Description

Use these properties to specify the name and description for the activity.

- **Name**: Specify a name for the activity.

  Property Type: Mandatory
- **Description**: Specifies a brief description to be displayed for the activity.

  Property Type: Optional. Set this property only if required.

Configuration

Use these properties to configure the functionality of the activity.

- **OPC UA Server Connection**: Use this property to select an OPC UA Server Connection. The drop-down list displays the list of all the OPC UA Server Connections created through the External Connection List.

  Property Type: Mandatory. This property must be set for the workflow to communicate with OPC UA Server.

  For more information, see OPC UA Server Connection.
- **Tag Operation**: Use this property to set the type of operation, either Read or Write, to be performed on the OPC UA Server.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Mandatory. This property must be set for the workflow to communicate with OPC UA Server.

- **Select Attribute**: Use this property to read or update the value of variables and XML variables, that can be used in the subsequent activities. The type of operation that can be performed using the Select Attribute property depends on the value selected in the Tag Operation property.

  Property Type: Mandatory. This property must be set for the workflow to communicate with OPC UA Server.
- **Retry Count**: Use this property to specify the maximum number of attempts the activity should make to connect to the OPC UA Server, if the first attempt to connect to the OPC UA Server fails. The default value is 3.

  Property Type: Optional. Set this property only if required.

Note: For the Write operation, the tags can be dynamically added to the Variables/XmlVariables. You can select these variables in the "Select Attribute" property.

Exception Behavior

- **Raise Error on unhandled linked output**: Each activity has the Raise Error on Unhandled Linked Output property. This property shows all the default configured mapped error outputs in red with their check boxes selected.

  Property Type: Optional. Set this property only if required.

  See Mapped Error Output for more details.

Looping Behavior

- **Loop through this action if output is**: Use this property to specify the looping behavior of the activity.

  Property Type: Optional. Set this property only if required.

OPC UA Server Connection

1. In the **OPC UA Tag Read Write** activity properties window, click **OPC UA Server Connection**.

   The OPC UA Server Connection configuration window appears.
2. Select a Connection from the drop-down list.

   The drop-down list shows only the connections in the External Connection List of type OPC UA Server Connections.
3. Click **Save**.

Select Attribute

Select Attribute Property window

You can select the variables or XML variables and set the values to them or read values from them in this property window. To open this window, click the button for the Select Attribute property.

Tag Operation is selected as Read

If the Tag Operation is selected as Read, you can read OPC UA server values into variables or XML variables of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

workflow.
On the left-hand side, a variable can be used to store the attribute value. The variable and type can be added from Variables property in Start activity.
Select the attribute from which the read action needs to be performed in the right-hand side. For the 'Read' operation, the OPC Client Browser will be displayed on the right-hand side.

Tag Operation is selected as Write

If the Tag Operation is selected as Write, you can update variables or XML variables of the workflow with OPC UA server values.
For write operation, the OPC Client Browser will be displayed on the left side of the screen. Click this link to populate the OPC Client Browser.
Select the attribute to which the write action needs to be performed in the left hand side. On the right hand side use expression builder or a variable to get value. The variable and type can be added from Variables property in Start activity.

Mapped Error Output

Each activity has the Raise Error on Unhandled Linked Output property clicking on which, all the default configured mapped error outputs in red with the respective check boxes selected will be displayed.

Mapped Error Outputs

The Mapped Error Outputs list contains the error outputs (shown in red) that cause the Workflow to complete abruptly. This can happen if the error outputs are not linked to the next activity in the Workflow. To change the mapped error outputs, select or de-select the relevant options and click Ok.

Workflow Behavior for Error Outputs

If the error outputs are not linked to the next activity in the Workflow, and if the 'Stop with Error' option is selected, the Workflow will complete with errors. Similarly, if the 'Exit Activity with Error Output' option is selected, the Workflow will not finish with errors, but will give an Error Encountered output.

Note: The changes made to this property will take effect only if you set the value for Raise Error on Unhandled Linked Output to Yes in the Start Activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Developer Guide

This section is for process developers who use AVEVA Work Tasks to create and implement business processes. It assumes that you have basic knowledge of business process design and are familiar with Business Process Management Notation (BPMN).

Get Started

This section provides information to help you get started on AVEVA Work Tasks. Complete the prerequisites as required to access the AVEVA Work Tasks features.

Prerequisites to Access AVEVA Work Tasks Features

Complete the following prerequisites to access the **AVEVA Work Tasks** features:

1. Install **AVEVA Work Tasks**.
2. Create a **AVEVA Work Tasks** repository.
   The AVEVA Work Tasks Repository Database contains workflow definitions, queues, calendars, settings, and workflow execution data for an application. For example, if you define workflows for your organization, the organization must be defined as a <your organization name> Repository. You can create Repositories from the Central Configuration Site.
   For more information on how to create and manage repositories, see *Administrative Guide*.

Installation and Administration

The *Administrator Guide* provides detailed instructions for:

- Product installation and activation
- Product configuration
- Setting up repositories
- Common system administration procedures and uninstallation

Interfaces and Features

This section provides information on the user interfaces and management features of AVEVA Work Tasks.

- Overview of User Interfaces
- Access and Manage AVEVA Work Tasks Features

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Overview of User Interfaces

The AVEVA Work Tasks functions are performed through the following interfaces:

Workflow List: The Workflow List is a list of all the workflows defined for a repository. The Workflow List provides options for categorizing, designing, and publishing workflows. Options in the Workflow List are wizard-driven, with wizards provided for all the core activities.

For more information, refer to the **Workflow List** topic under the **AVEVA Work Tasks** chapter in the *User Guide*.

Process Designer: The Process Designer provides a Visio-like interface with a drag-and-drop functionality for process modeling. For more information on the **Process Designer**, refer to the *About the Process Designer* topic under the *Working with the Process Designer* chapter in the *User Guide*.

Queue Management: The Queue Management window allows you to create and manage queues for allocation of workflow activities. For more information about **Queues**, refer to the *About Queues* topic under the *Working with Queues* chapter in the *User Guide*.

Calendar: The Calendar interface enables creation of hierarchical organization and resource calendars. Calendars are used to determine availability of resources, calculate timeouts, and perform other scheduling functions. For more information about working with **Calendars**, refer to the *About Calendars* topic under the *Working with the Calendars* chapter in the *User Guide*.

Work Items List: The Work Items List is a web-based interface to which workflow activities are delivered. For more information about the **Work items list**, refer to the *About Work Item List* topic under the *Working with the Work item List* chapter in the *User Guide*.

Business Activity Monitoring Dashboard: The **BAM Dashboard** enables monitoring of workflow execution and provides an array of real-time reports on workflow statuses and performance. For more information about using the **Business Activity Dashboard**, refer to the *About the BAM* topic under the *Business Activity Monitoring & Reports* chapter in the *User Guide*.

Access and Manage AVEVA Work Tasks Features

You can access the AVEVA Work Tasks features and interfaces using the following:

Enterprise Console

The Enterprise Console is a central portal with options for **Workflow design**, **Workflow Management**, **User Creation**, **BAM Analysis**, **Work Items Lists**, **Queues** and **Calendar Management**. The **Enterprise Console** is mapped to one or more **Repositories**. You can login to the **Enterprise Console** to access all the above functions of a specified repository. The **Enterprise Consoles** are created from **Farm Server Configuration Manager** and mapped to repositories in the **Repository Settings** page of the **Central Configuration**.

For more information on creating and configuring **Enterprise Consoles** and **Repositories**, refer the *Administrator Guide*.

For more information on working with the **Enterprise Console**, see *Enterprise Console* chapter in the *User Guide*.

Embed AVEVA Work Tasks Controls in Applications

AVEVA Work Tasks Controls for Process Designer, Queue, Calendar, BAM, and Work Item List can be embedded into an ASP.NET application.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For more information about setting up the development environment, and customizing AVEVA Work Tasks for use in applications, see *Developer Guide*.

AVEVA Work Tasks Services

Before you use AVEVA Work Tasks, ensure that the following services are running:

Server Installation

- **Workflow Engine**: This service is required for the execution and control of workflows.
- **Task Scheduler**: This service handles the execution of all scheduled tasks.
- **Communication Service**: This service handles incoming Email and SMS communication. In addition to the above services, the following service is required in the case of Enterprise Server installations.
- **Advance Server Service**: This service provides Load Balancing and Failover. This service will be enabled only if Load Balanced Server license is acquired.
- **Mobile Notification Service**: This service handles notifications sent to the AVEVA Work Tasks Pro mobile application.
- **Client Service**: This service is required to install, start, and monitor AVEVA Work Tasks services. In case of an Enterprise Server, this service will also install, start, and monitor Advance Server Service.

Client Installation

- **Advance Server Service** - This service provides Load Balancing and Failover. This service will be enabled only if Load Balanced Server license is acquired.
- **Client Service**: This service is required to install, start, and monitor AVEVA Work Tasks services. In case of an Enterprise Server, this service will also install, start, and monitor Advance Server Service.

Enterprise Console

For information about Using Enterprise Console, see *User Guide*.

Message Queue

The **Message Queue Connections** option allows you to configure the connection and queue details in the Enterprise Console. This can be used in Workflow Association and Event Provider.

The **Message Queue Repository Provisioning Tool** allows you to provision a workflow repository for Message Queue.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Message Queue Connections

The **Message Queue Connections** allows you to configure the connection and queue details in the Enterprise Console. This can be used in Workflow Association and Event Provider.

Note: Message Queue application must be installed and AVEVA Work Tasks must be configured with a provisioned repository.

To configure Message Queue connection details

1. Log on to the Enterprise Console.
2. On the menu, click **Tools** and then click **Message Queue Connections**.

   The **Message Queue Connections** window appears.
3. Click **New** and then click **Item** to create a new Message Queue connection item.

   The **Message Queue Connection** window appears.

Field Name	Description
Title	Enter the title for the connection.
Description	Enter the description for the connection.
Connection Type	Select **Send** option to send the message to the message queue or **Receive** option to receive message from the message queue.
Durable	Select **Yes** or **No** based on the **Durable** feature of the message queue.
Binding Type	Select either **WM Option** or **XML** or **JSON** based on the type of message sent to or received by the message queue.
User Name	Enter the user name with which you want to configure connection.
Password	Enter the password.
Additional Settings	Enter any additional settings.
Queue Name	Enter the name of the message queue.

4. Click **Save** to save the configured settings.

Note:
- After creating a new connection or editing the existing connection, restart the Workflow Engine to utilize the connection.

- To execute Quickflow, ensure to enable *rabbitmq_amqp1_0* plugin. To enable the plugin, run the below command in Message Queue Command Prompt (sbin dir) with Administrator mode:
`rabbitmq-plugins enable rabbitmq_amqp1_0`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- In the upgraded environment, the existing Message Queue Connection will have **Connection Type** as **Send**, **Binding Type** as **WM Option**, and **Durable** set to **Yes** as the default values.

- The Message Queue Connection is validated with RabbitMQ versions 3.10 and 3.10.5.

- While adding Virtualhost in Additional Settings, only root (/) is verified and supported. Custom Virtualhost is not supported.

Message Queue Repository Provisioning Tool

You can use the **Message Queue Repository Provisioning Tool** to provision a workflow repository for Message Queue. Only the provisioned repositories can be connected to Message Queue. Workflow repository provisioning for Message Queue should be done only after installing AVEVA Work Tasks.

The version of the Message Queue is 3.10.5.

Note:
In the upgrade environment, the new features added to the Message Queue will not work for the already provisioned repositories. For the new features to work, the repositories should be unprovisioned.

When you unprovision the repository,
- The Message Queue connections and the associations will be deleted.
- The workflow containing the Message Queue Activities is not modified and the Message Queue Activities should be manually removed for the workflow to work properly.

To provision a Repository for Message Queue

1. Run the **AVEVAWorkTasksMQProvision.exe** as an Administrator. This EXE is available in the **<Installed Path>\Bin** path. A window displaying all the repositories from the system appears.

   If you are not using the default Enterprise Console site, copy the **SE.Bpm.RabbitMQ.dll** from the Bin folder of the default Enterprise Console site to the Bin folder of your custom Enterprise Console site.

   The DLLs are available in the **<Installed Path>\Bin** path.
2. Right-click on any cell column of the particular repository, and then select the **Provision** option.

   Alternatively, you can click the row arrow corresponding to the particular repository. When the repository row gets selected, click **Provision**.

   The following message appears:

   Provisioning will modify the repository database by adding new tables and adding or updating of records in the existing tables. Do you want to Continue?
3. Select **Yes** to provision the workflow repository.

   The repository provision is successful.
4. Click **Close**.

   During provisioning, if you have already logged in to the repository, then logout and login to the repository after provisioning the repository to Message Queue.

To unprovision a Provisioned Repository

To unprovision an already provisioned repository, right-click on any cell column of the particular repository, and then select the **Unprovision** option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To configure Listener:

You can create and configure your own listener using the **Configure Listener** option. To create listener assembly, implement the **SE.Bpm.RabbitMQ.Interfaces.IRabbitMQConsumer** interface from **SE.Bpm.RabbitMQ.dll**. Follow the below steps to configure the custom listener.

1. In the **Message Queue Repository Provisioning Tool**, click **Configure Listener**.
   The **Configure Message Queue Listener Details** dialog box appears.
2. Enter the following details:
   Assembly Path: Enter the path of the custom dlls to configure custom listener.
   Class Name: Enter the class name (which implements the **SE.Bpm.RabbitMQ.Interfaces.IRabbitMQConsumer** interface from **SE.Bpm.RabbitMQ.dll**).
3. Click **Update** to change the listener configuration details.

Note: If the repository is already provisioned to Message Queue, ensure to unprovision the repository before configuring the Message Queue listener.

In the upgrade environment, the Listener Configuration screen displays new assembly details (e.g. SE.Bpm.AMQP.Listener.AMQPListener). But this will work only after you unprovision all the repositories and provision the required repository.

MQTT Connections

The **MQTT Connections** allow you to configure connection details to MQTT, based on WebSockets, in the **Enterprise Console**. These connections then can be used to access data from external systems.

The tested Message Broker versions are:

- RabbitMQ version 3.10.5
- Mosquitto version 1.5.4

To configure MQTT Connection Details

1. Log on to the **Enterprise Console**.
2. On the menu, click **Tools** and then click **MQTT Connections**.
   The **MQTT Connections** page appears.
3. Click **New** and then click **Item**.
   The **New MQTT Connection** window appears.
4. Enter the required connection information, and then click **Save**.
   The configuration is saved, and an information message appears.
   Note: See *Connection Information* for more details.
5. Click **Close**.
   The information message is closed.

Connection Information

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The connection information required to create an MQTT Connection is as follows.

Field Name	Description
Name (Mandatory)	Name for the MQTT Connection.
Description	Descriptive name for the MQTT Connection.
Server (Mandatory)	IP address of the MQTT server.
Port (Mandatory)	Port number of the Web Socket protocol in the MQTT server. The default value is **8080**. **Note**: The Broker should be enabled with Web Sockets.
Secure Connection	Allow or disallow secure WebSocket. • Select **Yes** to allow secure WebSocket using TLS. • Select **No** to disallow secure WebSocket. The default value is **No**. **Note:** For secure WebSocket, ensure to install CA Root Certificate on client under Trusted Root Certification Authorities. If the Enterprise Console Web Application is configured with secured protocol, you need to select **Yes** for the MQTT Connection to work properly. When the Enterprise Console Site is configured as Secured (HTTPS protocol), then the MQTT connections would work only with Secured configuration.
WebSocket Endpoint (Mandatory)	Enter the WebSocket Endpoint. For example, **mqtt** or **ws**. The default value is **mqtt**.
Client ID Prefix	Client ID prefix for the MQTT Connection, if required.
Anonymous	Allow or disallow all users. • Select **Yes** to allow all users. In such cases **User Name** and **Password** are not required.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description
	• Select **No** to disallow all users. In such cases **User Name** and **Password** are required. The default value is **No**.
User Name (Mandatory)	User ID for the MQTT Connection. **Note:** User Name is not available if **Anonymous** is set to **Yes**.
Password (Mandatory)	Password for the MQTT Connection. **Note:** Password is not available if **Anonymous** is set to **Yes**.
MQTT Version	Version of the MQTT. The default value is **3.1.1**.
Reconnection Attempts (Mandatory)	Number of times to try connecting when the MQTT Connection fails. The range is between **-1** to **20**. If the value is set to **-1**, then there is no limit to the number of attempts, and if the value is set to **0**, then there will be no attempts. The default value is **3**.
Reconnection Period (Mandatory)	Time (in milliseconds) between the connection attempts when the MQTT Connection fails. The range is between **10000** to **300000**. The default value is **30000**.
Quality of Service	Quality of service for the MQTT Connection. • **At Most Once**: The message is sent once. The message is also deleted, and no duplicate messages are possible. • **At Least Once**: The message is sent multiple times. The message is also stored, and duplicate messages are possible. • **Exactly Once**: The message is sent and acknowledgement is awaited. The message is also stored, and no duplicate messages are possible. The default value is **Exactly Once**.
Update Rate (Mandatory)	Time (in milliseconds) at which data is updated. The range is between **0** to **3600000**. If the value is set

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description
	to **0**, then the data is updated immediately. The default value is **0**.

To Test an MQTT Connection

1. In the **MQTT Connection** window, click **Test Connection**.
   The configuration is tested, and an information message appears.
2. Click **Close**.
   The information message is closed.

Note: When packaging artifacts that make use of MQTT Connections, the connections should be recreated at the destination.

Logic App Connections

The **Logic App Connections** allow you to configure connection details to Logic Apps in the **Enterprise Console**. These connections then can be used to call **Logic App Workflows**.

To configure Logic App Connection Details

1. Log on to the **Enterprise Console**.
2. On the menu, click **Tools**,and then click **Logic App Connections**.
   The **Logic App Connections** page appears.
3. Click **New** and then click **Item**.
   The **New Logic App Connection** window appears.
4. Enter **Title**, **Description**, and **Connection String**, and then click **Save**.
   The configuration is saved, and an information message appears.
   Note: Enter a valid connection string as the connection string is not validated.
5. Click **Close**.
   The information message is closed.

Note: When packaging artifacts that make use of Logic App Connections, the connections should be recreated at the destination.

Controls

The following controls are found in AVEVA Work Tasks:

- **Process Designer Control**: The Process Designer provides an interface to design the workflow processes.
- **Work Items List Control**: The Work items list allows the workflow actors to view and respond to the work items assigned to them in a user interface that is similar to Outlook.
- **Calendar Control**: The Calendar control allows you to define and configure multi-level organizational

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

calendars.

- **BAM Control**: The Business Activity Monitoring (BAM) control provides features that can be used by a process administrator to monitor and control various instances of the workflow(s) in a repository.
- **Queue Control**: The Queue control is used to group resources as a common pool and assign tasks to the group.
- **List Control**: List Controls lists out the Workflows, Navigation Details, Users, Event Providers and Event Associations.
- **Form List Webpart Control**: Forms List Webpart Control lists out the forms available.
- **Workflow Execution View Control**: The WorkflowExecutingReport Control shows the Process Execution View of the currently executing workflow.

Add Controls in the Application

AVEVA Work Tasks provides out-of-box server controls. You can configure the properties of AVEVA Work Tasks server controls to suit diverse applications.

The controls should be added using a panel.

To add Controls using a Panel

1. Drag and drop a panel.
2. In the underlying code, create a new instance of control and add the instance to the panel as in the example below:

```
Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject("Telematics");//Repository name
object loggedinUserId = "2d52b2c0-4cf8-4af6-acad-a0c9908eeab5";
Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext(loggedinUserId,
application, "skeltalist");
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl WIL = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
WIL.ApplicationName = uContext.Repository.ApplicationName;
WIL.UserIdString = uContext.LoggedInUserId;
WIL.ID = "WorkItemListControl";
WIL.DisplayGroupUI = true;
WIL.Height = Unit.Percentage(100);
WIL.ReadingPaneInPopupWindow = true;
Panel1.Controls.Add(WIL);//add control to the panel
```

Note:
- To load all the controls separately on aspx page, set the user context first. For example: Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext(loggedinUserId, application, "skeltalist").
- When utilizing these controls embedded in a custom application, a separate Client Connection License is used upon accessing the custom application.

Server Controls

AVEVA Work Tasks provides the following server controls out of box:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Process Designer Control
2. Work Items List Control
3. Calendar Control
4. BAM Control
5. Queue Control
6. List Control
7. Workflow Execution View Control

Create an ASP.NET Project in Visual Studio .NET

You can create an ASP.NET web site in Visual Studio and make the required changes to successfully embed the AVEVA Work Tasks controls in the application.

To create a Web Site

1. Open **Microsoft Visual Studio**. The **Visual Studio Start Page** appears.
2. On the **File** menu, point to **New**, and then click **Web Site**. The **New Web Site** dialog box appears.
3. In the **Templates** pane, select **ASP.NET Web Site**.
4. Select **HTTP** as the **Web Location**, and browse the path where the new web site has to be created. For example, *http://localhost:9000/TelematicsWebSite*.
5. Click **OK**. A new web site gets created.
6. Copy the following files and folders from the *AVEVA Work Tasks Installed Path* to the created web site's folder:
   a. Copy the **BPMUITemplates** folder from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\BPMUITemplates* to the created web site's folder.

      For example: If the TelematicsWebSite web site is created in *D:\Develop\TelematicsWebSite* folder, then copy **BPMUITemplates** folder to *D:\Develop\TelematicsWebSite*.

      Or

      Create a Virtual Directory in the name of **BPMUITemplates** mapping to the physical folder of BPMUITemplates under the created IIS WebSite.

      This step is required due to the following reasons:

      a. ASP.NET applications that use AVEVA Work Tasks should include these files.
      b. BPMUITemplates folder assists in the localization of the AVEVA Work Tasks controls.
      c. The User Interface can be changed according to the user.
      d. Changes to User controls like Work Item, Queue, Process Designer, BAM, and Calendar can be done in
      e. BPMUITemplates folder.

   b. Copy the **App_Code** folder from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\EnterpriseConsole* to the created web site's folder.
      For example, if the WMTestSite web site created in D:\Develop\WMTestSite folder, then copy App_Code folder to D:\Develop\WMTestSite\.
   c. Copy the **ComponentArt.Charting.lic** and **ComponentArt.Charting.WebChart.dll** file from the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\EnterpriseConsole\bin* to the created website's bin

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

folder.
Create a bin folder under the created web site's folder, if it does not exist.

d. Copy the **TempReports** folder from *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**EnterpriseConsole* to the created web site's folder.

e. Copy the **FrameworkData** folder from *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**EnterpriseConsole* to the created website's folder.

7. Create the following .aspx pages for embedding AVEVA Work Tasks Web Server controls.
   - Designer.aspx for Process Designer control
   - WorkItemList.aspx for Work items list control
   - Calendar.aspx for Calendar control
   - BAM.aspx for BAM control
   - Queue.aspx for Queue control
   - List.aspx for List control
   - FormsWebpart.aspx to view Fill a form control
   - Execute.aspx to execute the workflow
   - ExecuteReport.aspx to view the workflow status

   Note: Ensure that all the pages created above have the doctype tag **<!DOCTYPE html>**.

8. Add the following tag in the source of all the .aspx pages, except *Designerbase.aspx*, under the existing <DIV>tag:

```
<asp:Panel ID="Panel1" runat="server" Height="100%" Width="100%"></asp:Panel>
Add the style tag for <DIV>tag:
<div style="height:100%;">
Add the following style on each page within the <head> tag.
<style type="text/css''>
html, body, form
{
height: 100%;
}
</style>
```

9. Complete the following changes in the Web.config file of the custom application website or copy the Enterprise Console Web.config file to the custom application website.
   - Add the following entries within the **<system.web>** tag.

```
<pages validateRequest="false" clientIDMode="AutoID"
controlRenderingCompatibilityVersion="3.5"/>
<httpRuntime requestValidationMode="3.5"
encoderType="HtmlAttributeEncodingNot" maxQueryStringLength="9000"
enableVersionHeader="false" enableHeaderChecking="false"/>
```

   - Add the following entries within the **<system.web>** tag.

```
<compilation debug="false" targetFramework="4.8">
<assemblies>
<add assembly="System.Transactions, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089"/>
</assemblies
</compilation>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

10. Copy all the dll files which has entries in the web.config file of Enterprise Console to the custom site bin directory from AVEVA Work Tasks installed folder bin directory. Alternatively, you can add all the entries under the <assemblies> tag within Enterprise Console Web.Config to file to the custom site web.config file.
    - Add the following entries within the **<system.web>** tag.

```
<membership>
<providers>
<add name="SkeltaMemberShipProvider" type="SkeltaMemberShipProvider" />
</providers>
</membership>
<webParts>
<personalization defaultProvider="SkeltaPersonalizationProvider">
<providers>
<add name="SkeltaPersonalizationProvider"
type="SkeltaPersonalizationProvider.SkeltaPersonalizationProvider" />
</providers>
<authorization>
<allow users="*" verbs="enterSharedScope" />
</authorization>
</personalization>
</webParts>
<customErrors mode="RemoteOnly" defaultRedirect="error.html" />
```

    - Add the following entries within the **<httpHandlers>** tag.

```
<add verb="*" path="Reserved.ReportViewerWebControl.axd"
type="Microsoft.Reporting.WebForms.HttpHandler,
Microsoft.ReportViewer.WebForms, Version=11.0.0.0, Culture=neutral,
PublicKeyToken=89845dcd8080cc91" validate="false" />
<add path="Telerik.Web.UI.WebResource.axd" type="Telerik.Web.UI.WebResource"
verb="*" validate="false"/>
```

    - Add the following entries within the **<system.webServer>** tag.

```
<validation validateIntegratedModeConfiguration="false" />
<httpProtocol>
<customHeaders>
<remove name="Server" />
<remove name="X-AspNet-Version" />
<remove name="X-AspNetMvc-Version" />
<remove name="X-Powered-By" />
</customHeaders>
</httpProtocol>
<security>
<requestFiltering>
<requestLimits maxQueryString="9000"/>
</requestFiltering>
</security>
<staticContent>
<remove fileExtension=".woff" />
<mimeMap fileExtension=".woff" mimeType="application/font-woff" />
</staticContent>
```

    - Add the following entry within the <system.webServer><handlers> tag.

```
<add name="ReportViewerWebControl" verb="*"
path="Reserved.ReportViewerWebControl.axd"
type="Microsoft.Reporting.WebForms.HttpHandler,
Microsoft.ReportViewer.WebForms, Version=12.0.0.0, Culture=neutral,
PublicKeyToken=89845dcd8080cc91" />
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Add the following entries within the <system.web.extensions> tag.

```
<scripting>
<webServices>
<jsonSerialization maxJsonLength="4096000" />
</webServices>
</scripting>
```

11. Right-click the hosted website, go to **Manage Application**, and then click **Advanced Settings**. The **Advanced Settings** dialog box appears.
12. It is recommended to set the Application Pool property to **AVEVA Work Tasks Integrated**.

    If you use any other application pool or service to access Work Tasks, you need to add them to one of the following groups:

    - Administrators
    - aaAdministrators
    - aaConfigTools

      For more details, refer to Cryptography Settings.

Note:
- Ensure to copy **NewtonsoftJson.dll** and **AntiXSSLibrary.dll** files from the **Bin** folder of the AVEVA Work Tasks application to the **Bin** folder of the custom application.

- Register the Telerik tag in all the mentioned aspx pages.

```
<%@ Register TagPrefix="telerik" Namespace="Telerik.Web.UI" Assembly="Telerik.Web.UI" %>
```

- Add the following entry under the form tag of mentioned aspx pages.

```
<telerik:RadScriptManager runat="server" ID="RadScriptManager1" />
```

Create User Context

For every control, the AVEVA Work Tasks User Context object must get instantiated. Find below the steps to create User Context Object.

To create User Context Object

1. Before creating user context object, Create an ASP.NET Project in Visual Studio .NET.
2. Go to **UserContext.aspx.cs** page.
3. Create a user context object.

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
public partial class CreatingUserContext : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
Work Tasks.Entity.UserContext userContext = new Work Tasks.Entity.UserContext("john",
new Skelta.Core.ApplicationObject("LeaveReportsRepo"), "activedirectory");

}
}
```

The **UserId** value varies based on the authentication provider configured. This value should have the Entity instance mapped for the repository to be prefixed along with real id of the user.

- If the repository is mapped to **AVEVA Work Tasks List** user provider, the UserIdString will be skeltalist::[GUID]. Here the GUID is of the user "john", taken from SKEUsers table present in the Repository DB.

Example: skeltalist:: C99B4110-C8CC-410B-BF70-93F3A7E34EDF

That is, User object will be created as:

```
Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext("C99B4110-C8CC-410B-
BF70-93F3A7E34EDF", new Skelta.Core.ApplicationObject("LeaveReportsRepo"), "skeltalist");
```

- If the repository is mapped to the **Active Directory** user provider, the UserIdString will be activedirectory::[samAccountName]. Here, samAccountName is an active directory property value.

For example: activedirectory::john

That is, the user object will be created as:

```
Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext("john", new
Skelta.Core.ApplicationObject("LeaveReportsRepo"), "activedirectory");
```

- If the repository is mapped to the **Forms** user provider, the UserIdString will be sqlprovider::[UniqueIdentifier]. Here UniqueIdentifier is the unique identifier id for resource.

For example: sqlprovider::6fb03011-68b7-437e-972e-21c67e4e2fe7

That is, the user object will be created as:

```
Skelta.Entity.UserContext userContext = new
Skelta.Entity.UserContext("6fb03011-68b7-437e-972e-21c67e4e2fe7", new
Skelta.Core.ApplicationObject("LeaveReportsRepo"), "sqlprovider");
```

- If the repository is mapped to **ASP.NET Membership** user provider, the UserIdString will be membershipprovider::[UniqueIdentifier]. Here, UniqueIdentifier is the unique identifier id for resource.

For example: membershipprovider::a9bfc969-cc82-4c4b-99b9-50d9e1a91c01

That is, the user object will be created as:

```
Skelta.Entity.UserContext userContext = new
Skelta.Entity.UserContext("a9bfc969-cc82-4c4b-99b9-50d9e1a91c01", new
Skelta.Core.ApplicationObject("LeaveReportsRepo"), "membershipprovider");
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Process Designer Control

The Process Designer provides an interface to design the workflow processes. Workflows can be designed and deployed using the Process Designer.

To add Process Designer control using a panel

Before adding the Process Designer control in a panel, create an ASP.NET project in Visual Studio .NET 2012. You must add two .aspx pages for this control, one to embed the control and other to create the list definitions and call the Designer.aspx page.

- Designerbase.aspx
- Designer.aspx

1. Open the **Designerbase.aspx.cs** page.
2. Add the following code in the page load of this page.

```
string applicationName = " ApplicationName ";
string listName = "Workflow";
string workflowName = "WorkflowName"; // Use unpublished workflow Name
Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject(applicationName);
// Create User context as given in this topic http://sun.skelta.com/DeveloperGuide/
CreatingUserContext.html
Skelta.Entity.UserContext usrcn = new Skelta.Entity.UserContext("47D5EA43-B471-492C-
A2C3-41DC41C15EC8", application, "skeltalist", "", "", false);
Skelta.Repository.Web.List.ListPageParameters param = new
Skelta.Repository.Web.List.ListPageParameters(); param.ListName = listName;
Skelta.Repository.List.ListDefinition list =
Skelta.Repository.List.ListDefinition.GetList(application, listName);
param.ListId = list.Id;
Skelta.Repository.List.ListDataHandler ldh = new
Skelta.Repository.List.ListDataHandler(applicationName, listName);
param.ListItemId = ldh.GetListItemId(workflowName);
param.ApplicationName = applicationName;
//specify the version information //param.VersionStamp = "1";
param.LoggedInUserId = usrcn.LoggedInUserId;
string encParam = param.GetSecuredUri();
//Give the page name where the Process Designer control is placed.
string url = "Designer.aspx"; // specify the page where ProcessDesigner is embedded
if (url.IndexOf("?") > 0)
url += "&" + encParam;
else
url += "?" + encParam; this.Context.Response.Redirect(url);
```

3. Open the **Designer.aspx.cs** page.
4. Ensure that you have initialized the User Context object, as per the steps given in Create User Context
5. In the underlying code, create a new instance of control and add the instance to the panel in the Page_Load, as shown in the sample code:

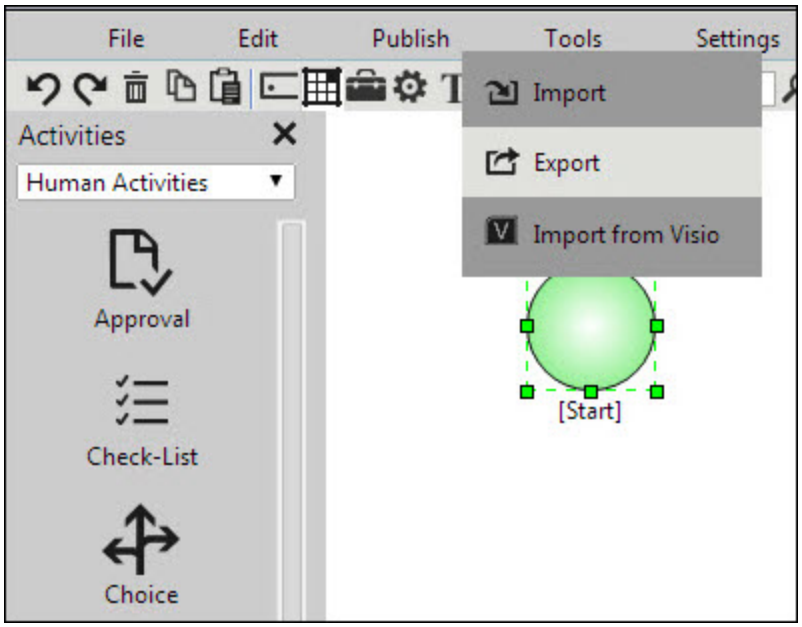
```
string applicationName = "Telematics Repository";
Skelta.Core.ApplicationObject application = new Skelta.Core.ApplicationObject
(applicationName);
Skelta.Entity.UserContext usrcn = new Skelta.Entity.UserContext
("81734457-B3DB-4BC9-894A-C3149828C65C", application, "skeltalist", "", "", false);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Workflow.NET.Web.Designer.ProcessDesigner pd = new
Workflow.NET.Web.Designer.ProcessDesigner();
pd.ApplicationName = usrcn.Repository.ApplicationName; //Provide Application or
Repository name. This is mandatory.
pd.WorkflowName = "Test Workflow"; //Provide Workflow name. This is mandatory.
pd.UserIdString = usrcn.LoggedInUserId; //Provide user id string. This is mandatory.
pd.ID = "ProcessDesigner"; // This is mandatory.
pd.Height = Unit.Percentage (100);
Panel1.Controls.Add (pd);
```

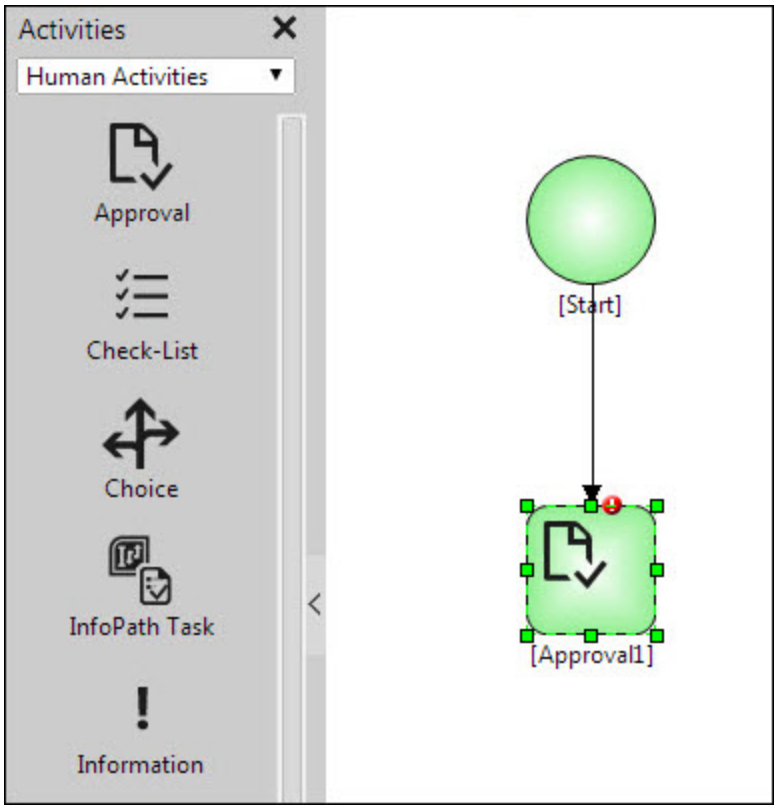
Properties that can be set for Process Designer Control

- **ApplicationName:** Provide the Application or Repository Name.
- **CanExportProcessDefinition:** If the CanExportProcessDefinition property is set to false, then the "**Export**" option will not be displayed and you cannot export workflows.



- **CanImportProcessDefinition:** If the **CanImportProcessDefinition** property is set to false, the **Import** and **Import from Visio** options will not be displayed and you cannot import workflows.
- **CanWriteInProduction:** If the **CanWriteInProduction** property is set to false, you can design the workflow but will not be able to publish it.
- **WorkflowName:** Provide the workflow name to design the workflow.
- **FileName:** This property should not be used since it is an obsolete property; instead of the **FileName** property, you can use the **Version** property.
- **CanWriteInStaging:** If the **CanWriteInStaging** property is set to false, you can only view the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



To use Process Designer as an Embedded Control in Custom Applications

Consider a scenario where a user creates a custom application and uses the Process Designer control as an embedded control. In such a scenario, the user will have to follow the steps given below in order to publish/ execute the workflow.

1. Repeat the steps given under the section **To add Process Designer control using a panel**.
2. Create another page in the Visual Studio and add the following code sample. Alternatively, if the page is already created through which the user can navigate to the Process Designer page, then the user has to use the code snippet given below in the event which calls the Process Designer page. For example, the event used here is 'Button_Click'.

```
protected void Button _Click(object sender, EventArgs e)
    {
        string applicationName = " ApplicationName ";
        string listName = "Workflow";
        string workflowName = "WorkflowName";
        Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject(applicationName);
// Create User context as given in this topic http://sun.skelta.com/DeveloperGuide/
CreatingUserContext.html
        Skelta.Entity.UserContext usrcn = new
Skelta.Entity.UserContext("47D5EA43-B471-492C-A2C3-41DC41C15EC8", application,
"skeltalist", "", "", false);
        Skelta.Repository.Web.List.ListPageParameters param = new
Skelta.Repository.Web.List.ListPageParameters();
        param.ListName = listName;
        Skelta.Repository.List.ListDefinition list =
Skelta.Repository.List.ListDefinition.GetList(application, listName);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
        param.ListId = list.Id;
        Skelta.Repository.List.ListDataHandler ldh = new
Skelta.Repository.List.ListDataHandler(applicationName, listName);
        param.ListItemId = ldh.GetListItemId(workflowName);
        param.ApplicationName = applicationName;
 //specify the version information
 //param.VersionStamp = "1";
        param.LoggedInUserId = usrcn.LoggedInUserId;
        string encParam = param.GetSecuredUri();
//Give the page name where the Process Designer control is placed.
        string url = "PDesigner.aspx"; // specify the page where ProcessDesigner is
embedded
        if (url.IndexOf("?") > 0)
            url += "&" + encParam;
        else
            url += "?" + encParam;
        this.Context.Response.Redirect(url);
    }
```

3. On click of the button, the **Process Designer** page will appear. The user will be able to perform all the actions related to the workflow.

Work Items List Control

The **Work items list** allows the workflow actors to view and respond to the work items assigned to them in a user interface that is similar to Outlook. Work items are delivered to the actor's Inbox with a detailed view available on selection of the item.

To add Work items list Control using a panel

1. Before adding the Work Items List control in a panel, create an ASP.NET project in Visual Studio .NET 2012.
2. Open **WorkItemList.aspx.cs** page.
3. Ensure that you have created the User Context object as per the steps described in Creating User Context.
4. In the underlying code, create a new instance of control and add the instance to the panel in the **OnInit**, as shown in the example below:

```
Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject("ListRepo");//Repository name
object loggedinUserId = "4952ffff-e0db-401b-bb7d-70a0fec1df07";
Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext(loggedinUserId,
application, "skeltalist");
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl wi = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
wi.ID = "WorkItemControl"; // This is mandatory.
wi.ApplicationName = userContext.Repository.ApplicationName; //Provide Application or
Repository name. This is mandatory.
wi.UserIdString = userContext.LoggedInUserId; //Provide user ID string. This is
mandatory.
wi.Height = Unit.Percentage(100);
wi.Width = Unit.Percentage(100);
wi.ReadingPaneInPopupWindow = true;
Panel1.Controls.Add(wi);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Call the following function within the <body> tag.

```
<body onload="ResizeGrid();">
```

Different properties that can be set for Work items list

The properties for the AVEVA Work Tasks Work items list Web Part are categorized as:

Actions:

- **Allow AutoOwnership on row click**: Selecting this option allows you to take ownership of the work items by clicking on the row.
- **Enable hold**: Selecting this option allows you to put work items on hold. The Hold button would be visible for the work item, only if hold is enabled.
- **Enable Export**: Selecting this option allows you to control the visibility of the export icon in the Work Item details grid status bar.

Display:

- **Show Inbox Details without Left Grid view**: On selecting this option, only the Inbox view without the folder list to the left tree menu will be displayed.
- **Show Manager View**: On selecting this option, only the Manager View for the list will be displayed.
- **Show Filter Bar Always**: Selecting this option enables you to either show or hide the filter bar.
- **Display left tree menu without workflow(s) grouping**: On selecting this option, you can view the Work items list page without grouping the workflow(s).
- **Show Queues**: Selecting this option displays a folder for the Queues that you are participating in.
- **Show Work Item Fields**: Selecting this option displays work item fields.
- **Record Size**: The number of messages displayed in the work item is based on the property specified here. For example, suppose, there are totally 35 messages and this property is set as 20, then 20 messages out of 35 will be displayed in the work items.
- **Retain the Sort Column**: To retain the selection of recently sorted column after submitting the work item and also while expanding the left tree menu view items. If this is set to true, then selection of the recently sorted column is retained. If set to false, then the selection of the recently sorted column is not retained.
- **Queue Name**: Enter the Queue name to be displayed in the left tree menu of the Work Items Web Part. If the Queue name is not specified, all the available queues will be displayed in the left tree menu.

  Multiple Queue names can be specified separated by a semicolon. For example: Manager;Support
- **Layout View Name**: This property loads a specific layout.
- **Show CustomLink in Left Grid Menu**: Selecting this option enables you to display a custom link in the left pane of Work Items Web Part.
- **Show Queues in the LHS grid view menu only if they have work items**: On selecting this option, the queues in the left hand side grid view menu will be displayed only if they have work items.
- **Display the count in the Left Hand Side Grid View**: Enabling this option will display the work items count in the left hand side grid view.
- **Display Classic Paging**: Enabling this option will allow you to display the Classic Paging properties. The navigation feature of Work item Control is not supported with Classic Paging enabled.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Show Standard Columns**: This property displays the work item field columns based on the Workflow Name property of Workitemcontrol.
- **Display the columns with auto fit**: If you have less number of columns to be displayed, enable or check this option. If you have more number f columns to be displayed, disable or uncheck this option. Then you can use the scroll bar to view all the columns.

Multi Select:

- **Enable Multi Select**: Selecting this option enables you to select multiple work items.
- **Show multi select window in center**: This option will display a multi select window in the center of the page.
- **Show detail view with multi select option enabled**: Selecting this option will display the detail view with the multi select option enabled by default.
- **Display multi select option by default for queue items**: Selecting this option will display the queue items with the multi select option by default.

Item Details:

- **Show Reading pane in popup window**: If this option is selected, the work item details will be displayed in a separate window and not in the Reading Pane below the Inbox.
- **ShowDetailsonRowDoubleClick**: Selecting this option enables you to open the Work Item Details page (Custom ASPX page) on double-click.
- **Display Navigation buttons Next and Previous in the Work Item detailed view**: Selecting this option enables you to navigate to the next or previous work item from the Work Item Details view page.
- **Enable Move to the next work item after submitting the work item from the detailed view**: Enabling this option will take you to the next work item when a work item is submitted from the detailed view.
- **Show Queue Detailed View Without Picking**: Selecting this option allows you to act on a work item without picking it.

Filter and Group:

- **Workflow Name**: Enter the workflow name to be displayed in the left tree menu of the Work items Web Part. If the workflow name is not specified, all the available workflows will be displayed in the left tree menu. Multiple workflow names can be specified with semicolon. Example:Leave Application;HRProcess
- **Group Total Columns**: This property is used to calculate the total of the column(s) name when the grouping operation is performed on columns. Multiple column names can be specified with semicolon. Example:Expense Amount,true;Leave Days,false
- **Group Count Columns**: This property is used to calculate the count of the columns when the grouping operation is performed on columns. Multiple column names can be specified with semicolon. Example:Leave Days;Expense Type
- **Group Average Columns**: This property is used to calculate the average of the columns when the grouping operation is performed on columns. Multiple column names can be specified with semicolon. Example:Expense Amount,true;Leave Days,false

DisplayGroupUI

If the DisplayGroupUI property is set to 'false', the option to enable and disable the Multi-select option icon will not be displayed in the Work Item List Control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If multiple work items related to approval activities get displayed in the Work items list, you can select all the check boxes related to the approval activity and then act on all the work items from the GroupUI window.

DisplayLeftTreeMenuAtApplicationLevel

If the DisplayLeftTreeMenuAtApplicationLevel property is set to 'true', then the work items of all the workflows will be displayed at Application Level and there will not be workflow segregation displayed in the Left Tree menu view of Work Item control.

DisplayManagerView

If the DisplayManagerView property is set to 'false', the Manager View will not be displayed. Using this property, the manager can see all the tasks assigned to his/her subordinates (peers) who report to him/her. The Manager can perform the following operations from the Manager view:

- Reassign a work item to another user.
- Delegate a work item to another user.
- Set the priority for the work items assigned to his subordinates (peers).

The Manager View will be displayed only for managers. For this property to function correctly, it is mandatory to have the purpose="manager" attribute set in the entity data source provider as well as at the Entity Level.

Set RecordSize

Using this property, we can control the number of work items that will be displayed per page.

Show Queue

If the ShowQueues property is set to 'true', only the "Queue" icon and the Queue will be displayed in the Left Tree menu view.

If the ShowQueues property is set to 'false', the Queue icon and the Queue will not displayed in the Left Tree menu view.

Work item Fields

If the ShowWorkItemFields property is set to 'true', the work item fields will be displayed in the work items details grid. However, this property will be effective only if you are loading the Work items list control with the Detail view and you have not customized the view by creating Global Layout or Workflow level Layout.

If the ShowWorkItemFields property is set to 'false' then the work item fields will be displayed in the work item details grid.

ShowInboxViewOnly

If the ShowInboxViewOnly property is set to 'true', only the work item details grid will be displayed and the Left Tree menu view showing the workflow segregation will not be displayed.

If the ShowInboxViewOnly property is set to 'false', the work items details grid as well as the Left Tree menu view showing the workflow segregation will be displayed.

ShowManagerViewOnly

If the ShowManagerViewOnly property is set to 'true' then the Manager can see all the tasks assigned to the subordinates (peers) who report to him/her without the Left Tree menu view of workflow segregation display. The Manager can perform the following operations from the Manager View:

Reassign a work item to another user

Delegate a work item to another user

Set the priority for the work items assigned to his subordinates (peers)

UserIdString

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Provides the UserIdString of the Actor who will act on the activity.

Different UserIdString formats

- If the repository is mapped to AVEVA Work Tasks List user provider, the UserIdString will be skeltalist::[GUID]. Here, the GUID is taken from the SKEUsers table present in the Repository DB.

  For example: skeltalist:: C99B4110-C8CC-410B-BF70-93F3A7E34EDF
- If the repository is mapped to Active Directory user provider, the UserIdString will be activedirectory::[samAccountName]. Here, samAccountName is an active directory property value.

  For example: activedirectory::stephen
- If the repository is mapped to Forms user provider, the UserIdString will be sqlprovider::[UniqueIdentifier]. Here, the UniqueIdentifier is the unique identifier ID for the resource.

  For example: sqlprovider::6fb03011-68b7-437e-972e-21c67e4e2fe7
- If the repository is mapped to ASP.NET Membership user provider, the UserIdString will be membershipprovider::[UniqueIdentifier]. Here, the UniqueIdentifier is the unique identifier ID for the resource.

  For example: membershipprovider::a9bfc969-cc82-4c4b-99b9-50d9e1a91c01

EnableHold

If the EnableHold property is set to 'true' then in the Manage Work Item drop-down list, "Hold" will be displayed so that the Actor can put the work item on hold.

The Hold option will be available only if the activity is assigned to a single user. If the activity is assigned to multiple users, the Hold option will not be displayed in the Manage Work Item drop down.

ShowFilterBarAlways

This property can be set for displaying or hiding the filter bar. For example, if the ShowFilterBarAlways property is set to false, the filter bar will not be displayed.

If the ShowFilterBarAlways property is set to true, the filter bar will be displayed.

Application Name

Provides the Application or Repository Name.

Workflow Names

Based on the Workflow Names property specified, the Left Tree menu view shows the workflows.

EnableExport

Based on this property, you can control the visibility of the Export icon in the work item details grid status bar. If the EnableExport property is set to 'true', the "EnableExport" icon will be displayed.

If the EnableExport property is set to 'false', the "EnableExport" icon will not be displayed.

QueueDisplayOrder

The QueueDisplayOrder property is used to display the queues in a specific order in which you want them to appear in the Left Tree menu view.

Create a SortedDictionary<int,string> that specifies the Order and the QueueName. Assign the SortedDictionary to the QueueDisplayOrder Property of the Workitemcontrol.

For example, if the Queue Names are A,B,C and if you want to display them in "C,A,B" order, the following code

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

can be used:

```
Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext();
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl wi = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
wi.ID = "WorkItemControl"; // This is mandatory.
wi.ApplicationName = uContext.Repository.ApplicationName; //Provide Application or
Repository name. This is mandatory.
wi.UserIdString =uContext.LoggedInUserId ; //Provide user id string. This is mandatory.
System.Collections.Generic.SortedDictionary<int, string> Qdisplay = new
System.Collections.Generic.SortedDictionary<int, string>();
Qdisplay.Add(1,"C");
Qdisplay.Add(2,"A");
Qdisplay.Add(3,"B");
wi.QueueDisplayOrder = Qdisplay;//Assign the SortedDictionary to the QueueDisplayOrder
Property of the Workitemcontrol
Panel1.Controls.Add(wi);
```

ShowStandardColumns

If the ShowStandardColumns property is set to True then the standard columns will be displayed. If the ShowStandardColumns property is set to 'false', then the specific work item fields related to the workflow along with the Subject column will be displayed.

The columns (fields) will get displayed based on the Workflow Name property specified. If the property is not set, then the work item fields for all the workflows of the repository will be displayed.

GroupTotalColumns

The Group Total Columns property is used to calculate the total of the column(s) name when the grouping operation is performed on columns.

For example, consider a scenario where the 'Leave Days' and 'LeaveReason' are added as work item fields for the workflow 'Employee Leave Details'. To calculate the total number of leaves taken by the employee:

1. Set the GroupTotalColumns property for the columns for which the total has to be calculated when the grouping operation is performed on other columns.
2. The following code snippet shows how to use GroupTotalColumns property:

   ```
   Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext();
   Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl wi = new
   Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
   wi.ID = "WorkItemControl"; // This is mandatory.
   wi.ApplicationName = uContext.Repository.ApplicationName; //Provide Application or
   Repository name. This is mandatory.
   wi.UserIdString =uContext.LoggedInUserId; //Provide user id string. This is mandatory.
   //The bool parameter in the SortedDictionary is to specify whether the currency symbol
   should appear before the column values are not.
   System.Collections.Generic.SortedDictionary<string, bool> groupcolumns = new
   System.Collections.Generic.SortedDictionary<string, bool>();
   groupcolumns.Add("LeaveDays", false);
   wi.GroupTotalColumns = groupcolumns;
   Panel1.Controls.Add(wi);
   ```
3. Go to Inbox. The total number of leaves are displayed as shown below when the grouping is done on any column. In this case, it is the Activity Display Name column.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

GroupCountColumns

The **GroupCountColumns** property is used to calculate the count of the columns when the grouping operation is performed on columns.

For example: Consider a scenario where the 'Leave Days' and 'LeaveReason' are added as work item fields for the workflow 'Employee Leave Details'. To calculate the number of rows based on LeaveReason column:

1. Set the **GroupCountColumns** property for the column for which the count is to be calculated.
2. The following code snippet shows how to use the GroupCountColumns property.

```
Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext();
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl wi = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
wi.ID = "WorkItemControl"; // This is mandatory.
wi.ApplicationName = uContext.Repository.ApplicationName; //Provide Application or
Repository name. This is mandatory.
wi.UserIdString =uContext.LoggedInUserId; //Provide user id string. This is mandatory.
System.Collections.Generic.SortedDictionary<string, string> groupcntcolumns = new
System.Collections.Generic.SortedDictionary<string, string>();
groupcntcolumns.Add("LeaveReason", "LeaveReason");
wi.GroupCountColumns = groupcntcolumns;
Panel1.Controls.Add(wi);
```

3. Go to Inbox. The total count is displayed when we group the Activity Display Name.

GroupAverageColumns

The GroupAverageColumns property is used to calculate the average of the columns when the grouping operation performed on a column. The user can also display the currency symbol ($) for numeric fields.

This property is of type SortedDictionary<string,bool> where key<string> specifies the column name and value<bool> specifies whether the currency symbol needs to be displayed or not.

For example: Suppose that the 'Expense Amount' and 'Expense Type' work item fields are added to the 'Expense Approval' workflow and you want to calculate the average of the 'Expense Amount' column.

1. Set the GroupAverageColumns property with the column name for which the total is to be calculated.

```
Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext();
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl wi = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
wi.ID = "WorkItemControl"; // This is mandatory.
wi.ApplicationName = uContext.Repository.ApplicationName; //Provide Application or
Repository name. This is mandatory.
wi.UserIdString =uContext.LoggedInUserId; //Provide user id string. This is mandatory.
System.Collections.Generic.SortedDictionary<string, string> groupavgcolumns = new
System.Collections.Generic.SortedDictionary<string, string>();
groupavgcolumns.Add("Expense Amount", "Expense Amount");
wi.GroupAverageColumns = groupavgcolumns;
Panel1.Controls.Add(wi);
```

2. Go to **Inbox**. The average of **'Expense Amount'** will be displayed.

WorkflowDisplayOrder

The WorkflowDisplayOrder property is used to display the workflows in a specified order.

Create a SortedDictionary<int,string> variable, which specifies the Order and the Workflow Name. Assign the SortedDictionary to the WorkflowDisplayOrder Property of the Workitemcontrol.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For example, if the Workflow Names are Employee Leave Details, Expense Approval, and Purchase Order and if you want to display them as Expense Approval, Purchase Order, and Employee Leave Details, then use the following code.

```
Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext();
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl wi = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
wi.ID = "WorkItemControl"; // This is mandatory.
wi.ApplicationName = uContext.Repository.ApplicationName; //Provide Application or
Repository name. This is mandatory.
wi.UserIdString =uContext.LoggedInUserId; //Provide user id string. This is mandatory.
System.Collections.Generic.SortedDictionary<int, string> Qdisplay = new
System.Collections.Generic.SortedDictionary<int, string>();
wDisplayOrder.Add(3, "Employee Leave Details");
wDisplayOrder.Add(1, "Expense Approval");
wDisplayOrder.Add(2, "Purchase Order");
wi.WorkflowDisplayOrder = wDisplayOrder;
Panel1.Controls.Add(wi);
```

LayoutViewName

This property is used to load a specific view when we load a Work items list control.

ShowCounter

This property is used to show or hide the counters for the work items Inbox. When this property is set to False, you will not be able to see the number of items count under each section in the Left tree menu view in the work item Inbox.

```
ShowCounter = true;
```

If this property is set to False, you will not be able to see how many items there are in the work item Inbox.

Sample Code

```
Skelta.Entity.UserContext usrcn = new Skelta.Entity.UserContext("John", new
Skelta.Core.ApplicationObject("Telematics"), "activedirectory", "", "", false);
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl wi = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
wi.ID = "WorkItemControl";
wi.ShowCounter = false;
wi.ApplicationName = uContext.Repository.ApplicationName; //Provide Application or
Repository name. This is mandatory.
wi.UserIdString =uContext.LoggedInUserId; //Provide user id string. This is mandatory.
Panel1.Controls.Add(wi);
```

ShowDetailsOnRowDoubleClick

This property enables you to open the Work Item Detail view on double-click of the row in the Work Item details grid.

Sort on the subject column in Work Item Inbox has been enabled. Also, the sort order in Work Item Inbox will be retained after submitting a work item.

ShowQueuesWithWorkitems

Showing the queues with work items under the left hand side workflow grouping of the Work items list control is controlled by the Workitemlist Control Property - ShowQueuesWithWorkitems. Based on this property, you can control the behavior of showing only the queues with work items under the workflows even though you may be a participant of those queues.

ShowDetailViewWithMultiSelect

Showing the work items detailed view with the Multi Select option is controlled by the Workitemlist Control

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

property (ShowDetailViewWithMultiSelect). Based on this property, you can control the behavior of showing the Work Item Detail view with the Multi Select window enabled.
Showing the Work Item Detail view with the Multi Select option also depends on the ShowDetailsonRowDoubleClick property.

If the ShowDetailsonRowDoubleClick is set as 'true', the Work Item Detail view will open on double-clicking the row with the Multi Select option enabled.

If ShowDetailsonRowDoubleClick is set as 'false', then the Work Item Detail view will open on selecting the row with Multi Select option enabled.

You can select all the work items within a group and act on it with the help of Selectallwithingroup option available as context menu with the Multi Select mode.

ShowMultiSelectWindowInCenter

The Multi Select window displayed in the center of the screen is controlled by the Workitemlist Control Property - ShowMultiSelectWindowInCenter. Based on this property, you can control the behavior of showing the Multi Select window.

QueueDetailViewWithoutPicking

Showing the Queue Detail view without picking the item from the Queue is controlled by the Work items list control property - QueueDetailViewWithoutPicking. Based on this property, you can enable or disable this feature in the Work items list control. By default, this feature is disabled.

Viewing the Queue Detail View page is possible in two ways.

- Double-click the item from the grid to get the Detail view.
- Through the context menu

The Queue Detail View page can be seen from the context menu option available from the Queue Work Item. Select Form Instances > Main Form to view the work items in detail.

EnableMoveNextPreviousWorkitem

This property enables you to navigate between different work items from the Work Item Detail view. This prevents the need to go back to the main details grid which shows the list of work items. You can also sort the Work Item Detail grid in some particular order and then navigate through the work items in that order.

EnableAutoMoveNextWorkitem

This feature enables you to view the Work Item Details for the next work item after submitting a work item. Moving to the next work item on submission of the work item from the detailed view is controlled by the WorkItemlist control property - EnableAutoMoveNextWorkitem.

A check box Show next work item on submission is available on the Work Item Detail view.

If selected, details of the next work item are shown after submission of the current work item.

If de-selected, the Work Item Detail view window will get closed without navigating to the next work item.

Split Bar in the Work Item List control

Split bar has been implemented in the Work Item List control. It can be used to adjust the width of the Work Item List control, if the workflow name or the Queue name is very long.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Use Events of Work Item List Control

Code snippets for using events of Work Item List Control are explained below

Sample code snippets for adding a work item control and adding events. There are several delegate events available for work item control. They are InitializeDataSourceEvent, InitializeLayoutEvent, InitializeWorkItemRowEvent, ColumnsAlignmentEvent, ColumnstoMakeInVisible events.

The following code snippet shows how to add a work item list control using the events.

```
protected override void OnInit(EventArgs e)
{
Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext();
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl WIL = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
WIL.ApplicationName = uContext.Repository.ApplicationName;
WIL.UserIdString = uContext.LoggedInUserId;
WIL.ID = "WorkItemListControl";
WIL.DisplayGroupUI = true;
WIL.Height = Unit.Percentage(100);//680;
WIL.ReadingPaneInPopupWindow = true;
WIL.EnableHold = true;
WIL.ShowInboxViewOnly = false;
WIL.ShowManagerViewOnly = false;
WIL.DisplayLeftTreeMenuAtApplicationLevel = false;
WIL.ShowStandardColumns = true;
WIL.ShowDetailsonRowDoubleClick = true;
WIL.ShowWorkItemFields = true;
WIL.ColumnstoMakeInVisibleEvent += new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl.ColumnstoMakeInVisible(W
IL_ColumnstoMakeInVisibleEvent);
WIL.InitializeDataSourceEvent += new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl.InitializeDataSource(WIL
_InitializeDataSourceEvent);
WIL.InitializeWorkItemRowEvent += new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl.InitializeWorkItemRow(WI
L_InitializeWorkItemRowEvent);
WIL.InitializeLayoutEvent += new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl.InitializeLayout(WIL_Ini
tializeLayoutEvent);
WIL.ColumnsAlignmentEvent += new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl.ColumnsAlignment(WIL_Col
umnsAlignmentEvent);
PanelWIL.Controls.Add(WIL);
}
```

Sample code snippet for adding a new column to the dataset which gets passed from the event used for showing the work item details grid.

```
/// <summary>
/// To add a new column to the dataset which gets passed from the event
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
void WIL_InitializeDataSourceEvent(object sender,
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorklistDataSourceEventArgs e)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
DataColumn checkdatacolumn = new DataColumn();
checkdatacolumn.ColumnName = "New Column";

if (!e.workItemListDataSet.Tables[0].Columns.Contains(checkdatacolumn.ColumnName))
e.workItemListDataSet.Tables[0].Columns.Add(checkdatacolumn);
}
```

Sample code snippet for adding a new column to view it in the work item details grid.

```
/// <summary>
/// To add the new column to the see in the grid
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
void WIL_InitializeLayoutEvent(object sender,
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorklistInitialzeLayoutEventArgs e)
{
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkitemListColumns workitemListcolumn =
new Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkitemListColumns();
workitemListcolumn.Name = "New Column";
workitemListcolumn.Caption = "New Column";
workitemListcolumn.Bound = false;
workitemListcolumn.ColumnType = "text";
workitemListcolumn.Width = 50;
workitemListcolumn.AllowGrouping = false;
workitemListcolumn.AllowSorting = true;
workitemListcolumn.AllowSizing = false;
workitemListcolumn.WorkitemListColumnCollection.Add(workitemListcolumn.Name,
workitemListcolumn);
e.WorkItemColumnsList = workitemListcolumn.WorkitemListColumnCollection;
}
```

Sample code snippet for setting values for the added column for the row in the work item details grid.

```
/// <summary>
/// To set the values for the added column for the row in the grid
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
void WIL_InitializeWorkItemRowEvent(object sender,
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorklistDataRowEventArgs e)
{
  DataRowView dr = e.WorkItemRowView;
string value = dr.Row["ExecutionId"].ToString();
e.WorkItemColumnTextandValue.Add("New Column", value);
}
```

Sample code snippet for aligning the columns and setting the formatstring. We have to specify the exact ColumnName and not the caption of the ColumnName.

```
/// <summary>
/// Event to use to align the columns and setting formatstring
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
void WIL_ColumnsAlignmentEvent(object sender,
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorklistAlignmentArgs e)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkitemListAlignmentColumns
wilAlignColumns1 = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkitemListAlignmentColumns();
 wilAlignColumns1.ColumnName = "ActivityDisplayName";
 wilAlignColumns1.ColumnAlignment = "right";

Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkitemListAlignmentColumns
wilAlignColumns2 = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkitemListAlignmentColumns();
 wilAlignColumns2.ColumnName = "CreatedDateTime";
wilAlignColumns2.DataFormatString = "dd.MM.yyyy";



e.WorkitemAlignmentColumnList.Add(wilAlignColumns1.ColumnName, wilAlignColumns1);
e.WorkitemAlignmentColumnList.Add(wilAlignColumns2.ColumnName, wilAlignColumns2);

}
```

Options available for ColumnAlignment are: Left, Right, and Center.

Options available for DataFormatString are:

For Date Columns, you can define this format:

1. "dd.MM.yyyy" - It will display the date like 04.08.2008
2. "d" - It will display as short date like 08/26/2008
3. "D" - It will display as long date like Tuesday, August 26, 2008

For Numeric Columns you can define this format:

1. "N" - Displays as general format.
2. "F" - Displays as Fixed format.
3. "c" - Displays as Currency Format.

Sample code snippet for hiding the columns in the Work Item Details grid.

```
/// <summary>
/// To hide the columns
/// </summary>
/// <returns></returns>

string[] WIL_ColumnstoMakeInVisibleEvent()
{
string[] columns = new string[2] { "ResponseBy","CreatedDateTime"};
return columns;
}
```

Layout Customization

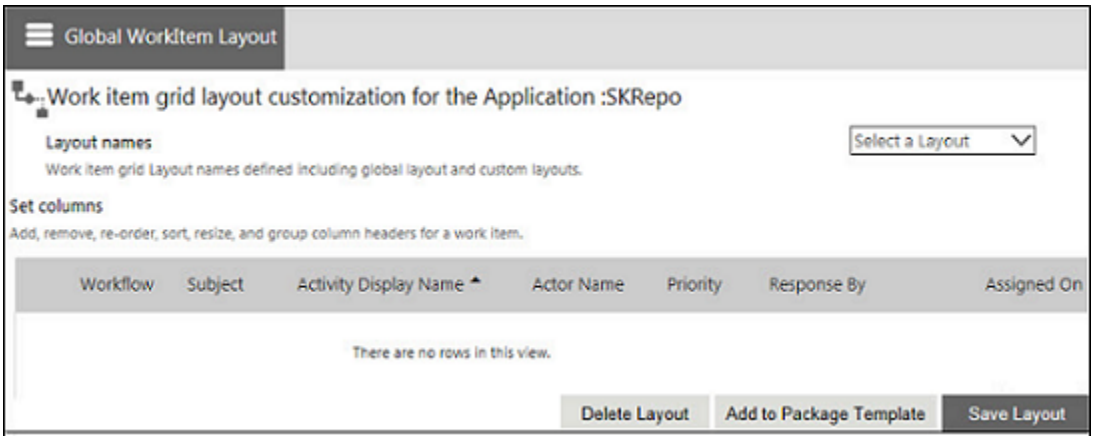
You can define have a specific view for a repository so that the defined view appears when you load the work item list control.

You can save a specific layout or global layout by selecting **Settings > Global Work Item Layout** from the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Enterprise Console pull-down menu.

On clicking **Global Work Item Layout**, the following page will be displayed:

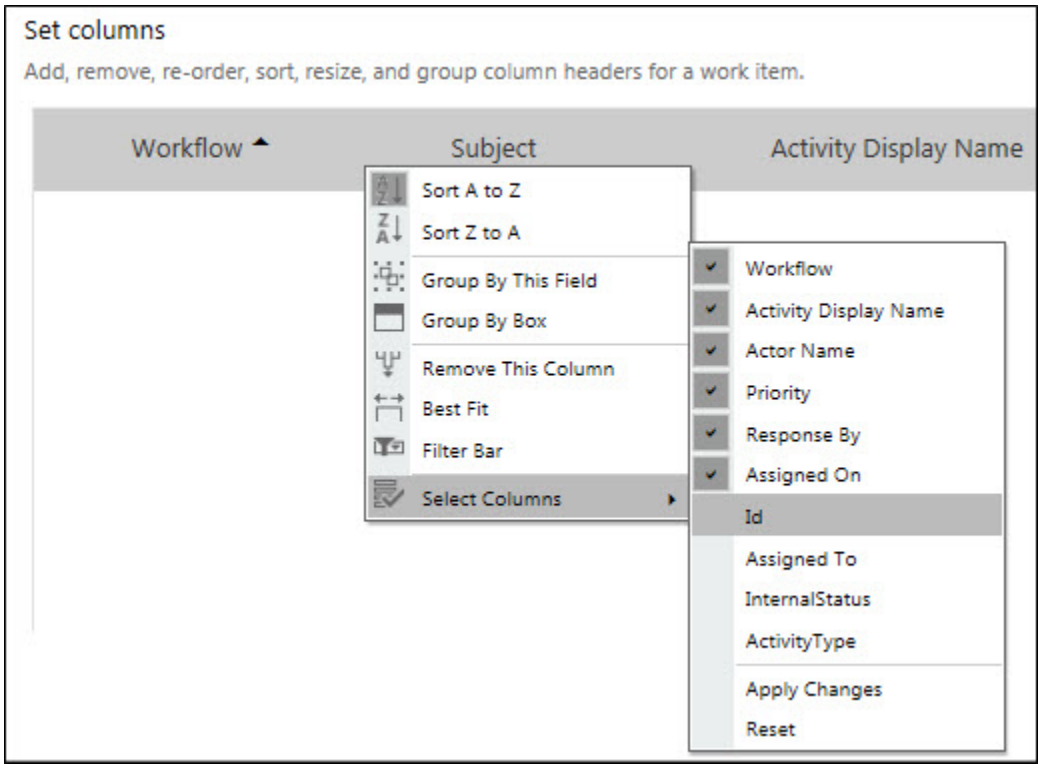


There are two ways to set layouts:

1. GlobalLayout
2. CustomLayout

To save Layout as Global Layout

1. Select **Settings > GlobalWorkItem Layout** from the Enterprise Console pull-down menu.
2. In the global layout page, select **GlobalLayout** from the **Layout names** drop-down.
3. Right-click any column fields and select the required column fields, as shown below



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Select **Apply Changes** and then **Save Layout**.
5. Go to Inbox. The layout will be displayed with specified column fields defined.

There may be a scenario where we want to have different layout specific to different processes.

To save Layout as Specific Layout

1. Select **Settings > GlobalWorkItem Layout** from the **Enterprise Console** menu.
2. In the global layout page, select **CustomLayout** from the **Layout names** drop-down.
3. Click **New Custom layout**.
4. Give a layout name in the **New Custom layout name** text box. For example, 'Leave Details View'.
5. Right-click any column and select the required column fields.
6. Select **Apply Changes** and then **Save Layout**.
7. The following code snippet shows you how to specify a custom layout for the work item control. Set the **LayoutViewName** property with the custom layout name. For example, 'Leave Details View'.

```
Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext();
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl wi =  new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
wi.ApplicationName = uContext.Repository.ApplicationName; //Provide Application or
Repository name. This is mandatory.
wi.UserIdString =uContext.LoggedInUserId; //Provide user id string. This is mandatory.
System.Collections.Generic.SortedDictionary<int, string> Qdisplay = new
System.Collections.Generic.SortedDictionary<int, string>();
wi.LayoutViewName = "Leave Details View";
//LayoutViewName property is used to load the specified layout.
```

Note: If the LayoutViewName has not been specified, then the Global Layout created for the repository will be considered as the default layout for the Work Item list control.

8. Go to the **Work items list** Inbox. The layout will be displayed with specified column fields.

Note: If you save the customized layout with the **Actor Name** column, this column will be displayed in the Work Item List control only for the Queue view. However, this column will not be displayed for the Inbox and Manager views.

Create Custom Link in Work Item List Control

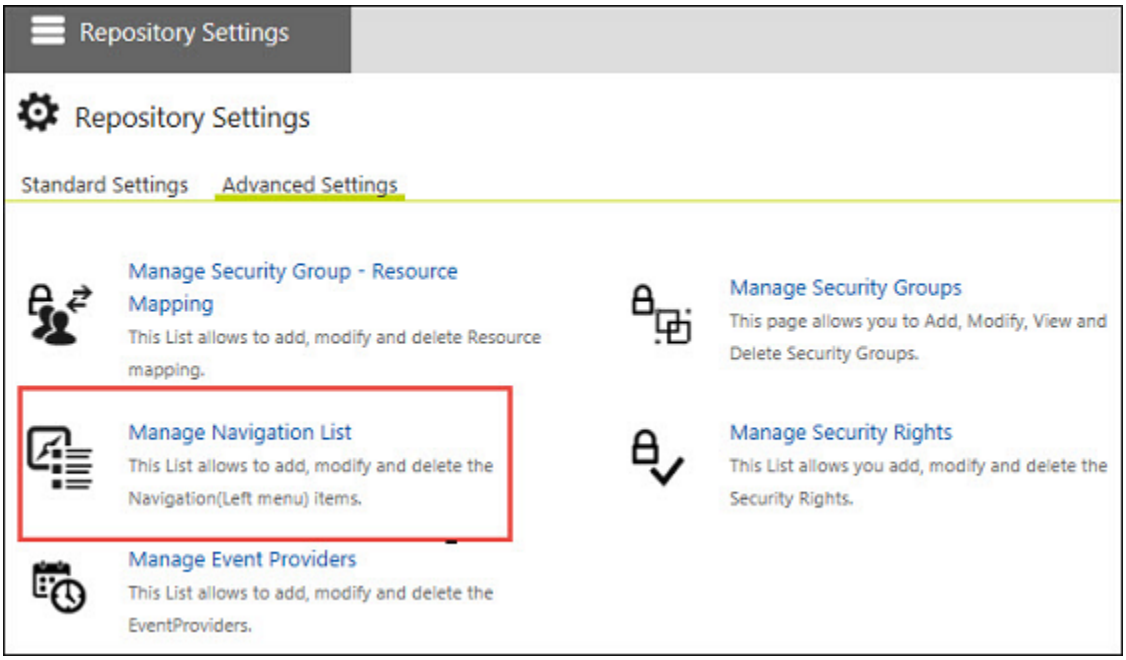
This feature enables you to create custom links in the left hand side tree view menu of the Work items list control.

Scenario

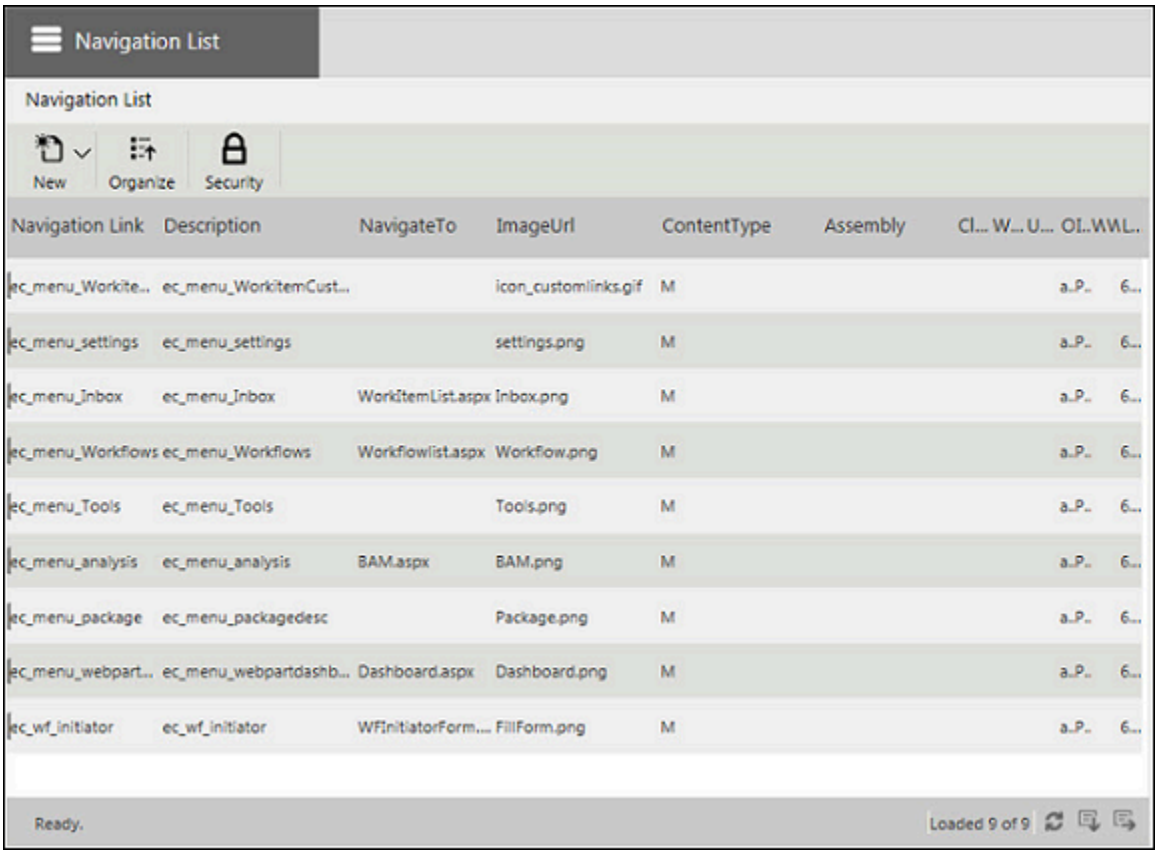
John is the manager in Telematics organization acting on the HR and Accounts Payable Process. The manager wants to view a custom page like the Workflows list page from the work items Inbox on click of a link from the LHS tree menu view of Work items list control. Also, wants this custom link to be secure so that only the users with proper rights should be able to view it and also wants sub links under the main custom link with different navigations.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

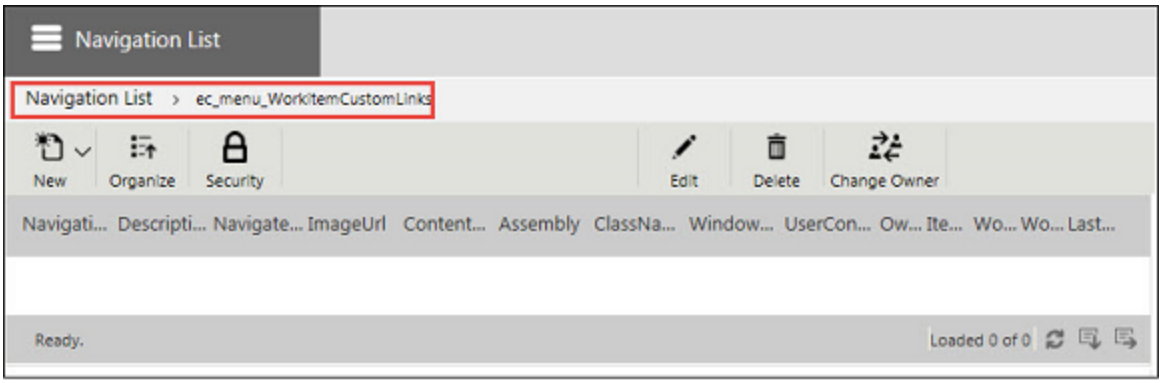
You can create custom links through the **Navigation List of Repository Settings**.



Whenever a new repository is created, an entry - **ec_menu_WorkitemCustomLinks** is created in the Navigation List. You can create sub links under this Navigation link or can edit the link for adding a page if no sub link is needed.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Calendar Control

AVEVA Work Tasks Calendar control allows you to define and configure multi-level organizational calendars. The Calendar interface enables creation of hierarchical organization and resource calendars. The Calendars are used to determine availability of resources, calculate time slots, and perform other scheduling functions.

To add Calendar control using a panel

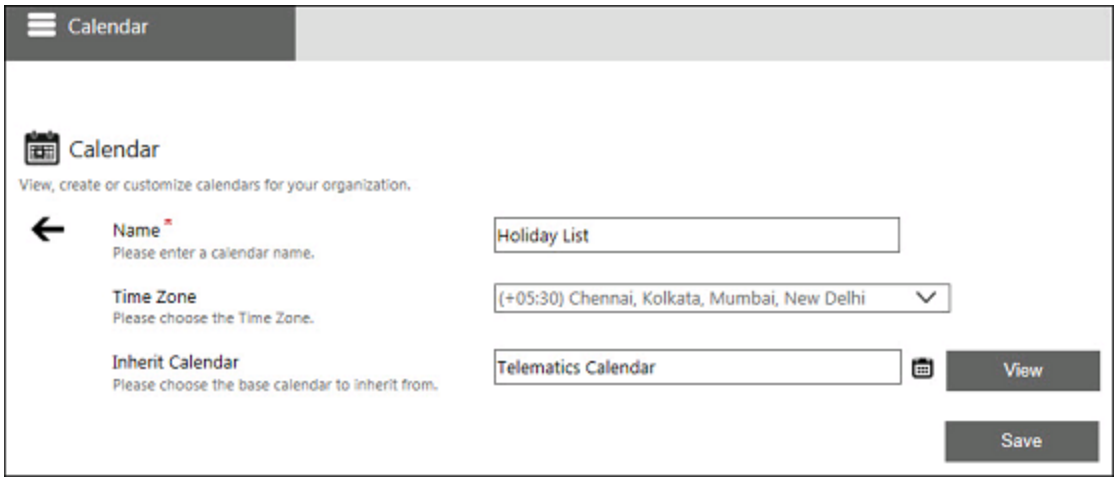
1. Before adding the Calendar control in a panel, create an ASP.NET project in Visual Studio .NET 2012.
2. Open **Calendar.aspx.cs** page.
3. Make sure that you have created the User Context object as per the steps given in Create User Context
4. In the underlying code, create a new instance of control, and add the instance to the panel in the Page_Load as shown in the following example.

```
Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject("ListRepo");//Repository name
object loggedinUserId = "4952ffff-e0db-401b-bb7d-70a0fec1df07";
Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext(loggedinUserId,
application, "skeltalist");
Skelta.HWS.Web.CalendarControl.CalendarControl cc = new
Skelta.HWS.Web.CalendarControl.CalendarControl();
cc.ApplicationName = userContext.Repository.ApplicationName; //Provide Application or
Repository name. This is mandatory.
cc.ID = "Calendar"; // This is mandatory.
Panel1.Controls.Add(cc);
```

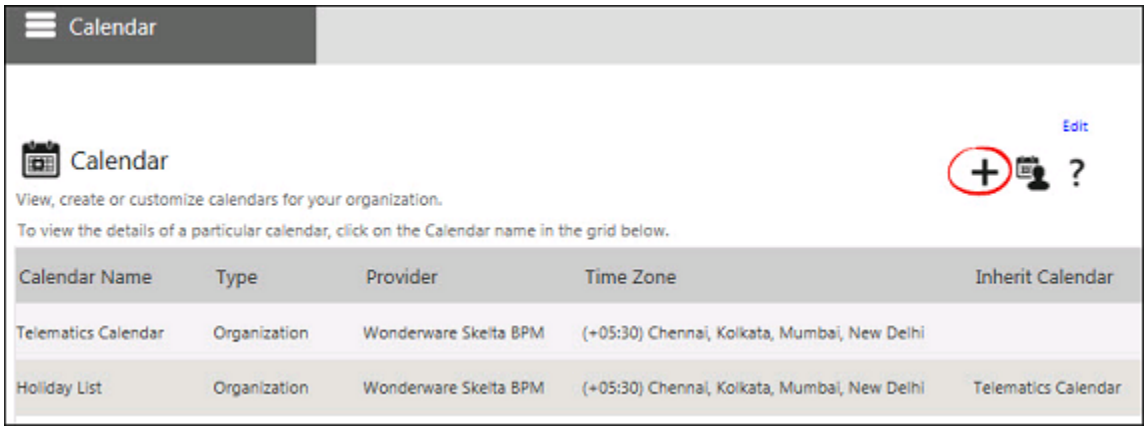
Properties that can be set for Calendar control

- **ApplicationName**: Provide the name of the application in which you want to use the calendar.
- **CalendarName**: Set the name of the calendar.
- **UserIdString**: Calendar can be created for a specific actor by providing UserIdString of that actor. Click Create User Context to view the different UserIdString formats.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



- **Can Add calendar**: If **CanAddCalendar** property is set to false, the "**Add New Calendar**" button will not be displayed and you will not have the rights to add calendar. If the **Can Add Calendar** property is set to true then the user can also Add/Delete Calendar Workslot.



- Can Add calendar workslot
- Can Delete calendar workslot
- **Can Delete calendar**: If CanDeleteCalendar property is set to false, the "Delete" button will not be displayed and you will not get rights to delete calendar.
- **Can List calendar**: If CanListCalendar property is set to false, the "List" button will not be displayed and you cannot get the list of calendars.
- **Can View calendar**: If the CanViewCalendar property is set to false, the "View" button will not be displayed and you will not have rights to view the inherited calendar.
- SecurityUserIdString

Properties that are not used in Calendar control

- CanAddCalendarWorkSlot
- CanDeleteCalendarWorkSlot

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

BAM Control

The Business Activity Monitoring (BAM) control provides features that can be used by a process administrator to monitor and control various instances of the workflow(s) in a repository. It provides a graphical representation of the various instances of the workflows using three different views.

To add BAM control using a panel

1. Before adding the Process Designer control in a panel, Create an ASP.NET Project in Visual Studio .NET.
2. Go to **BAM.aspx.cs** page.
3. Ensure that you have initialized the User Context object as per the steps given in Create User Context.
4. In the underlying code, create a new instance of control and add the instance to the panel in the Page_Load, as in the example below.

```
Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject("ListRepo");//Repository name
object loggedinUserId = "4952ffff-e0db-401b-bb7d-70a0fec1df07";
Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext(loggedinUserId,
application, "skeltalist");
Skelta.BAM.UserControls.BAMControl bam = new Skelta.BAM.UserControls.BAMControl();
bam.ApplicationName = userContext.Repository.ApplicationName; //Provide Application
or Repository name. This is mandatory.
bam.ID = "BAM";
bam.UserIdString = userContext.LoggedInUserId; //Provide user id string. This is
mandatory.
bam.Height = unit.Percentage(100);
bam.Width = Unit.Percentage(100);
Panel1.Controls.Add(bam);
```

5. If the bin folder does not exist in the user project (D:\Develop\SkeltaTestSite), create the bin folder under user project (D:\Develop\SkeltaTestSite).
6. Copy **ComponentArt.Charting.WebChart.dll** and **ComponentArt.Charting.lic** from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\EnterpriseConsole\bin* to bin(*D:\Develop\SkeltaTestSite\bin*) folder of the user project.
7. Create "TempReports" folder under the project.
   This step is required because temporary data (Example like Output files generated at the time of exporting the report) of BAM reports is stored in the "TempReports" folder.
8. Create BAM Reports folder in path *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks*.
9. Provide complete "Access rights" to NETWORK SERVICE account to the folders specified below in Windows Server 2003 machine. In the case of Windows XP, provide full "Access rights" to ASP.NET Machine account to the same folders specified below:
   a. TempReports: Which is under project
   b. Image folder: Exists in the path *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\BPMUITemplates\Default\BAM folder*
   c. BAMReports: Exists in the path *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks*.

Properties that can be set for BAM control

- ApplicationName: Provide application name or repository name to get information about workflow status.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- UserIdString: Provide UserIdString of the actor to get the status of workflow where the actor is part of workflow. Click Create User Context to view the different UserIdString formats.
- WorkflowNames: Provides filtering so that the status of only the workflows specified in the WorkflowNames property only be displayed.
- ShowCategories: Shows only those categories as mentioned in the ShowCategories property, i.e. the categories as shown in the left pane of the BAM Reports (Dashboard, Analyze).

Note: The BAM Control properties ShowNodes, HideNodes, and ShowTree are obsolete and should not be used with AVEVA Work Tasks.

Add Individual BAM Report to aspx Page

In some scenarios there may be a requirement to render an individual report in an aspx page, and also to control some of the characteristics of the report just by specifying the properties of the BAM control.

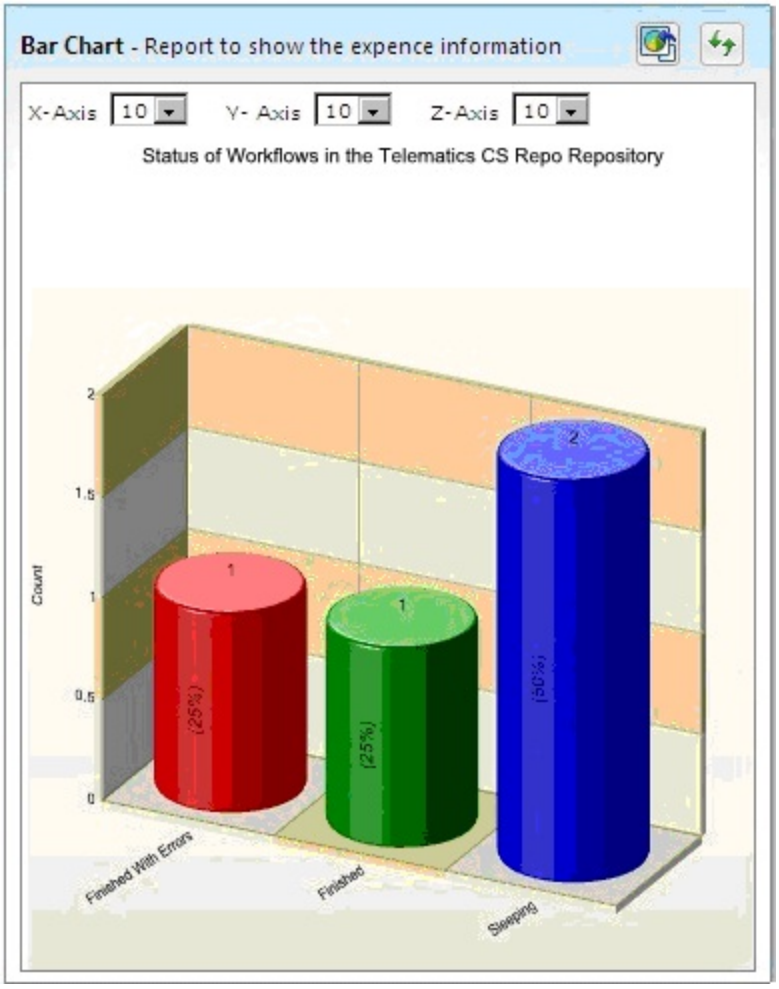
Features

- Support for rendering an individual report in an aspx page.
- Characteristics of the report can be controlled just by specifying the report properties.
- Security enabled report rendering.
  To enable rendering of individual report in an aspx page, BAM report has been enhanced with some more properties to it. Below are the properties that can be used to render an individual report in an aspx page.
  - **IsAnIndividualReport**: If this property is set as true, it enables the individual report rendering. Else default configurations will be considered for rendering the report.
  - **ReportName**: This is the hierarchy of the report in the BAM Control. For example, to render all the executing instances in an aspx page, ReportName property can be set with 'Analyze/Execution/Executing'.
  - **ReportID**: This is a unique Id for any report. This information will be taken from the database where the report information is stored.
    'ID' column of 'SKBAMMenu' table, this information will be unique.

    Note: If both the properties 'ReportID' and 'ReportName' are set then, 'ReportName' will be considered for rendering the report.
  - **Width**: Width of the control can be set using this property.
  - **Height**: Height of the control can be set using this property.
  - **Web Part Border**: If this property is set as true, it will render the individual reports with border else it will render them without border.

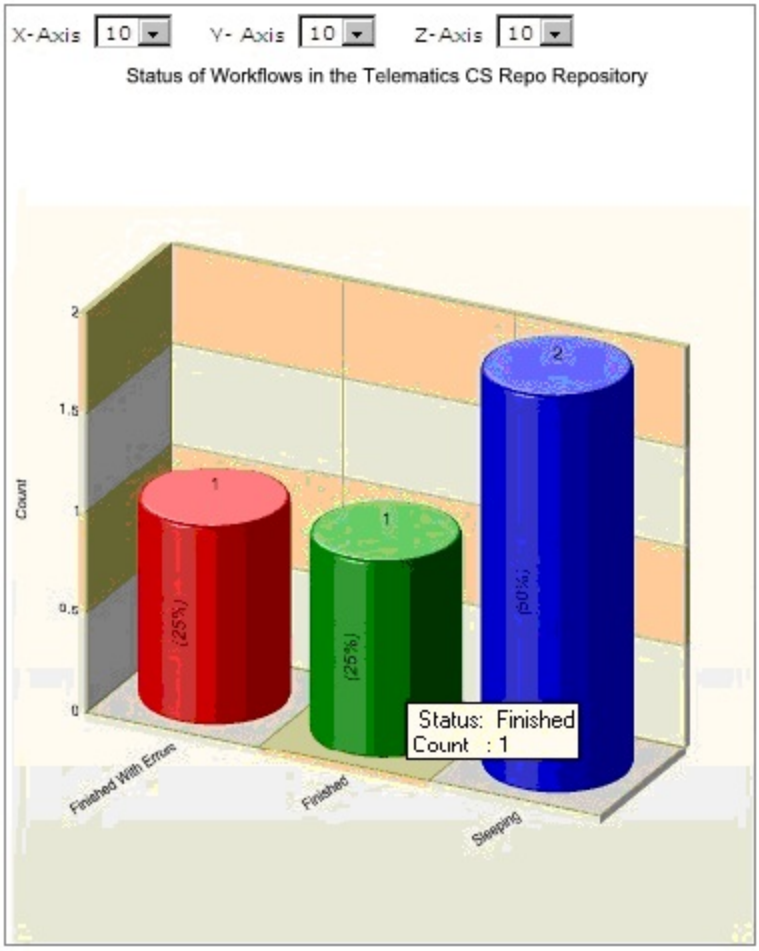
With Border:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Without Border:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



- ReportTitle: Title of the report which will be displayed on the top left corner of the report.
- ReportDescription: Description for the report which will be displayed next to the report title.

Scenario
'XYZ' organization is one of the leading supply chain management companies. 'XYZ' has a web site in which home page needs to be displayed with one of the customized BAM reports.
The above scenario can be achieved by using Business Activity Monitoring control. By specifying the properties of the BAM control, it is possible to render an individual report in an aspx page.

Design of the above scenario involves the following steps:
Step 1: Drag and drop the BAM control to any aspx page
Step 2: Set the BAM properties
The following code sample shows how to set the properties of the BAM control to render the report individually.
Sample ASPX page:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="IndividualBam.aspx.cs"
Inherits="IndividualBam" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
    <title></title>
    <style type="text/css">
    html,body,form
    {
    height:100%;
    }
    </style>
  </head>
  <body>
    <form id="form1" runat="server">
    <div style="height:100%;">
    <asp:Panel ID="Panel1" runat="server" Height="100%" Width="100%"></asp:Panel>
    </div>
    </form>
  </body>
  </html>
```

Sample Code file:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
public partial class IndividualBam : System.Web.UI.Page
{
protected void Page_Load(object sender, EventArgs e)
{
//Instance of BAM Control
Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject("Telematics Repository");//Repository name
object loggedinUserId = "81734457-B3DB-4BC9-894A-C3149828C65C";
Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext(loggedinUserId,
application, "skeltalist");
Skelta.BAM.UserControls.BAMControl bamControl = new
Skelta.BAM.UserControls.BAMControl();
//Application Name
bamControl.ApplicationName = "Telematics Repository";
//UserID
bamControl.UserIdString = userContext.LoggedInUserId;
//BAM ControlId
bamControl.ID = "myBAM1";
//ReportName
bamControl.ReportName = "Workflow Instances";
//ReportId [If ReportName is set, this property will be ignored.]
bamControl.ReportId = "DF2878B5-A97B-478F-BD14-89E591D46BB9";
//To Show as an Individual Report
bamControl.IsAnIndividualReport = true;
//Report Title
bamControl.ReportTitle = "Bar Chart";
//Report Description
bamControl.ReportDescription = "Report to show the information";
//Height
bamControl.Height = Unit.Percentage(100);
//Width
bamControl.Width = Unit.Percentage(100);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//Add the Control to Panel
Panel1.Controls.Add(bamControl);
}
}
```

View Specific Workflow Instances in BAM Control

BAM is used for viewing all the workflow instances in different categories like Running Workflows, Failed Workflows, Aborted Workflows, Paused Workflows and Completed Workflows. This feature allows you to filter and show only specific workflow instances in the BAM control.

Showing the specific instances of the workflow in BAM is controlled by the property – **InstancesToBeShown**. This property is an ArrayList property to which you can add the Executionids of the workflow instances for which you want to see the details.

The property InstancesToBeShown is set say for the Execution ids 25, 24, 23, 22, 21. The report will be shown only for these workflow instances.

Queue Control

The Queue control is used to group resources as a common pool and assign tasks to the group.

To add Queue control using a panel:

1. Before adding the Process Designer control in a panel, Create an ASP.NET Project in Visual Studio .NET.
2. Go to **Queue.aspx.cs** page.
3. Make sure that you have initialized the User Context object as per the steps given in Create User Context.
4. In the underlying code, create a new instance of control and add the instance to the panel in the Page_Load as in the example below:

```
Skelta.Core.ApplicationObject application = new Skelta.Core.ApplicationObject
("Telematics Repository");//Repository name
object loggedinUserId = "81734457-B3DB-4BC9-894A-C3149828C65C";
Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext
(loggedinUserId, application, "skeltalist");
Skelta.HWS.Web.QueueControl.QueueControl que = new
Skelta.HWS.Web.QueueControl.QueueControl();
que.ID = "Q1"; //This is mandatory.
que.ApplicationName = userContext.Repository.ApplicationName; //Provide Application
or Repository name. This is mandatory.
que.UserIdString = userContext.LoggedInUserId; //Provide user id string. This is
mandatory.
Panel1.Controls.Add (que);
```

Different Properties that can be set for Queue control are:

- **ApplicationName** - Provide the application or repository name where Queue has to be created.
- **UserIdString** - Provide the actor UserIdString who wants to create or view the queue. By default, the user who creates the Queue will be the owner of the queue. Click Create User Context to view the different UserIdString formats.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

List Control

List Controls lists out the Workflows, Navigation Details, Users, Event Providers and Event Associations.

To add List control using a panel:

1. Before adding the Process Designer control in a panel, Create an ASP.NET Project in Visual Studio .NET.
2. Go to List.aspx.cs page.
3. Make sure that you have initialized the User Context object as per the steps given in Create User Context
4. In the underlying code, create a new instance of control, and add the instance to the panel in the Page_Load as shown in the following example:

```
Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject("Telematics Repository");//Repository name
object loggedinUserId = "81734457-B3DB-4BC9-894A-C3149828C65C";
Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext(loggedinUserId,
application, "skeltalist");
Skelta.Repository.Web.ListControl listControl = new
Skelta.Repository.Web.ListControl();
listControl.RepositoryName = userContext.Repository.ApplicationName;
listControl.ListName = "Users";
listControl.ID = "listControl1";
listControl.LoggedInUserId = userContext.LoggedInUserId;
listControl.FilterColumn = "";
listControl.FilterValue = "";
listControl.Height = Unit.Percentage(100);
Panel1.Controls.Add(listControl);
```

Different Properties that can be set for List Control:

- **Repository Name:** Provide the repository name created by the user.
- **Filter Column:** Provide column name for which user needs to apply filter. Example: Filter column = Title
- **Filter Value:** Provide the value for filtering. Example: If Filter value=test then "test" is searched based on "List Name" provided. Suppose, if the List Name is 'Workflow' then the workflow with name 'test' will be searched.
- **Filter Query:** Use the FilterQuery property if you want to filter with more than one condition.

  Sample code:

```
Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext();
Skelta.Repository.Web.ListControl listControl = new Skelta.Repository.Web.ListControl
();
listControl.RepositoryName = userContext.Repository.ApplicationName;
listControl.ListName = "Workflow";
listControl.ID = "listControl1";
listControl.LoggedInUserId = userContext.LoggedInUserId;
ListControl1.FilterColumns = "ItemType";
ListControl1.FilterQuery = "ItemType = 2 or ItemType = 3";
Panel1.Controls.Add(listControl);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

In this example, the ItemType refers to the filter query conditions to be specified in order to categorize the list of workflows.

- **List Name:** Provide the List Name.
  - **Example1**: If List Name= Workflow, then the workflows are displayed depending on List Id or List Item.
  - **Example 2**: If List Name= NavigationDetails, then information regarding the navigation of control is displayed.
  - **Example 3**: If List Name=Users, then all the "list provider" users are displayed.
  - **Example 4**: If List Name= Event Providers, then different providers for association are displayed.
  - **Example 5** :If List Name = Event Associations, then all the workflows for which association is created is displayed.
- **LoggedInUserId**: Provide the user ID of the user. Click here to view the different UserIdString formats.

Form List Webpart Control

Forms List Webpart Control will list out the forms available.

To add FormsList WebPart control using a panel

1. Before adding the Process Designer control in a panel, Create an ASP.NET Project in Visual Studio .NET.
2. Go to FormsWebPart.aspx.cs page.
3. Ensure to initialize the User Context based on r the steps given in Create User Context
4. Add the following style within the <head> tag.

```
<style type="text/css">
html, body, form {
height: 100%;
margin:0;
padding:0;
width:100%;
box-sizing:border-box;
}
</style>
```

5. Modify as follows:

```
<div style="width:100%;height:98%;box-sizing:border-box;">
```

6. In the underlying code, create a new instance of control and add the instance to the panel in the **Page_Load** as in the example below.

```
Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject("ListRepo");//Repository name
object loggedinUserId = "89b95a29-4931-44f1-8f4d-3e78036a936f";
Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext(loggedinUserId,
application, "skeltalist");
Skelta.WebParts.FormList frmWebPart = new Skelta.WebParts.FormList();
frmWebPart.ID = "formListWebpart";
frmWebPart.ApplicationName = userContext.Repository.ApplicationName;
frmWebPart.LoginName = userContext.LoggedInUserId;
frmWebPart.Height = new Unit(100, UnitType.Percentage);
frmWebPart.Width = new Unit(99, UnitType.Percentage);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
frmWebPart.InlineFormHeight = new Unit(90, UnitType.Percentage);
Panel1.Controls.Add(frmWebPart);
```

Different Properties that can be set for FormListWebPart Control

- **ApplicationName**: Provide the ApplicationName or Repository Name for the workflow report to be viewed.
- **LoginName**: Provide the User ID of the user who has logged in. Click here to view the different UserIdString formats.

Workflow Execution View Control

The WorkflowExecutingReport Control shows the Process Execution View of the currently executing workflow.

To add workflow executing report control using a panel

1. Before adding the Process Designer control in a panel, Create an ASP.NET Project in Visual Studio .NET.
2. Go to **ExecuteReport.aspx.cs** page.
3. Make sure that you have initialized the User Context object as per the steps given in Create User Context.
4. In the underlying code, create a new instance of control and add the instance to the panel in the Page_Load, as in the example below.

```
Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject("Telematics Repository");//Repository name
object loggedinUserId = "81734457-B3DB-4BC9-894A-C3149828C65C";
Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext(loggedinUserId,
application, "skeltalist");
Workflow.NET.Web.Report.WorkflowExecutionReport wf = new
Workflow.NET.Web.Report.WorkflowExecutionReport();
wf.Application = userContext.Repository.ApplicationName;
wf.ExecutionId = 1;
wf.Height = Unit.Percentage(100);
Panel1.Controls.Add(wf);
```

Different Properties that can be set for WorkflowExecutingReport Control are:

- Application:Provide the ApplicationName or Repository Name for the workflow report to be viewed.
- ExecutionId:Provide the ExecutionId (from the table SWExecute) of the workflow for which report has to be viewed.
- OnlyAnalystsView: If the OnlyAnalystsView property is set to true, then only the AnalystsView will be displayed (Progress View and Execution View will not be displayed).
- CanControlWorkflow: If the CanControlWorkflow property is set to true, then you can **Abort Executing Workflow** and **Pause Executing Workflow**.

Different views in the Process Execution Window

- The Analyst's View is a simplified view of the process design intended for end-users. You can save the simplified view of the process as a file and associate this filename in the Start Activity to enable this view for the end users.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The Progress View shows the workflow progress on the Process Diagram, up to the point that the Execution View is selected. It consists of the execution view mapped on to the Process Diagram. You can click a completed activity to view the Step Details window. You can view the step details, variables, contents, content (raw content passed to the workflow) using the corresponding tabs. The links that have already been traversed are represented by thick lines.
- The Execution View shows the workflow execution status up to the point the execution has happened.

Configure New Repository

This section explains you how to configure a new repository in AVEVA Work Tasks.

Set up Source Elements Directory

AVEVA Work Tasks allows extensive customization of its User Interface and of several other functional aspects. Most of this configuration information is stored in the *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**WorkflowElements**Default* directory.

Each application or workflow can point to a specific WorkflowElements Directory, so that the necessary customization of the entire user interface or functionality can be done at the application or workflow level.

The **AVEVA Work Tasks Templates** directory contains a number of XML files, which define the Actions used by AVEVA Work Tasks, as well as the behavior of Controls such as BAM, Work Item List. By modifying these files, you can extend or customize AVEVA Work Tasks.

The **Source Elements Directory** is configured as a part of **AVEVA Work Tasks Repository** settings.

Set up Email Addresses for Repositories

Email addresses to be used to send emails from AVEVA Work Tasks are specified in the Email Settings field in Repository Management and in the Farm Configuration Wizard.

Email settings specified at the repository level, farm and activity levels are used as follows:

- Activities using mail channel: Emails are sent from the email address specified for the repository.
- Send email activity: The email address specified in the action property is used.
- Exception Email: All exceptions logged in the console will be emailed to the mail address configured in Farm Configuration Global Settings in the "From Email" and "To Email" fields.
- Exception emails during activity execution are sent to the address configured in the "From Email" and "To Email" fields of the repository. Exception email will be sent to the Submitted by user if the workflow initiation fails in the engine.
- Repository "From mail": It is used to send activity mail to the actor.
- Action Property "From Email Address": It is used for sending information (like adding the task) to the actor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Set up User Providers for Repositories

AVEVA Work Tasks uses the concept of User Providers for connecting to users participating in the workflow process. This allows it to use a wide variety of User Data Sources, such as Active Directory or Native-Application-specific Application Database. AVEVA Work Tasks provides a default user database which is available in the User Provider options for a repository as AVEVA Work Tasks List.

Refer to *Administrator Guide* for configuring user providers.

Set up Document Web View Control for Repositories

AVEVA Work Tasks uses ASP.NET Server Controls to render the workflow data to various participants in the workflow. This data is displayed to the participants when they are performing actions such as Approvals or working on tasks, and is entirely application specific. A class implementing the INativeWebFormUIControl, ISkeltaAddInProvider interface can be configured to display the information specific to the application, in the activity list.

To configure the default document view handler of AVEVA Work Tasks

1. Create a new html file which should be the new default document view and put that file in the *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks*[*WorkflowElements**Default**en-US**Activities**Activity**Custom* folder with any name.
2. Create a class library project and add reference to the following dlls in the AVEVA Work Tasks Bin folder:

   ```
   Workflow.NET.NET2.dll
   Skelta.HWS.dll
   Skelta.HWS.WorkListChannel.Web.dll
   ```

   Create a class which implements the following interfaces

   INativeWebFormUIControl

   ISkeltaAddInProvider

   WebControl

   Inside the class, override the Render() function where you can change the default properties of the document view. In the function, change the default page url to the new page which you have created.

   ```
   using System;
   using System.Collections.Generic;
   using System.Text;
   using Workflow.NET;
   using Workflow.NET.Engine;
   using Workflow.NET.Interfaces;
   using Skelta.HWS.WorkListChannel.Web;
   using Skelta.HWS;
   using Skelta.Core;
   using Skelta.Events;
   using System.Web.UI.WebControls;
   using System.Web.UI;
   namespace CustomDocView
   {
   public class docview :WebControl, INativeWebFormUIControl, ISkeltaAddInProvider
   {
   #region INativeWebFormUIControl Members
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
private WorkItem _WorkItem;
private Context _Context;
private string _Settings;
private Guid _Id;

private WebWorkItemAdapter _WorkItemAdapter;
private string _UserId;

public docview()
{
}
public docview(string settings)
{
this._Settings = settings;
}
public void SetContext(WorkItem workItem, string channelName, string formName)
{
_WorkItem = workItem;
_Context = _WorkItem.CurrentContext;
_WorkItemAdapter = new WebWorkItemAdapter();
_WorkItemAdapter.ProcessWebChannelRequest();
_UserId = Convert.ToString(_WorkItemAdapter.LoggedInActor.RealActorId);
}
protected override void Render(HtmlTextWriter htw)
{
Workflow.NET.Parser parseHtml = new Parser();
string strTempDirectoryPath =
_WorkItem.HWSActivity.CurrentContext.ProcessDefinition.ActionsDefinition.DirTemp;
string strDirectoryPath =
_WorkItem.HWSActivity.CurrentContext.ProcessDefinition.ActionsDefinition.DirSourceEle
ments;
parseHtml.Replace("application", _WorkItem.Application.ApplicationName);
parseHtml.Replace("workflow", _WorkItem.CurrentWorkflow.WorkflowName);
parseHtml.Replace("instanceid", _WorkItem.ExecutionId.ToString());
parseHtml.Replace("submittedby",
_WorkItem.CurrentContext.SubmittedBy.Properties.Name.Value.ToString());
parseHtml.Replace("executiondetailsid", _WorkItem.ExecutionDetailsId.ToString());
parseHtml.Replace("workitemid", _WorkItem.Id.ToString());
parseHtml.Replace("activityname", _WorkItem.CurrentActivity.Name);
//parseHtml.Replace("datasubmitted", _WorkItem.CurrentContext.Data);
object cssPath = Workflow.NET.TemplateExpressionBuilder.GetUrl("");
string relativeUrl = cssPath.ToString();
parseHtml.Replace("VirtualDirectory", relativeUrl);
//Change the path of the default page which should be displayed for document view by
default.
//use the HTML controls instead of asp.net controls for designing the custom document
page
htw.Write(parseHtml.CheckAndProcessFileGetString(strDirectoryPath +
"Activities\\Activity\\CustomDocument.htm", strTempDirectoryPath + "cache",
"Activities\\Activity\\"));
base.Render(htw);
}
#endregion
#region ISkeltaAddInProvider Members
public Guid Id
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

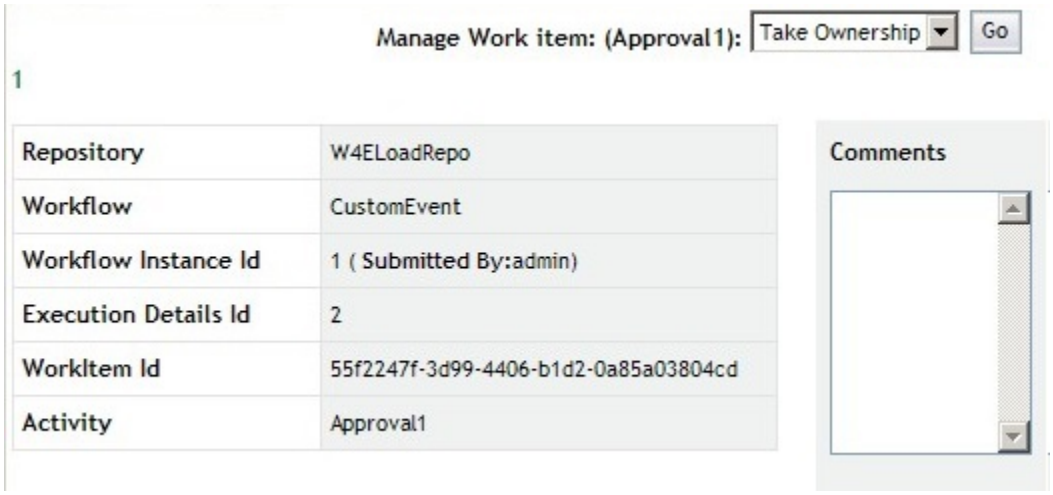
```
get { return _Id; }
}
public void InitializeProvider(string settings, Guid id)
{
_Settings = settings;
_Id = id;
}
public string Settings
{
get { return _Settings; }
}
#endregion
}
}
```

3. Compile the project and create the dll.
4. Copy the dll to the AVEVA Work Tasks bin folder.
5. The custom document view is loaded through AddInProvider. An entry has to be made to the AddInProvider table in the repository database as given below.

   You will find an existing in SKAddInProviders with 'Name' as 'DocumentWebViewControl'. Rename it and execute the below Query.

   INSERT INTO SKAddInProviders (Type,[Name],ClassName,[Assembly],Settings,IsGacAssembly) VALUES ('DocumentView','DocumentWebViewControl','Namespace.ClassName','bin\<DLLName>.dll','',1)

   As you can see in the figure below, the document view is displayed without the 'Data Submitted By' field which is present in the default Document view.



Content for the html file CustomDocument.htm

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="<%#VirtualDirectory%>Common/StyleSheet/
Global.css">


</head>

 <body leftmargin="0" topmargin="0">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 <table width="100%" border="0" cellpadding="0" cellspacing="0">
<tr>
 <td>
 <table width="100%"  border="0" cellpadding="4" cellspacing="1" class='tablebg'>
<!--<tr>
 <td colspan="2" bgcolor="#ffffff">
 <font face="VERDANA" size="1"><b>@@Execution Details@@</b></font>
 </td>
</tr>-->
<tr>
 <td width="32%" class="lefttdbg"><font class="subtitle">@@Application@@</font></td>
 <td width="68%" class="righttdbg"><font class="description"><%#application%></font></td>
</tr>

<tr>
 <td width="32%" class="lefttdbg"><font class="subtitle">@@Workflow@@</font></td>
 <td width="68%" class="righttdbg"><font class="description"><%#workflow%></font></td>
</tr>
<tr>
 <td width="32%" class="lefttdbg"><font class="subtitle">@@Workflow Instance
Id@@</font></td>
 <td width="68%" class="righttdbg"><font class="description"><%#instanceid%> (
<b>@@Submitted By@@</b><%#submittedby%>) </font>
 </td>
</tr>

<tr>
 <td class="lefttdbg"><font class="subtitle">@@Execution Details Id@@</font></td>
 <td class="righttdbg"><font class="description"><%#executiondetailsid%></font></td>
</tr>
<tr>
 <td class="lefttdbg"><font class="subtitle">@@WorkItem Id@@</font></td>
 <td class="righttdbg"><font class="description"><%#workitemid%></font></td>
</tr>
<tr>
 <td class="lefttdbg"><font class="subtitle">@@Activity@@</font></td>
 <td class="righttdbg"><font class="description"><%#activityname%></font></td>
</tr>


 </table>

 </td>

</tr>
 </table>
 </body>
</html>
```

Activities

AVEVA Work Tasks supports many activities while creating a workflow. The activities are organized in different groups based on the function it performs. Each activity has a corresponding activity icon and, you can use them

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

in the Design Area by dragging and dropping or by using the Context menu option while designing a workflow in Process designer.

You need to configure each activity by specifying appropriate values for the different properties in the Activity Properties area. The properties are described under each Activity topic in this chapter.

Note:
- A colon character (:) is not supported for the Activity Name while designing a workflow in Process Designer.
- While defining the output for activities, where you can define custom output for the action, make sure you do not use the names of the pre-existing menu items of the pull-down menu.
- Alphanumeric, space, and dot are the only supported characters for activity outputs.

Start Activity

Activity Description

The Start Activity is the first activity in a workflow. The icon corresponding to this activity is displayed by default in the Design Area. The Contents and Variables used in the execution of the workflow are declared here. The starting point for redirection can also be set in this activity. This will enable redirection at the beginning of the workflow until it is turned off later in the workflow.

Note: Workflow utilizes CDATA sections exclusively and nesting is not allowed in CDATA section. So, while passing Variables in workflow, CDATA should be avoided.

Activity Properties

The Start activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking the Activity Properties tool in the tool bar or by selecting the appropriate option from the right-click menu for the Start activity.

Note: Every time you input a value for a property, ensure to save the data. Click the **Save** button on the **Activity Properties** window. You need to save the property so that the subsequent properties in the activity can inherit the required information.

The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

You can use these properties to specify the name and description of the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

You can use these properties to specify the web service details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Variables - The variables used in the workflow, including Local Variables, can be specified using this property.

Property Type: Optional (*Set this property only if needed*.)

Steps to set the variable property

1. To specify the variables, click the button for the 'Variables' property to open the property window.
2. Enter the variable name in the **Workflow Variables** field. Prefix the variable name with '@' if it is a local variable.
3. Select the corresponding variable type from the **Type** drop down field.
4. After entering the variable details, click **Add Row** to add this variable and to display a new row for entering another variable.
5. Remove rows if necessary by selecting them and clicking **Remove Selected**.
6. After entering all the variables, click **Update**.

Steps to set Variable for Queue Action user

1. To specify the variable, right-click the Start activity icon in the workflow and select Activity Properties.
2. To declare the variable, click the icon near Variables property.
3. In the Workflow Variables, define the variable in the following format:

   ^+CurrentActivityName+'QueueActingUsers'

   For example, ^HRFormQueueActingUsers

   HRForm is the activity name.
4. Select the Type as "**Array**".
5. Click **Update** to save the variable.
6. Click **Save** in the Activity properties pane to save all the data entered.
   The value of the user who acts on the HRForm activity will be stored in the variable. The Approval1 task will be assigned to the same user.

See Start - Variables for a detailed description of the Variables property window.

Contents - The contents that can be passed to the workflow can be specified using this property.

Property Type: Optional (Set this property only if needed.)

Steps to set the contents property

1. To specify the content, click the button for the 'Content' property to open the property window.
2. Enter the content description in the **Content Description** field.
3. Select the content type from the **Type** drop down field.
4. Enter the XPath expression in the **XPath Expression** field.
5. Enter the attribute name in the **Attribute Name Blank for InnerString** field. This parameter is optional and needs to be set only if necessary.
6. After entering the content details, click **Add Row** to add this content and display a new row for entering another content.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. Remove rows if necessary, by selecting them and clicking **Remove Selected**.
8. After specifying the contents, click Update.

See Start - Contents for a detailed description of the Contents property window.

XML Variables - The XML Variables used in the workflow can be specified using this property.

Property Type: Optional (Set this property only if necessary.)

Steps to set the XML Variables property

1. To specify the XML Variables, click the button for the 'XML Variables' property to open the XML Variables window.
2. Click **New Variable** to add a new variable.
3. This will enable the **Variable Name** field. Enter the name of the new XML Variable in this field and a description in the **Description** field.
4. Now select the XML Variable type from **Type** drop down. You can select from either **XMLDocument** or **XMLIterator**.
5. Click the **Schema** tab caption 'Schema' to specify the schema for the XML variable.
   - If XMLDocument was selected in Step 4, clicking this tab will display the corresponding set of controls.
     - You can select Variable or Content as the option for internal storage from Use [ ] for Internal Storage drop down field.
     - Specify a file name or browse for the file containing XML Data. Click the Load button to load the contents of the file. The XMLschema corresponding to the loaded file is displayed in the configuration area.
       You can also build the schema from scratch within this area.
   - If XMLIterator was selected in Step 4, clicking this tab will display the corresponding set of controls.
     - You can select the XML Variable from which you can extract the node to be set as the XML Iterator.
     - Select the node to be set from the Select Node drop down field. The XML schema corresponding to the selected node is displayed in the configuration area.
       You can also build the schema from scratch within this area.

     a. Click the Save Variable button, to save the new XML Variable.
     b. Click Close to close the Property window.

See XML Variables for a detailed description of the XML Variables window.

Analyst's Process Model Filename for Execution View - This property is used to specify the file name of the Process Model, which has been saved as the Business Analyst's view. This view will be displayed to the Analyst as the Process Execution view. You can specify this property by entering the file name in the text box for the property or by building an expression for the file name using the Expression Editor, which you can open by clicking the **Expression Editor** icon next to the property name.

Property Type: Optional (Set this property only if necessary.)

SOA Object Instances - This property is used to create multiple instances of a class. Without instantiating a class, you cannot consume the methods available in a class. The Static member of the class can be consumed without creating any instance.

Property Type: Optional (Set this property only if necessary.)

See *SOA Object Instances in SOA Property topic in Developer Guide* for more information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Exception Behavior

You can use these properties to raise exceptions if the output is not linked to the next activity in the workflow and also define and configure email templates and users for the exception mails.

Raise Error on Unhandled Linked Output? – This property can be configured to raise an error if the output is not linked to the next activity in the workflow.

Property type: Optional (This property needs to be set only if necessary)

The Configurable values for this property are **Default [Backward Compatible],Yes** and **No**. If the value is set as **Default [Backward Compatible],** then the workflow will behave as per the existing behavior.

Scenario

Telematics Organization maintains different workflows like Invoice Workflow and Leave Workflow under the Telematics repository for their HR and Accounts Payable Processes. John has designed a workflow using the Human Activities and has assigned the human activities to the actors. When the activities are getting executed, if a particular actor is not found, then the Workflow Engine will give the output as 'Not Enough Resources to Acknowledge'. If this output is not linked to the next activity, then the workflow will finish successfully. However, if 'Yes' is selected for the Raise Error on Unhandled Linked Output property, the workflow will raise errors at finish.

Steps to set the Raise Error on unhandled linked output

- Select Yes or No from the **Raise Error on Unhandled Linked Output** drop-down list.

Each activity has the Raise Error on Unhandled Linked Output property. If you click on it, all the default configured mapped error outputs in red with the respective check boxes selected will be displayed.

Mapped Error Outputs

The Mapped Error Outputs list contains the error outputs (shown in red) that cause the workflow to complete abruptly. This can happen if the error outputs are not linked to the next activity in the workflow. To change the mapped error outputs, check or uncheck the relevant options and click Ok.

Workflow Behavior for Error Outputs

If the error outputs are not linked to the next activity in the workflow, and if the 'Stop with Error' option is selected, the workflow will complete with errors. Similarly, if the 'Exit Activity with Error Output' option is selected, the workflow will not finish with errors but will give an Error Encountered output.

Note: The changes made to this property will take effect only if you set the value for Raise Error on Unhandled Linked Output to **Yes** in the Start Activity.

Exception Mail Template Configuration – This property can be used to define and configure Email Templates and Users for the exception mails, whenever an exception occurs in workflow execution.

Property type: Optional (This property needs to be set only if necessary)

The Exception Mail Template and Users at the repository level are configured from the Central Configuration site

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

section. For more information, refer to *Configuring Exception Mail Template* under the *Administration Tasks* in the *Administrator's Guide*.

Property type: Optional (This property needs to be set only if necessary)

You can configure Exception Email Templates for two groups - Group A and Group B, both at the workflow level.

It is not mandatory to set both Users Groups A and B.

For example Administrator type of users can be configured under Group A and Normal type of users can be configured under Group B.

Email Templates can be configured through two modes – one is by selecting an existing predefined Default or Sample template, and the other is by using an editor.

Based on the Exception Mail Configuration, the members of each group will receive mails.

The Template file has to be placed in the folder 'Exception Template' under the 'Workflow elements' folder. In the Template file, we can place the Variables, Contents and XMLVariables Placeholders in the html file.

Scenario

John has designed a workflow using Human Activities and has assigned these activities to the actors. When the activities are getting executed, if a particular actor is not found, then the Workflow Engine will give the output as 'Not Enough Resources to Acknowledge'. If this output is not linked to the next activity, the workflow will finish successfully. However, if the Raise Error on Unhandled Linked Output property is selected as 'Yes', then the workflow will raise errors at finish.

There should be a notification to the users about the Exception occurred in an activity. These users should be configurable and the template for the exception mails should be configurable.

Steps to configure the Exception Mail Template

1. Click the edit button of the **Exception Mail Template Configuration** property. The **Exception Mail Settings** window appears.

   Users can be configured through the User Lookup screen or the Role Lookup screen. By default, the User is selected.
2. Click the **User Lookup** icon. The **User Lookup** screen appears.
3. Select the required Provider from the **User-based Advanced Search in provider** drop-down list.
4. Key-in the name of the user in the **Search For** field and select the Display Name from the drop-down list.
5. Click **Search**.
6. All the users get listed in the **Available Resources** field.
7. Select the required user and click the right arrow to display the users in the **Selected Resources**.
8. Click **Update**.
9. To configure a user through the Role Lookup, select the **Role** radio button.
10. Select the required provider from the provider drop-down. Also, note that the provider drop-down doesn't appear, if the repository has a single provider only.
11. Key-in the name of the required role and click **Search**.
12. All the users get listed in the **Available Roles** field.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

13. Select the required role and click the right arrow to display the users in the Selected Roles.
14. Click **Update**.
    The user's name (either User or Role) will now appear in the **Exception Mail Group A** field.
15. Select the template from the **Template File** drop-down list.
    You can select either a Default template or a Sample template.
16. Follow the same steps to select a User or Role and a template for the **Exception Mail Group B**.
17. Click **OK**.

To create your own template for the exception mail

1. Select the **Create/Open** option in the **Exception Mail Settings** screen. The Editor screen appears.
2. Select the required values from the **Field** and **Standard Placeholders** drop-down lists and click **Append**.
3. Use the editor to design your template and click **Update** and **Close**.

Note: You must use only the actor variable, and not any of its sub properties, to initialize a variable of type actor.

Start - Variables

Variables Property Window

You can set the workflow variables in the Variable property window. To open this window, click the button in the 'Variables' property. The Variables dialog box is displayed in a grid, which has a blank row where you can enter the parameters for a new variable.

The dialog has two tabs at the top to manage the rows of variables in the grid.

- **Add Row** - You can click this tab to add the newly entered variable and display a new row to the grid where you can add another variable.
- **Remove Selected** - You can click this tab to remove the selected variable in the grid.

The grid has the following parameters that you need to specify for the new variable.

- **Workflow Variables** - You can use this field to enter the variable name. Prefix the variable name with '@' if it is a local variable.
- **Type** - You can use the drop-down list in this field to select the data type for the variable. The drop-down list displays the following data types - string, number, date, array, actor, object, boolean, float, and decimal.

There are two buttons and a check box at the bottom of the dialog box.

- **Update** - You can click this button to save the newly added variable(s).
- **Close** - You can click this to close the property window without saving the newly added variables.
- **Last Row needs to be added** - You need to select this check box to save the last row added.

Note: To initialize the **Date** variable, enter the date in mm/dd/yyyy format.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Start - Contents

Contents Property window

You can set the workflow contents in the Contents property window. To open this dialog box, click the button in the 'Contents' property. The contents are displayed in a grid, which also has a blank row where you can enter the parameters for a new content.

The dialog box has two tabs at the top to manage the rows of contents in the grid.

- **Add Row** - You can click this tab to add the newly entered content and display a new row to the grid where you can add another content.
- **Remove Selected** - You can click this tab to remove the selected content row in the grid.

The grid has the following parameters that you need to specify for the new content.

- **Content Description** - You can use this field to enter the content description.
- **Type** - You can use the drop down in this field to select the data type for the variable.
- **XPath Expression** - You can enter the XPath expression for the content in this field.
- **Attribute Name Blank for InnerString** - You can enter the attribute name in this field to get the XML attribute. If left blank, it will retrieve the XML value.

There are two buttons and a check box at the bottom of the dialog.

- **Update** - You can click this button to save the newly added content(s).
- **Close** - You can click this to close the property window without saving the newly added contents.
- **Last Row needs to be added** - You need to select this check box to save the last row added.

XML Variables

AVEVA Work Tasks uses XML variables to provide data inputs to workflows. XML variables can be configured in the Start activity through the XMLVariables property.

About XML Variables

AVEVA Work Tasks uses XML variables to provide data inputs to workflows. XML variables can be configured in the Start activity through the **XMLVariables** property.

Different Types of XML Variables:

The following table lists different XML variables:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

XML variable	Description
XML Document	Using 'XML Document' variable, the required items can be fetched.
XML Iterator	'XML Iterator' variable allows you to define variables for repeating XML elements and attributes.
XML List	Using 'XML List' variable, the data from any list can be fetched.
XML List Iterator	'XML List Iterator' variable is used to iterate through the list items (parent items) of a list and helps to retrieve or update data from a workflow.
XMLForms Data Grid	'XML Forms Data Grid' variable is used to read or modify the selections of a Data Grid control used in Forms or Lists

The XML Variable Interface provides the following functionality:

- Defining XML schema for the XML Variable either by specifying the complete schema or loading an XML document which is the source of the schema
- Defining XSL transforms for the defined variable
- Defining variables of type XML Document to retrieve data from XML files
- Defining variables of type XML Iterator to retrieve data within repeating nodes within an XML schema. The XML Iterator represents a collection of XML Nodes at some level in the XML variable schema. It allows you to perform actions on XML nodes such as looping through XMLnodes and getting the value of a specific node at any location.
- Defining variables of type XML Forms Data Grid to retrieve or modify selections of Data Grid Control used within Forms or Lists which maps with the Grid Configuration for the selected XML node of the Form

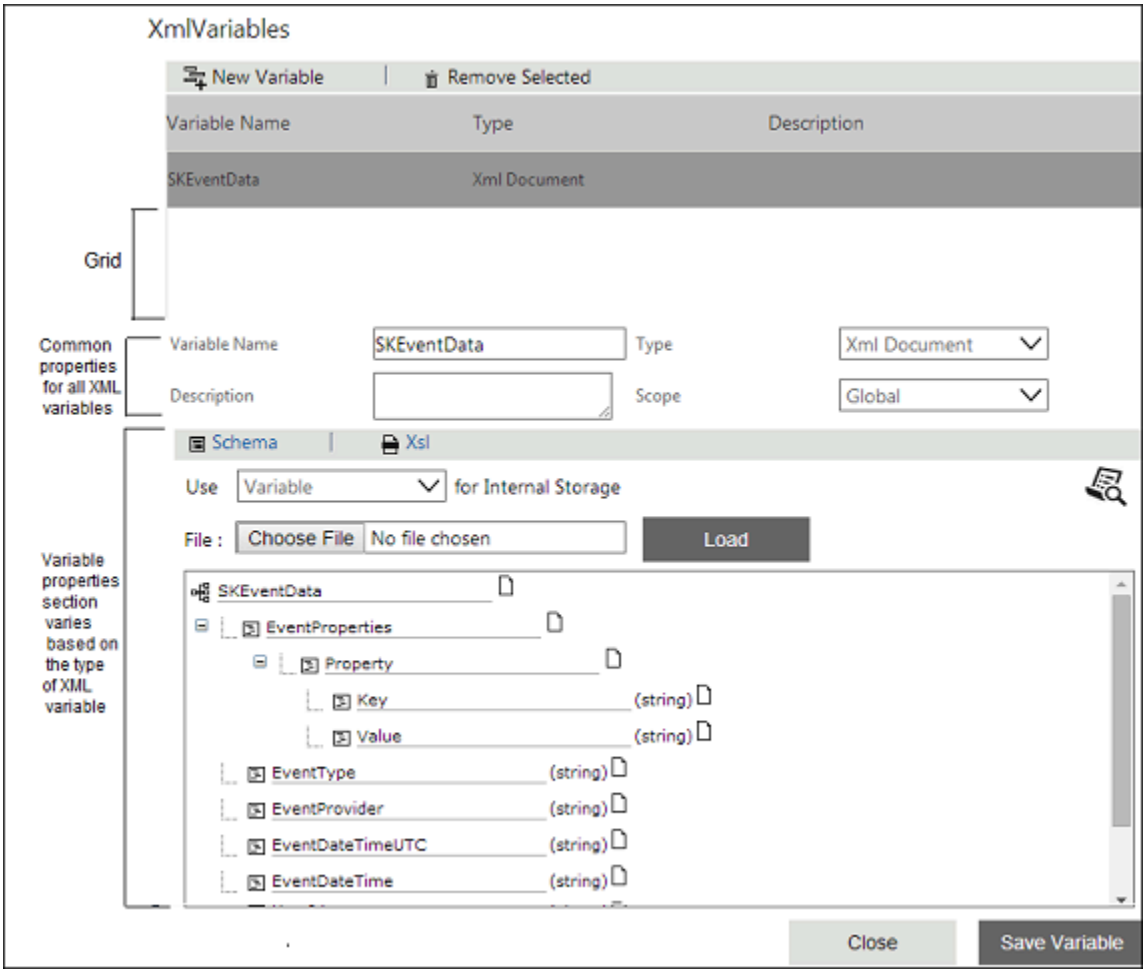
Note:
- The special characters such as ! = | ~ @ # $ % ^ & * ( ) + ` - { } : " < > ? [ ] ; ' , . / and space are not supported for XML Variable names.
- Support for Special Characters in XML node Names of XML Variables: Special characters such as, < > { } * ? [ ] ( ) - _ , = | ; ` +, and % are allowed while naming XML node of XML Variables. XMLVariable will not support **XML node names** with special characters, such as \ ~ ! ' : " # ` & . @ ^ and $.

XML Variable User Interface

The XML Variable User Interface is for configuring XML Variables. This window is opened when you click the **XMLVariables** property of the **Start** activity. All XML Variables added for the workflow are displayed in rows in the **grid**. The variable properties that are specific to a XMLVariable are displayed in the Variable Properties section for an XML Variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

XML Variable User Interface



Common Properties Available for All XML Variables

The following properties are common for all XML Variables:

XML Variable Property	Description
Variable Name	This property is used to specify the name of the variable.
Type	This property specifies the type of the XML Variable. The available XML Variable types are XML Document, XML SAPTable,XML ListIterator, XML List and XML Iterator.
Description	This property specifies the description of the XML Variable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

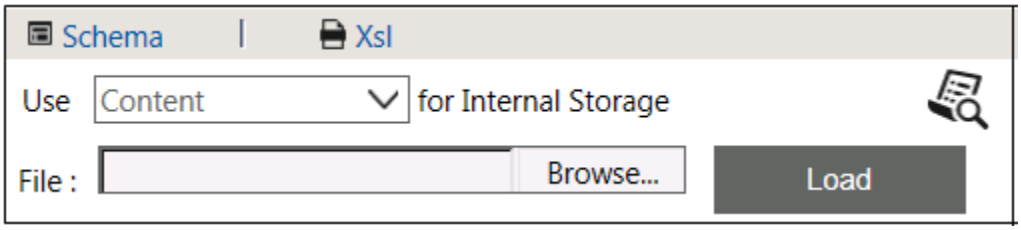
XML Variable Property		Description
Scope	Global	If the XML Variable is of Global scope, then you have to set the value of this field as Global for the variable. Where that variable can be used anywhere in that workflow instance.
	Local	If the XML Variable is of Local scope then the value of the variable is local to a workflow instance.

XML Variable of Type 'XML Document'

An XML Variable of type **XML Document** can be used to fetch the required items.

XML Variable of type 'XML Document'

When the variable type is 'XML Document', the following properties are displayed as seen in the image below.



Property Description

XML Variable Property	Description
Internal Storage	This property is used to specify the internal storage type of the XML Variable. • If the selection is Content, then XML Variable value will be stored as content in workflow context. You cannot update values for such XML variables, since the content data cannot be changed during workflow execution. • If the selection is Variable, then the same will be stored as a variable. You can update the values for such XML variables using "Update Variable" activity under engine activities.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

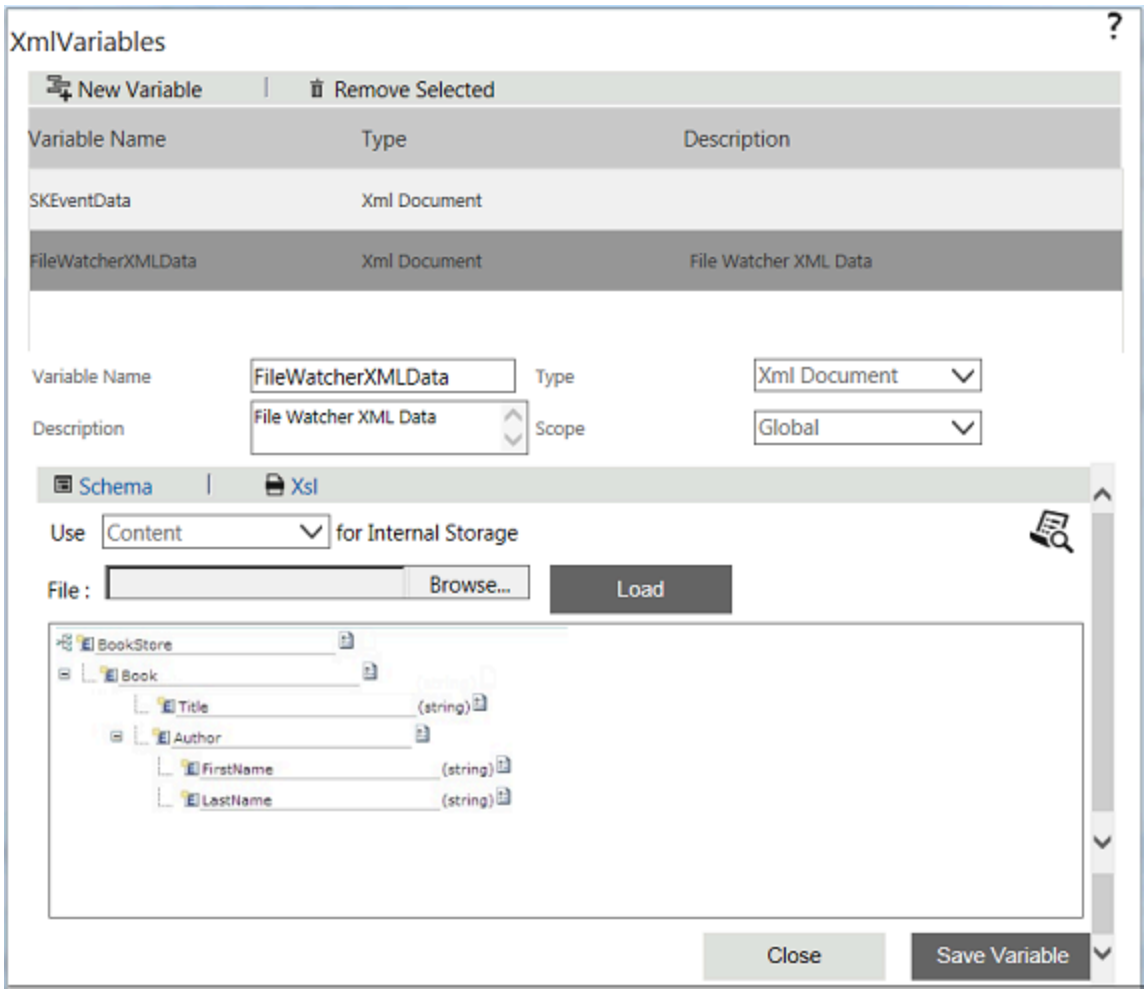
XML Variable Property	Description
	Note: The default storage option of XML variable is Variable.
File	Location of the schema file, if any XML schema is available. Otherwise user can create the schema using Schema designer.

Create XML Variable of Type 'XML Document'

To add a XML Document Variable

1. Click **New Variable** button in the XML Variable User Interface.
2. Enter the following information:
   - **Variable Name**: Enter the name of the variable.
   - **Type**: Select XML Document from the drop-down list. This field defaults to XML Document. The XML Iterator type option is available only if another variable of XML Document type exists.
   - **Description**:- Enter a description for the variable.
   - **Scope**: Select the scope of the variable. The scope of the variable can be either Global or Local. Select Local, if the value of the variable is local to a workflow instance.
3. **Schema**: Build the XML Schema. You can either build the schema or use the existing XML file to create the schema.
4. Specify how the variable is to be used in Use for Internal Storage field. The field defaults to Content. To store as variable, select Variable in this field. If the scope of the variable is **Global**, you must set this field to **Variable**.
5. Click **Save Variable** to save the new variable.
6. Click **OK** to save the node properties.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



See Also: Build XML Schema

Build XML Schema

You can build a XML Schema in two ways:

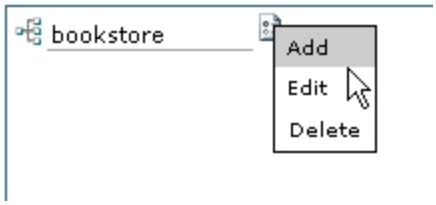
1. Define schema in the XML Variable interface
   a. Adding a Node
   b. Editing a Node
   c. Deleting a Node
2. Loading schema from an existing file

Defining Schema in the XML Variable Interface

When a new variable is created, the schema displays the root node by default. Click the string "root"and type the name over it.

The following image shows the node options:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



To add a node

1. To add a node, click the Properties icon to the right of the name. Select **Add**. The **Properties** window for the new node appears.
2. Specify the following properties of the node:
   - **Name**: Enter a name of the node.
   - **Type**: Select the node type. The options available are **element** and **attribute**.
   - **Data Type**: Select the data type. This property specifies the data type of the XML Node. The available data types are string, integer, long, double, decimal, date, time, dateTime, boolean, and base64binary.
   - **Repeating**: Select this check box, if the node is a repeating one.
3. Click **OK**.
4. Click **Save Variable** to save node properties.
5. Click **OK**.
6. To add more nodes, repeat step 1 to step 6.

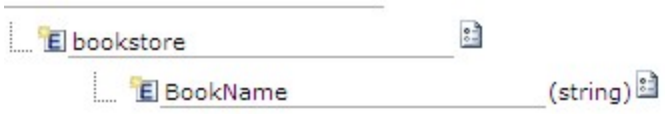
Note: Select the respective node where you want add a node. If a node is of Attribute type, then you cannot add a node under it. The Add option will be disabled for Attribute type of node.

To edit an existing node

1. To edit an existing node, click the **Properties** icon to the right of the node to be edited.
2. Select **Edit**. The Properties window for the existing node appears.

3. Modify properties in the Properties window.
4. Click **OK**.

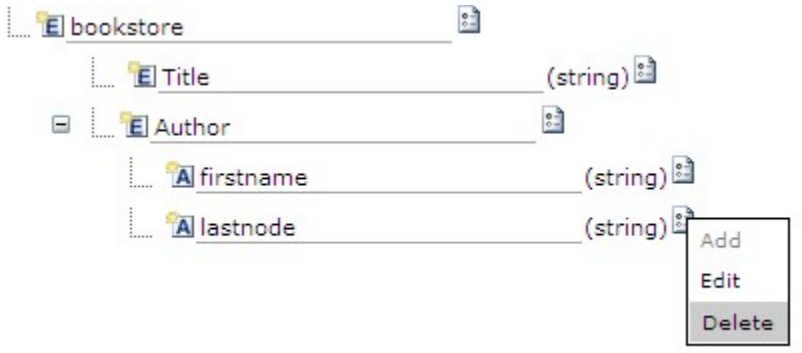
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



5. Click **Save Variable** to save node properties.
6. Click **OK**.

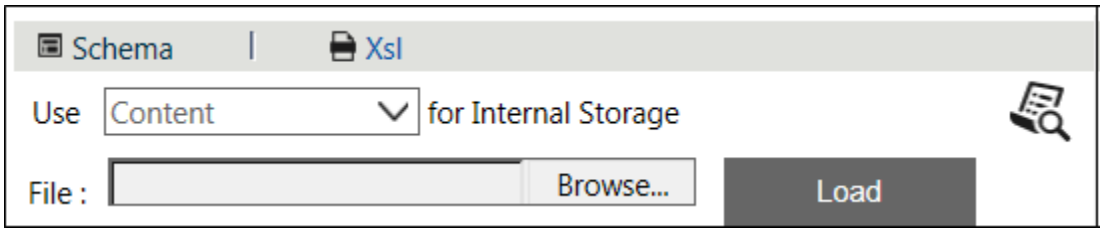
To delete a node

1. To delete a node, click the **Properties** icon to the right of the node to be deleted.
2. Select **Delete**



To load Schema from an Existing File

1. Click **Browse** to select file with schema information. You can select XML files to load schema



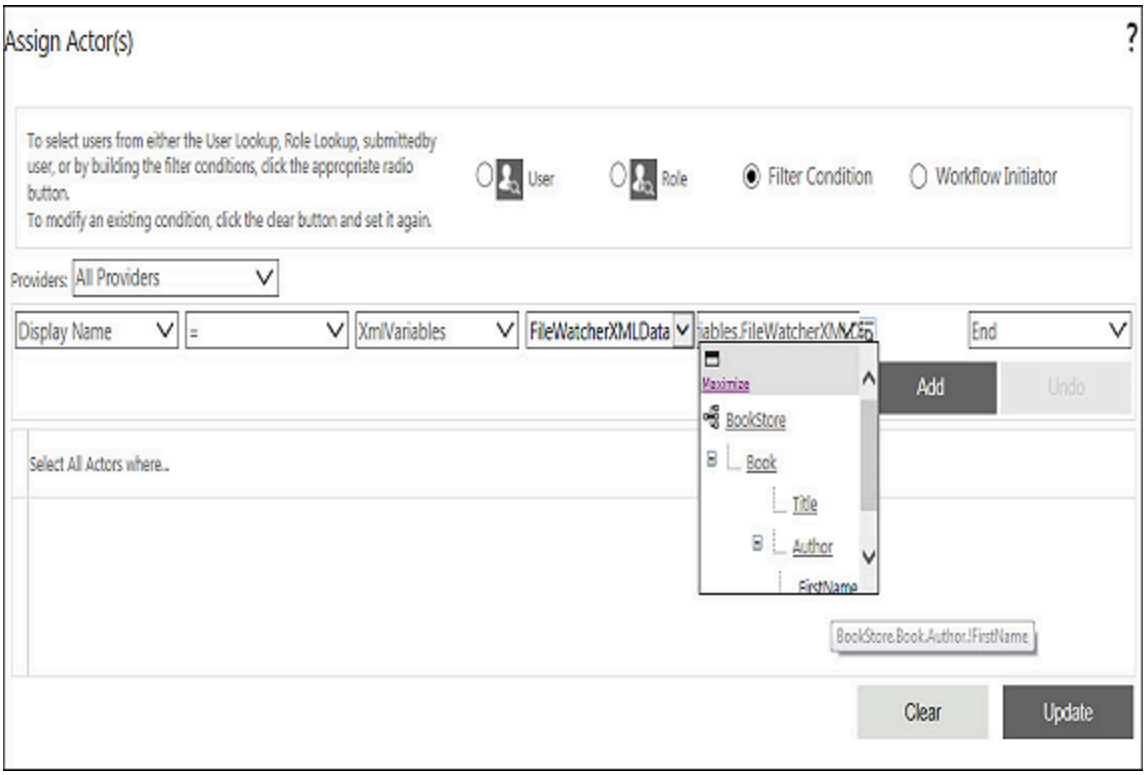
2. The XML schema is loaded from the file.
3. To edit a node, right-click the node, and then select **Edit**. Modify properties in the Properties window.
4. To delete a node, right-click the node, and then select **Delete**

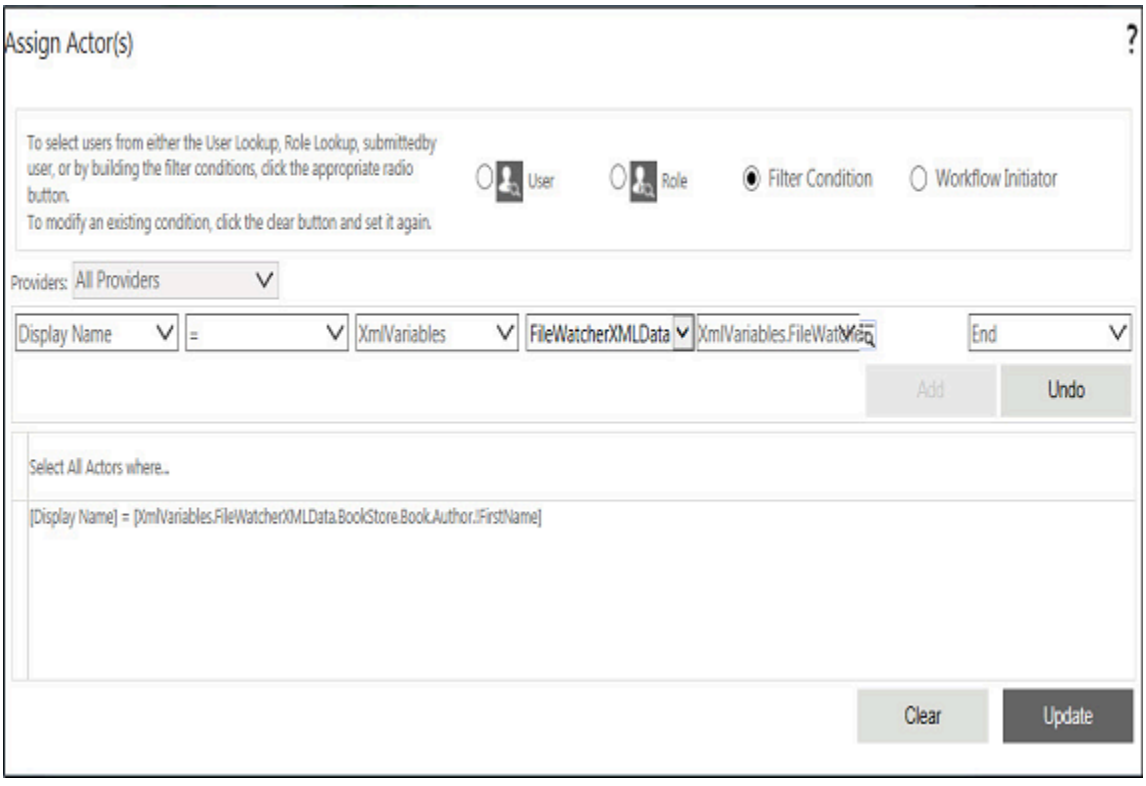
Use XML Document of Type 'Variables'

XML Variables can be used like regular workflow variables or contents to build conditions for setting Activity Properties. In the example below, the condition for the Assign Actor property of the **Approval** activity has been built by specifying the "firstname" element from the "Author" node of the XML Variable.

Example of using XML Variable to set the Assign Actors property

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



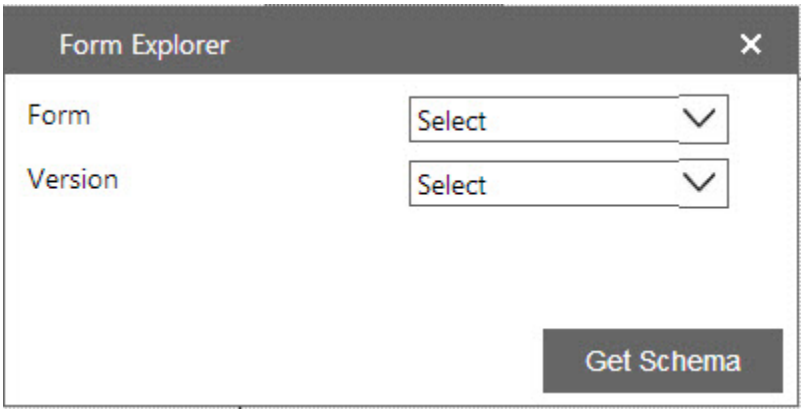


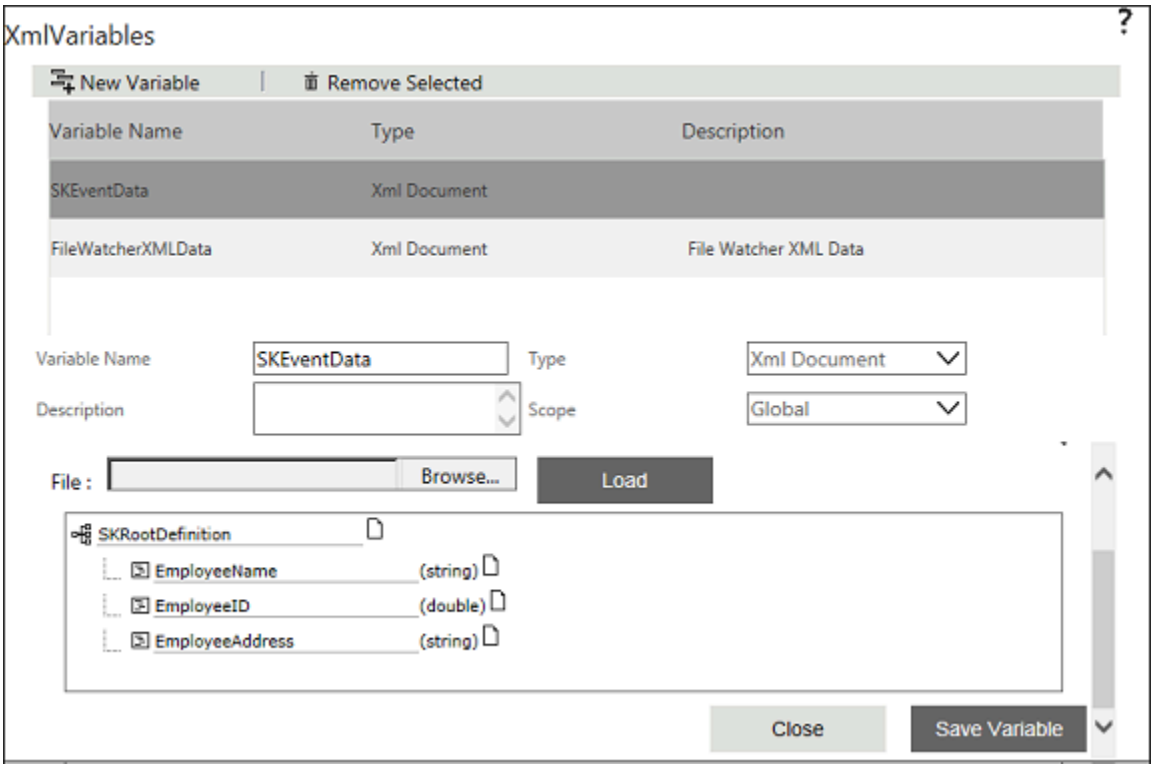
Determine XML Variable Schema from Forms

After a form is associated to a particular workflow, the form creates the XML Variable named **SFFormData** of type **XML Document** in the **Start** activity of that workflow. This will have the schema of the form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

AVEVA Work Tasks provides the **Form Explorer** icon to choose the form and its version. This icon will get displayed only for XML Variable of type XMLDocument. Any XML Variable of type XML Document can now get its schema from any form and its version available in the **Manage Forms** List.





Before introducing, when some changes were done in the form, the XML Variable had to be deleted and the form had to be re-associated in order to get the updated list of controls.

For more information, refer Forms and Lists Guide.

XML Variable for List

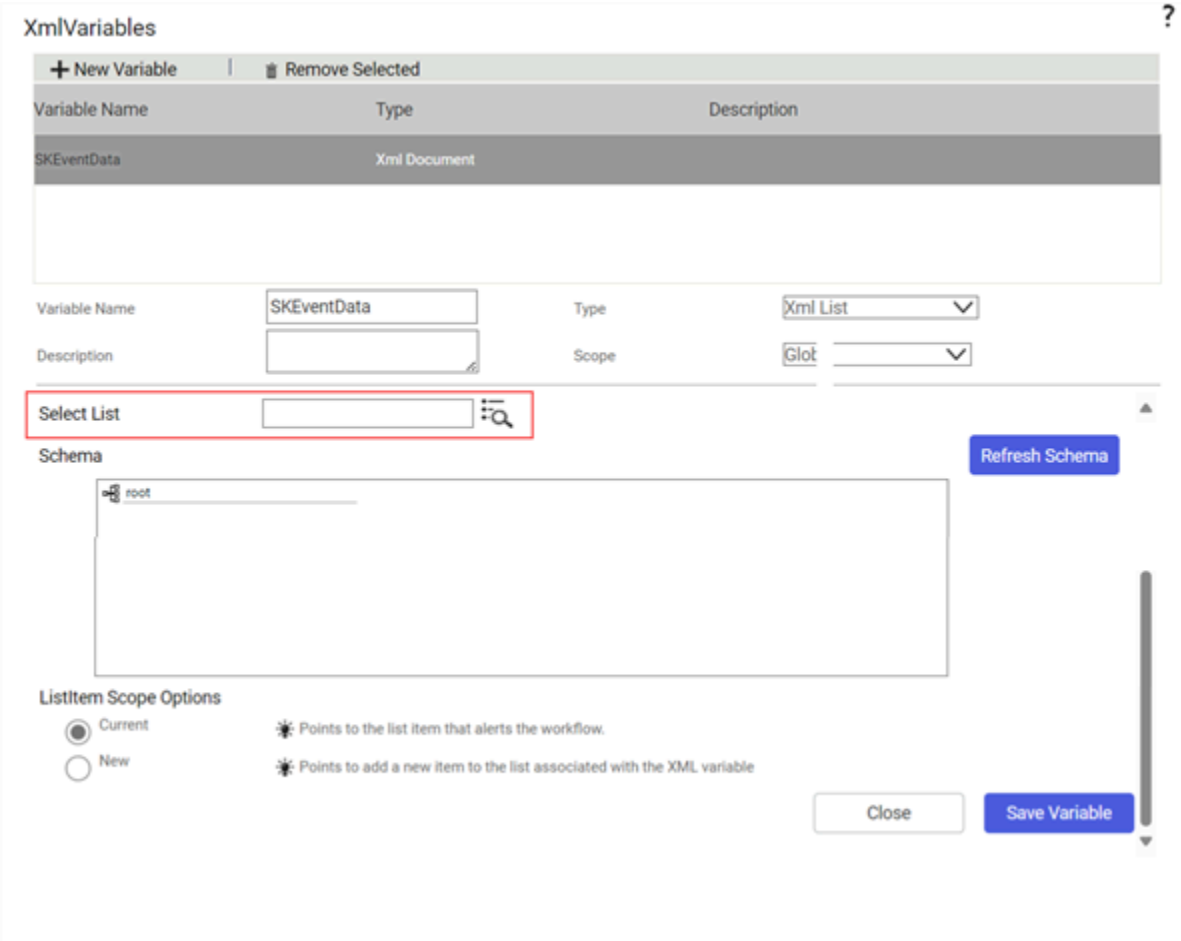
An XML List type XML Variable can be mapped to any list schema, that is either to lists provided out of the box or to lists created dynamically using create list option, to perform retrieval of data of a list item.

Using XML list XMLVariable, you can fetch the data from any list. XML Variable for XML list can be mapped to any XML List schema for inserting, updating and retrieving list items. Also you can attach the list to a workflow very easily in a few steps.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

XML Variable of type 'XML List'

When the variable type is changed to **XML List**, the following extra properties and **ListItem Scope Options** are displayed as seen in the image below.



Property Description

'XML List' XML Variable Property	Description
Select List	User can enter list name in this field or use the icon to select the required list.

The following are the ListItem Scope Options:

- **Current:** The XML Variable will be pointed to the list item that alerts the workflow.
- **New:** This points to add a new list item to the list associated with the XML Variable. Accessing this XML Variable before adding an item will not give any details. After adding a list item, this XML Variable will have the details of the added item. To add another item, use a **List Navigator** activity to re-initialize the scope as New.

See Creating an XML List type XML Variable for more information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

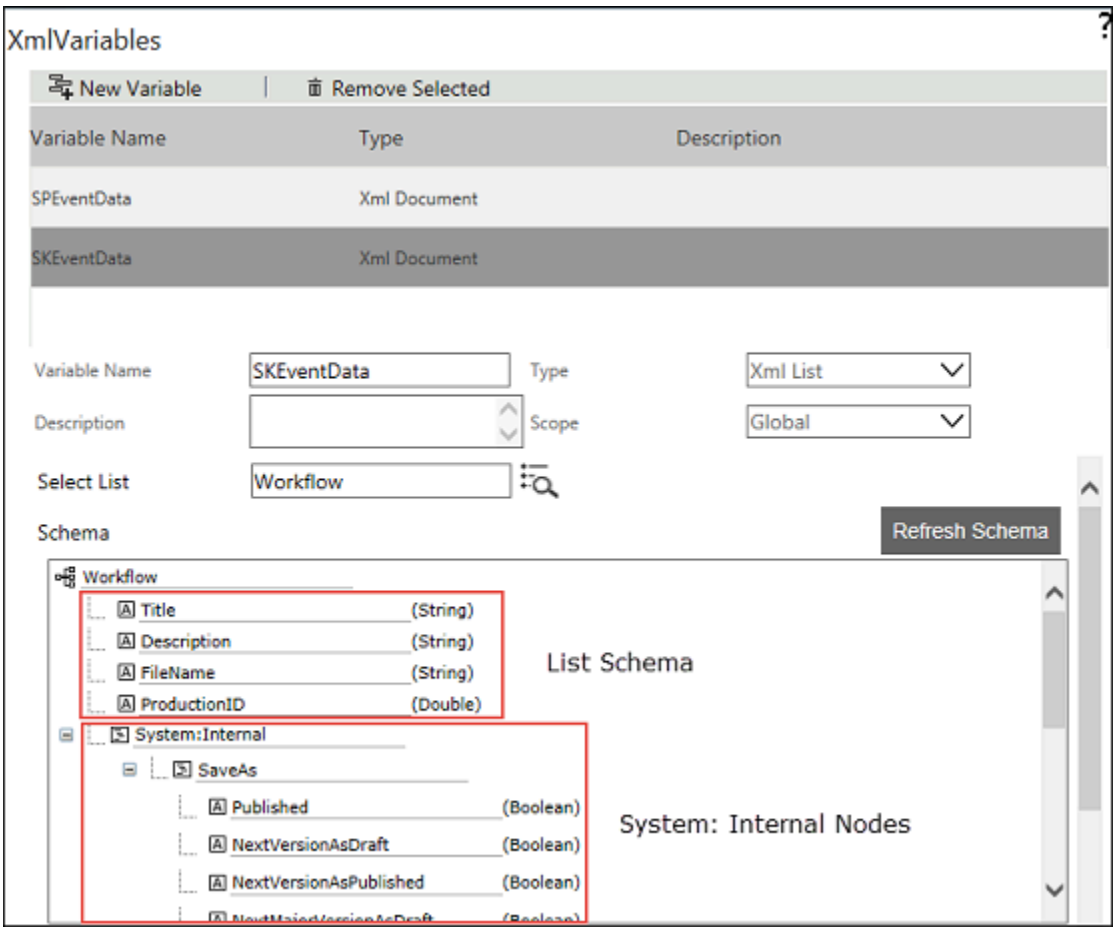
ListItem Scope of XML List of Type XML Variable

List XML variable can be set to **Current**, **New** and **List Query** scopes. You can set Current and New scopes at the time of creating XML List type XML variable and all the three scopes including List Query can set up in the List Navigator activity properties.

- **Current:** The XML variable will be pointed to the list item which initiated the workflow.
- **New:** The XML variable will be pointed to add a new list item to the list. Accessing this XML variable before adding an item will not give any details. After adding a list item, this XML variable will have the details of the added item. To add another item, use a list navigator action to re-initialize the scope as New.
- **List Query:** This property allows you to build a query for the list and fetch the items based on the query.

Schema of XML List of Type XML Variable

The schema of XML list includes schema of the list and system internal nodes.



System Internal Nodes

When you create a list XML variable for the list, system internal nodes are also created automatically for internal use. This will help you to save your list item as a different version as **Published** or **Draft** status when you update an item. It will also help you to save the item **As Folder** and **Published**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If you want to add a new listitem with **Published** status, select **Published** from the **SaveAs** option under **System: Internal** and then set value as **True**.

Create an XML List of Type XML Variable

XML Variable of List type can be configured in the Start activity through the XML Variable property. XMLVariable of List type can be mapped to any list schema i.e. either to lists provided out of the box or to lists created dynamically using the create list option, to perform retrieval of data of a list item.

To create an XML List type XML Variable

1. Select the **Design the Workflow now** option from the **Workflow Options** screen while creating a workflow for the list.

   The **Process Designer** screen appears.
2. Right-click the **Start**, and then select **Activity Properties** from the options. The Activity Properties screen appears.
3. Click the **edit** icon in the XMLVariables field. The XmlVariables screen is displayed
4. Click **New Variable** to add a new variable.
5. Type a variable name in the **Variable Name** field.
6. Select X**ML List** from the Type drop-down list.
7. Type the List name in the **Select List** field or click the icon to select the list.
8. The XML schema displayed in the tree-view are the XML nodes that are mapped to the controls while creating the list. Also you can see the System Internal nodes that are created automatically for internal use. This will help you save the list item in another version as draft or published and help to create folder.
9. Select the scope from the **Choose the ListItem to work on** section. The available options are:
   - **Current**: This will help to work with the list item that initiated the workflow.
   - **New**: The XML variable will be pointed to add a new list item to the list.
10. After entering all the required data, click **Save Variable.** A message is displayed saying 'Variable Updated'. Close the window. Now you have created an XML List type XML Variable.

    Note: You can modify the variable later. Click the **Refresh Schema** to reload the changes made to the list. After refreshing the schema, click **Save Variable** to save the changes.

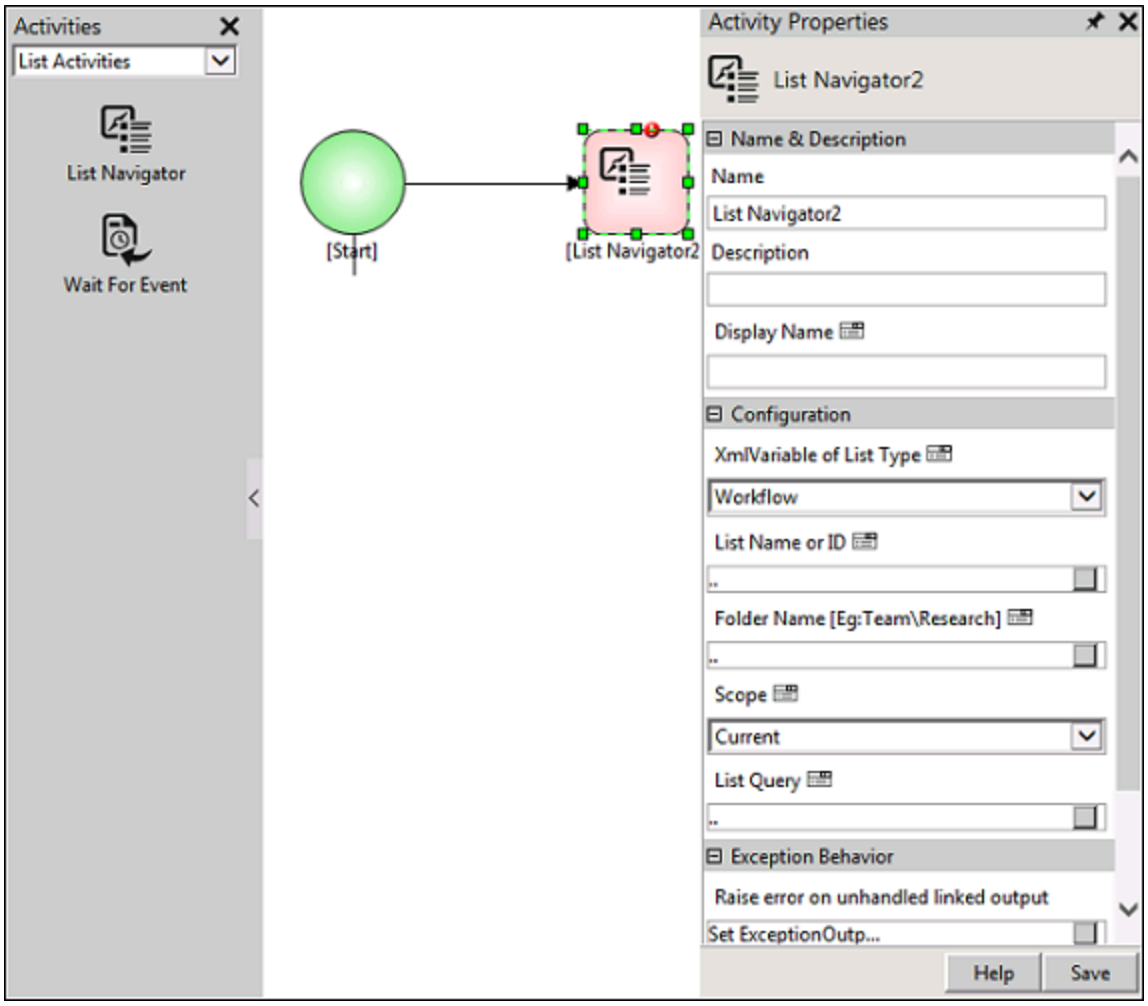
Change Scope of the XML List of Type XML Variable

The scope (**Choose the list item to work on** options) of the XML list XML variable can be changed during the runtime. Once the scope is changed, the XML list XML variable will use the changed scope. You can change the scope of the XML list XML Variable using the List Navigator activity during workflow execution. Using this action, you can also navigate the XML list XML variable to another list which has a similar schema.

To change the scope of the XML List XML Variable

1. Click **Activities** and select **List Activities** from the options.
2. Drag and drop **List Navigator** activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Select the XML variable, for which the scope needs to be changed, from the **XML Variable of List Type** drop down.
4. Type/select the list name in the **List Name or ID** field. This property helps to change the list that you need to work on. The list entered here should have the same schema of the list from which the XML variable was created. If you do not want to change the working on list, then leave this field blank. Once you change the list, the XML variable will use the changed one.
5. For example, you have a list called **Purchase Orders** and another list called **Approved Purchase Orders** which has the same schema as the **Purchase Orders** list. If you want to add/update some item in the **Approved Purchase Orders**, you do not have to create separate XMLvariables for both the list. You can use the same XML variables since both lists have the same schema. In this case, create XML variable for Purchase Orders. Enter the List Name as Approved Purchase Orders and set the scope as New. This will point to the Approved Purchase Orders list and add/update the item in that list.
6. Click the drop-down and select the scope from the options. The available options are **Current**, **New** and **List Query**. Here you can change the scope.

Note: You can also use list id. Id should be inside curly braces. Format to use {ListId}. This is optional and list id need not be given when the list name is provided and vice versa. Also, they can provide both as {ListId}ListName format.

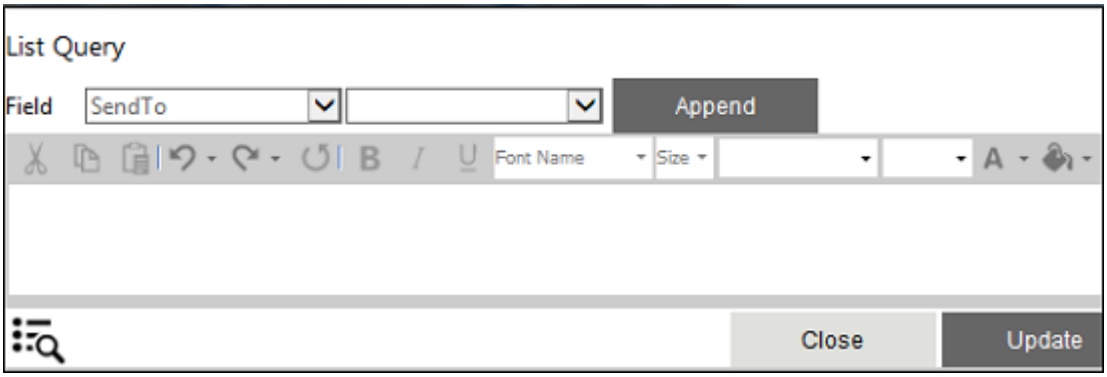
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Set up List Query Scope

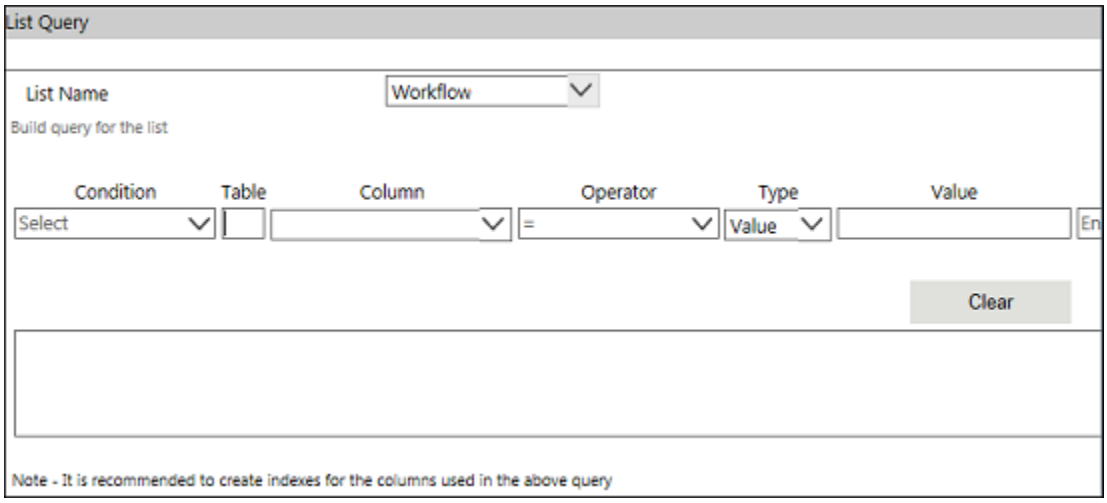
This property allows you to build the query for the list and retrieve the data based on the query. You need to point to any particular item to perform a task based on the respective item. This can be done by building the query using List Query.

To set up the list query scope

1. Select **List Query** as the scope and click **Save**. Click the edit icon of **List Query** property. The following screen is displayed.



2. Click the search icon. The following screen appears.



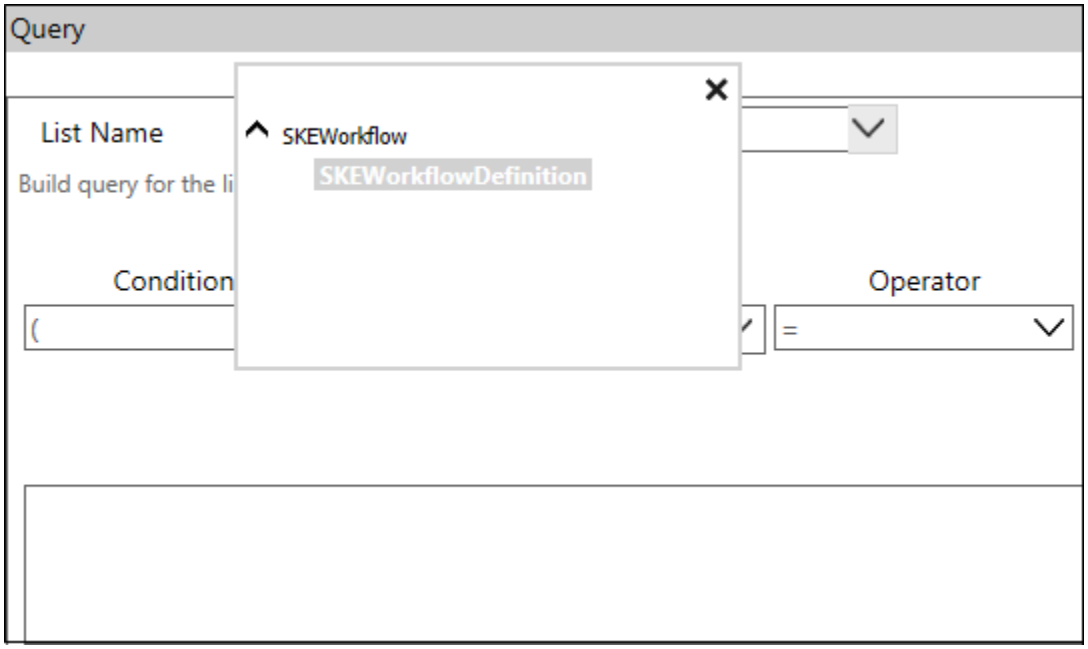
3. The following are the field values for List Query window:

Field	Description
List Name	Select the list from List Name drop-down.
Table	By clicking on this text box, a pop up allows you to select the corresponding table for query. The respective table name will display in the pop up window based on the list you select.
Column	From this drop down you can select the column or

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	field of the corresponding list.
Operator	From this drop down you can select an operator (either one of =,>, <,>=, <=, <>, IN, Not IN, Like) to build the query.
Type	From this drop down you can select a type. The value will be automatically displayed in the text box.
Condition	From this drop down you can select the condition [either of End, And, Or, '),' '('.
Add	You can click this button to add a query in the field.
Clear	You can click this button to clear the query contents.
OK	You can click this button to save this query.
Close	You can click this button to close this window.

4. Enter the details, and click **Add** to add the List Query.
5. Click **OK** and then click **Close**.
6. For example, you have a list named Workflow and you want to update the records that have the Id value '12345' and '6789' . This query can be built using list query.
7. Click the icon to build the query for the list Workflow. The **List Query** screen appears.
8. The list name *Workflow* is selected automatically. If the list is not selected, select your list from the **List Name** drop-down. Click the **Table** field. The pop up window allows you to select the table of the '*Workflow*'.

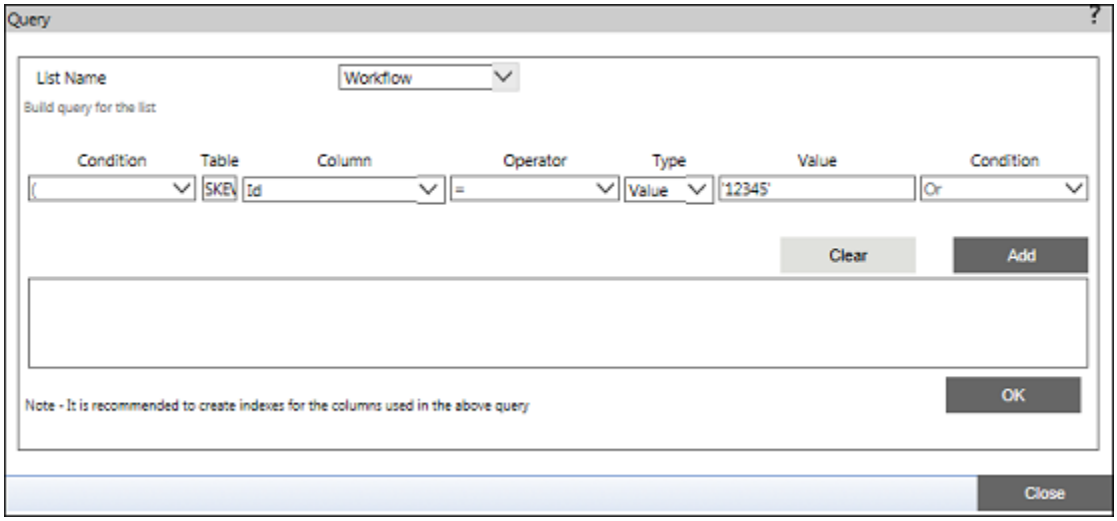


9. Click **SKEWorkflowDefinition**. This will be displayed automatically in the text box. Here, the table names

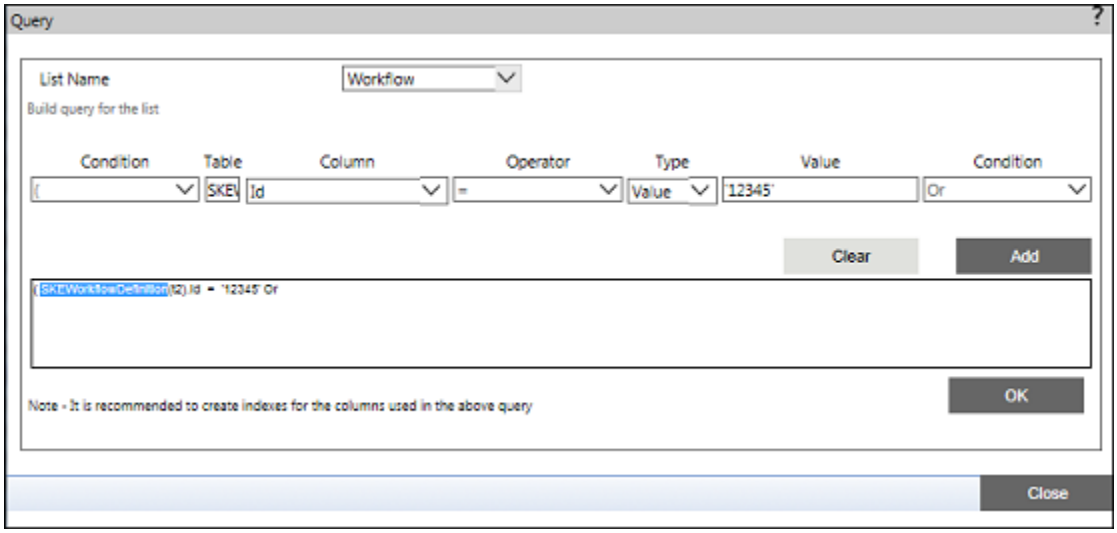
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

displayed in the tree view are the parent table (one on the top) and the child table (the one below).

10. Click the **Column** drop-down.
11. Select **int1** from the options since the field **Reference No** is mapped to this column of the database.
12. Select '=' operator from the **Operator** drop-down list.
13. Select **Value** from the **Type** drop-down list.
14. Type **'12345'** in the text box.
15. Select **OR** from the **Condition** drop-down list to add another query.

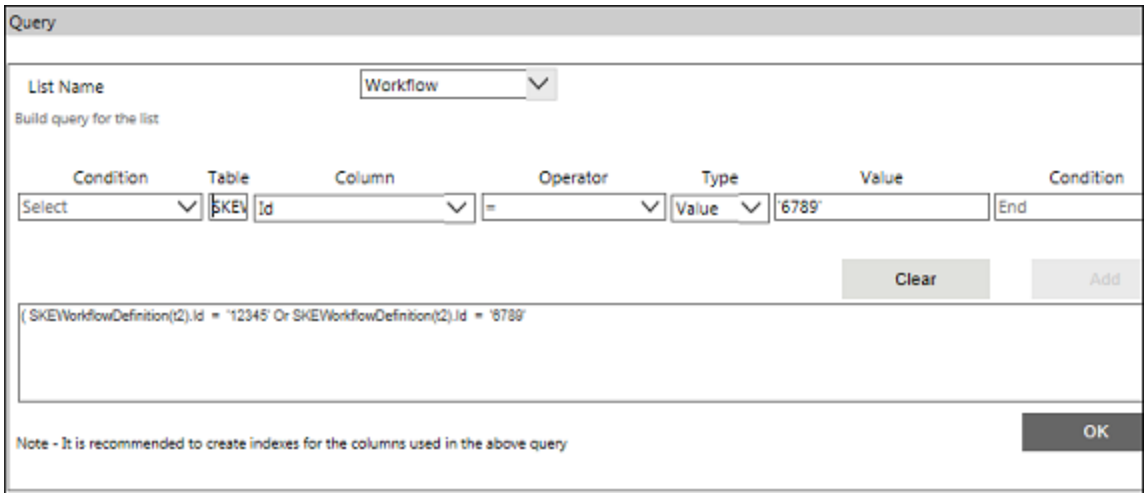


16. Click **Add** to add the condition in the text field.

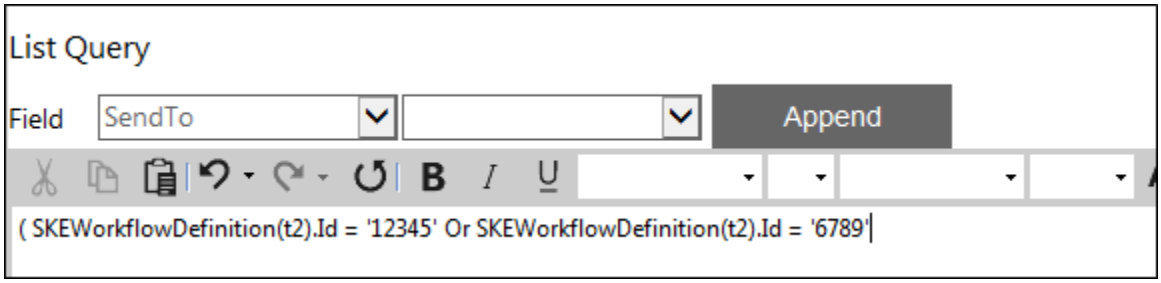


17. Here, **SKEWorkflowDefinition** is the table name where the list has been stored and **t2** is the internal name that has been used for the main table. If there is any child table, it will be displayed as **t2**. Id is in the field name, to which you want to point your XML variable and **12345** as the value for that field.
18. Repeat the same steps for the list item that has **'6789'** as **Reference No**.
19. Enter the value as **'6789'**.
20. Select **End** from the **Condition** drop-down list.
21. Click **Add**. The build query will be displayed in the text field.

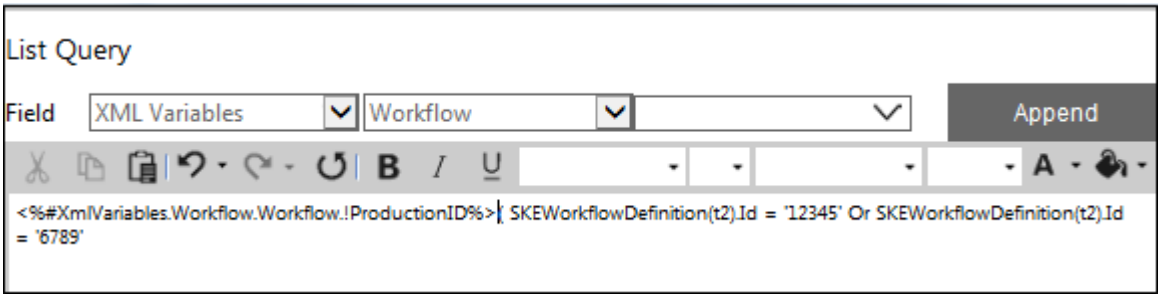
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



22. Click **OK**, and then click **Close**.



23. Click **Update**.
24. You can replace the value for the query, with the variables. To replace, select the text/value to be replaced, and then select the respective field from the drop-down list and click **Append**.



25. Click **Update**, and then close the window.

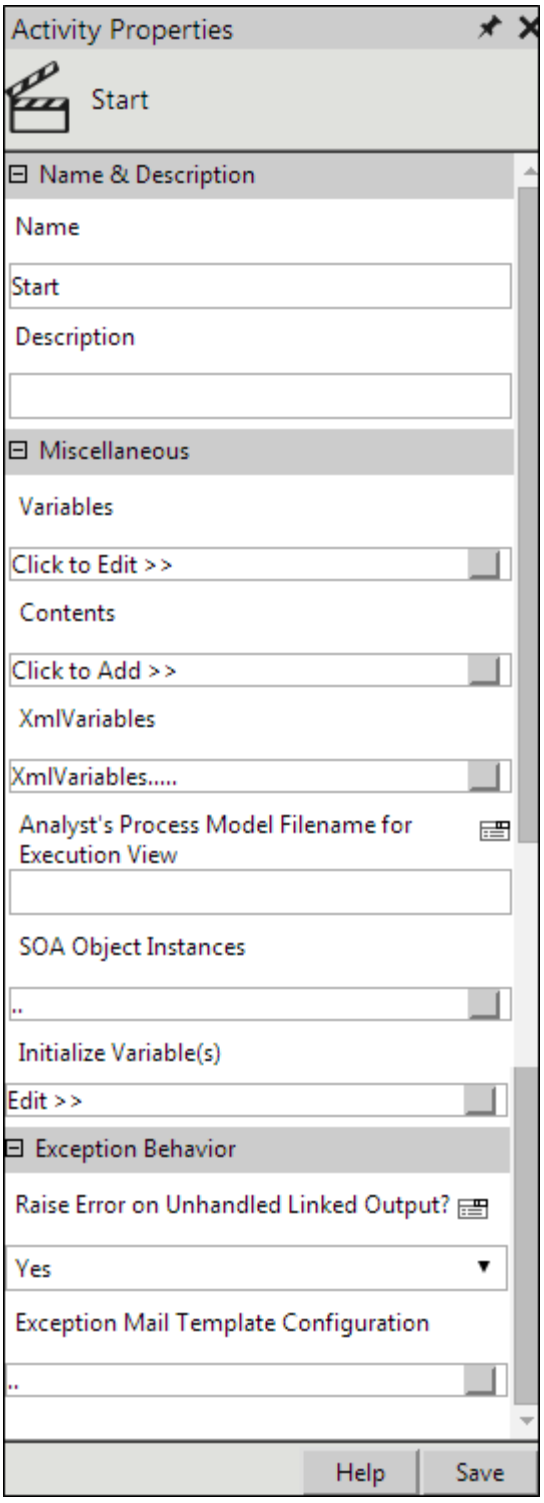
Create an XML List Iterator of Type XML Variable

XML List Iterator type XML variable is used to iterate through the list items (parent items) of a list and helps to retrieve/update data from a workflow.

To create an XML List Iterator type XML Variable

1. Right-click the **Start** Activity, and select **Activity Properties** from the options. The **Activity Properties** appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

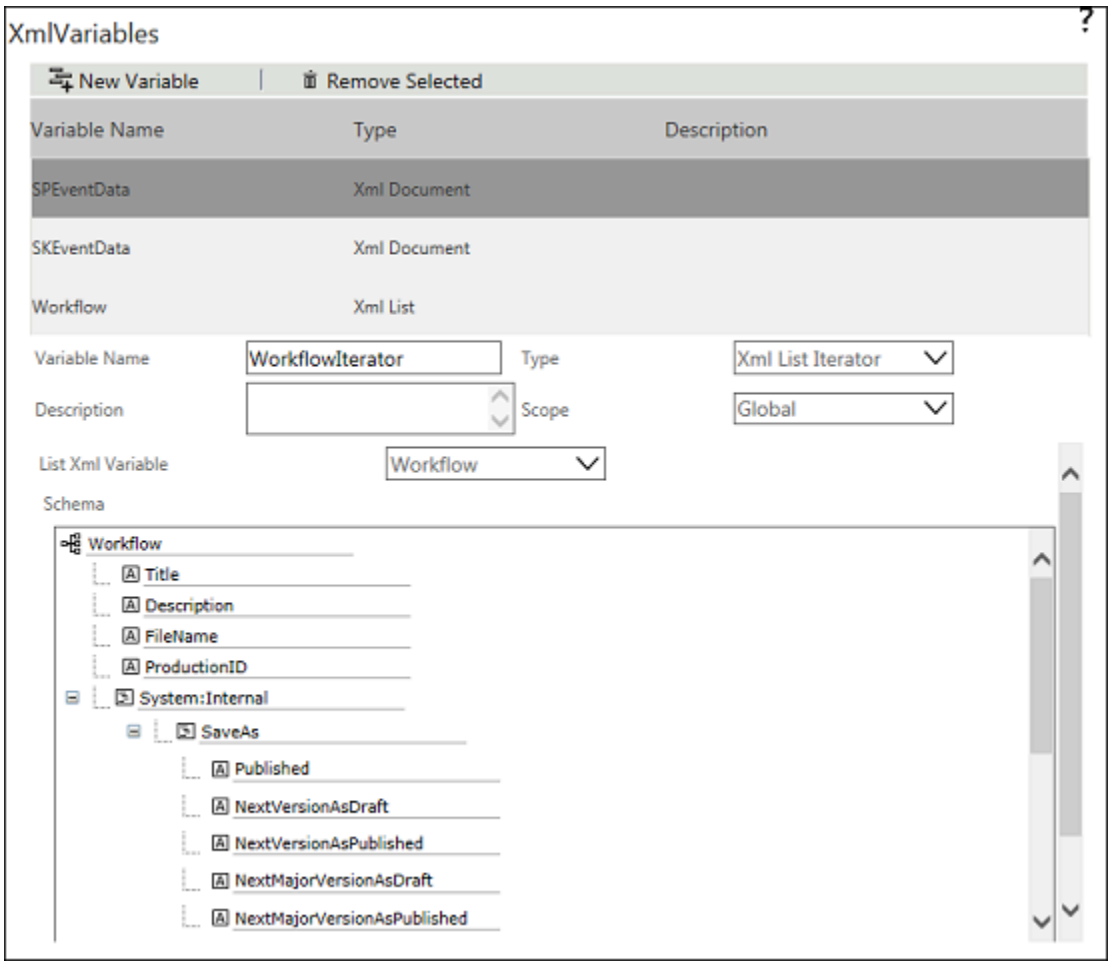


2. Click the edit icon in the XMLVariables field. The XMLVariables screen appears, with the list of the existing XML variables.
3. You have to create a XML List type XML variable before creating the XML List Iterator type XML variable. If you have already created XML List type XML variable, you can proceed to the following step. See Creating XML List type XML Variable to create XML List XML variable.
4. Click New Variable to add a new variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Enter a variable name in the Variable Name field.
6. Select XML List Iterator from the Type drop-down list.
7. Select a XML Variable from the List XML Variable drop-down list.

Note: If you need to loop through the grid items of an item while iterating through the list items, you need to select the respective **XML List Iterator** type XML variable.



8. Click **Save Variable**. A message is displayed saying 'Variable Updated'.
9. Close the window.

Create an XML Iterator of Type XML Variable for Grid Items

XML Iterator type XML variable is used to iterate though the child items (grid items) of a list item in a workflow.

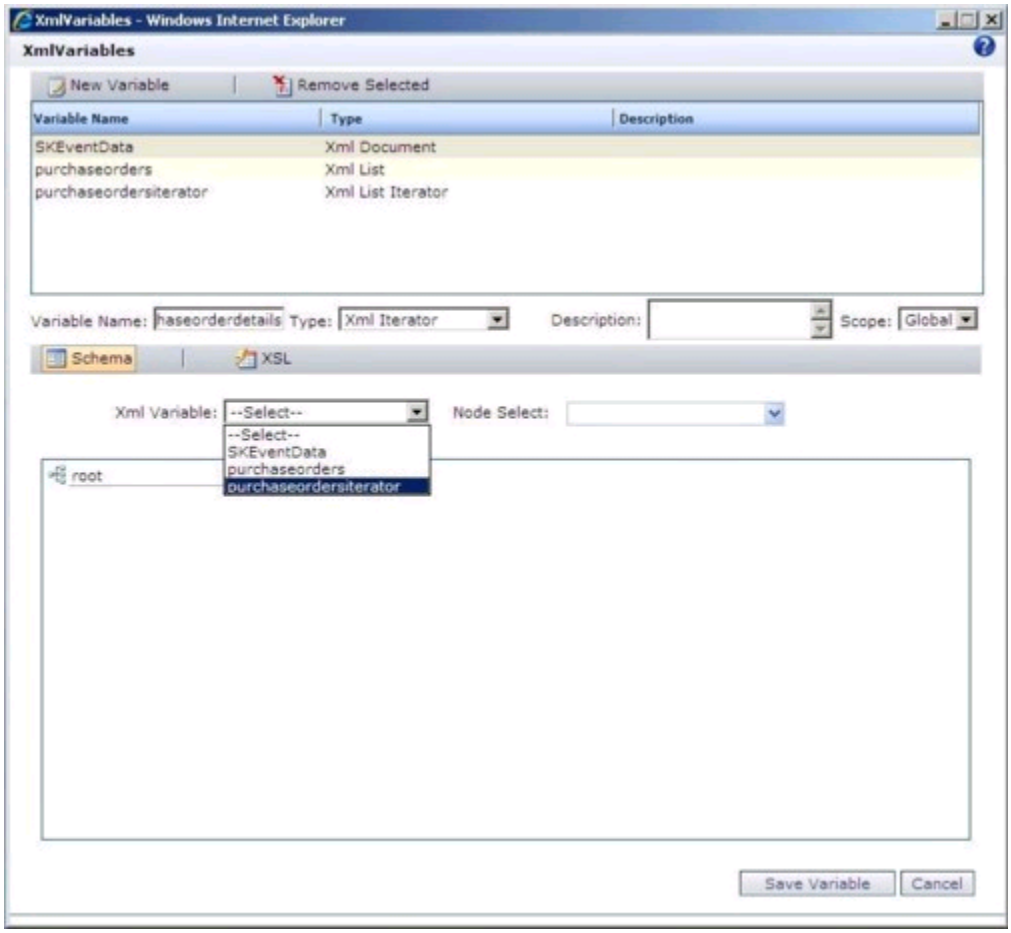
To create an XML Iterator type XML Variable for grid items

1. Right-click the **Start** Activity, and select **Activity Properties**.
2. Click the edit icon in the **XMLVariables** field. The **XMLVariables** screen appears, with the list of the existing XML variables.
3. You have to create a XML List Iterator type XML variable before creating XML Iterator type XML variable. If you have already created XML List Iterator type XML variable, you can proceed to the following step. See Creating XML List type XML Variable to create XML List XMLvariable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

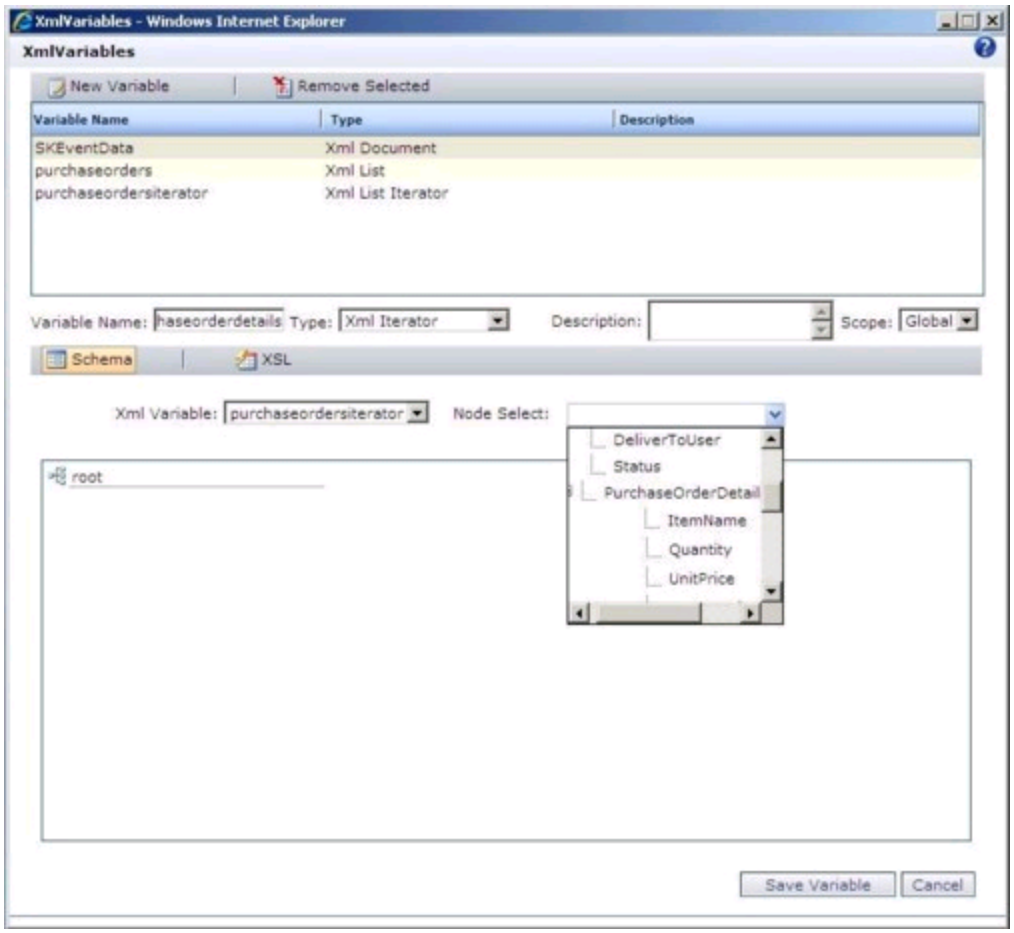
4. Click **New Variable** to add a new variable.
5. Type a variable name in the **Variable Name** field.
6. Select **XML Iterator** from the **Type** drop-down list.
7. Select a **XML Variable** from the **XML Variable** drop-down list as shown in the following screen.

Note: If you need to loop through the grid items of an item while iterating through the list items, you need to select the respective XML List Iterator type XML variable.



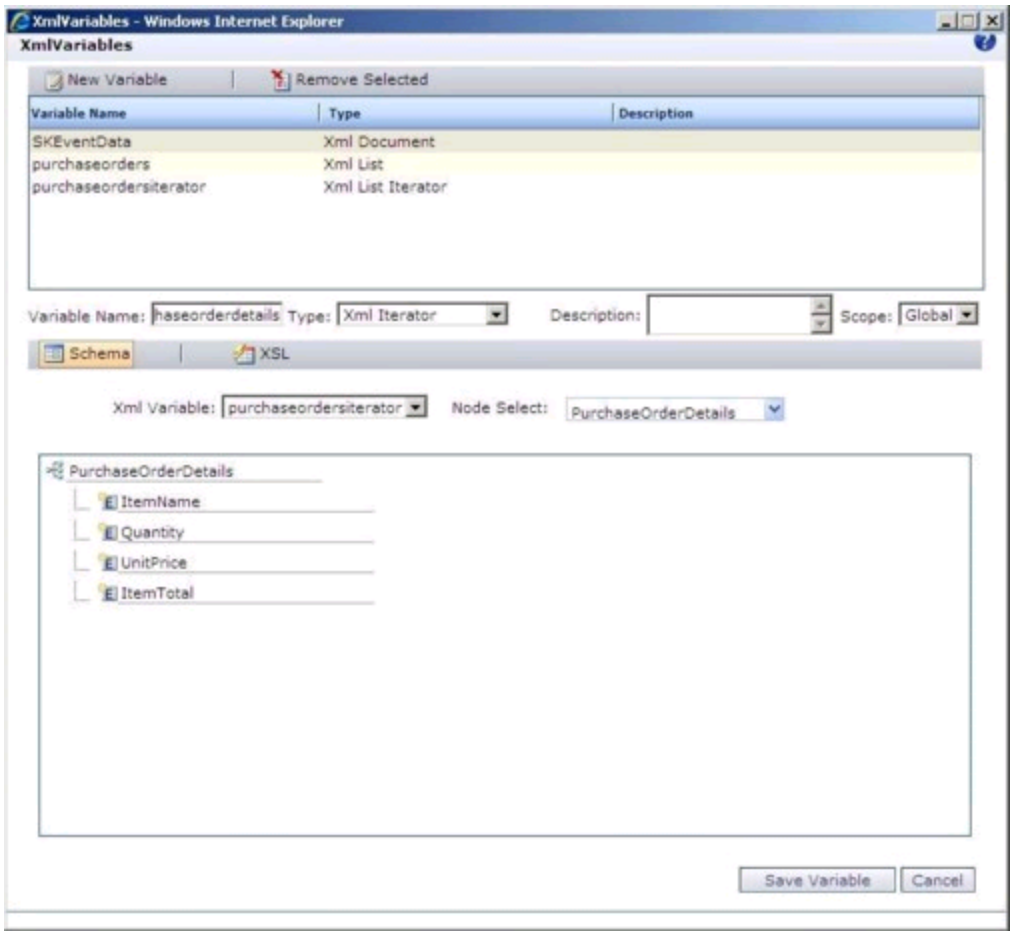
8. Select a repeating node from the **Node Select** drop-down list. Select the XML node which is mapped for the grid in the list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



9. When you select a node, the following screen is displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



10. Click **Save Variable**. The message 'Variable Updated' is displayed. Close the window.

Create Workflow for a List in a Single Step

The List Workflow option in the Workflows ribbon bar helps to create and associate a workflow to a list in a single step.

To create a Workflow for a list

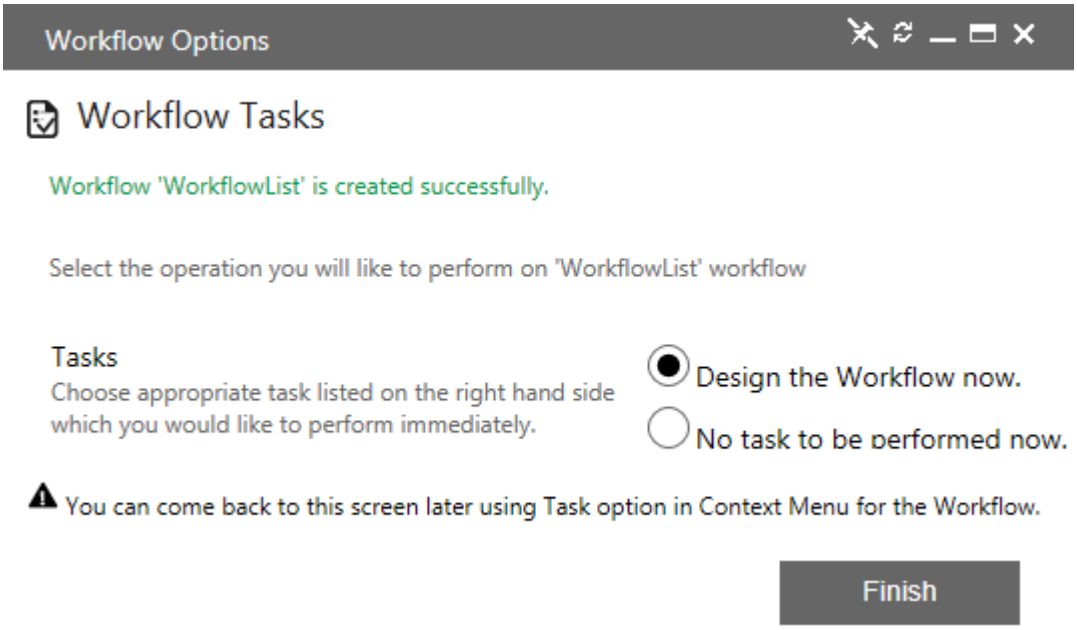
1. Log on to the **Enterprise Console** and create a list.
2. Click the **Enterprise Console** menu on the top and select **Workflows** from the drop-down list. The Workflows screen is displayed.
3. Click **New** and then click **List** on the ribbon bar. The **New List Workflow** screen appears.
   Enter the required details.

Field	Description
New List Workflow	
Title	Enter a title of workflow for the List in this field.
Description	Enter a description for the workflow in this field.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Associate to List Event - This can also be done using the 'Associate' icon in the AVEVA Work Tasks Workflow List Menu.	
List	Select a list ID. The **List Explorer** screen opens. Select the list name from the drop-down list and click **Submit**.
Event	Select an event from this drop-down list. The available options are **OnCreate**, **OnDelete**, **OnPublish**, **OnUnPublish**, and **OnVersionChange**.

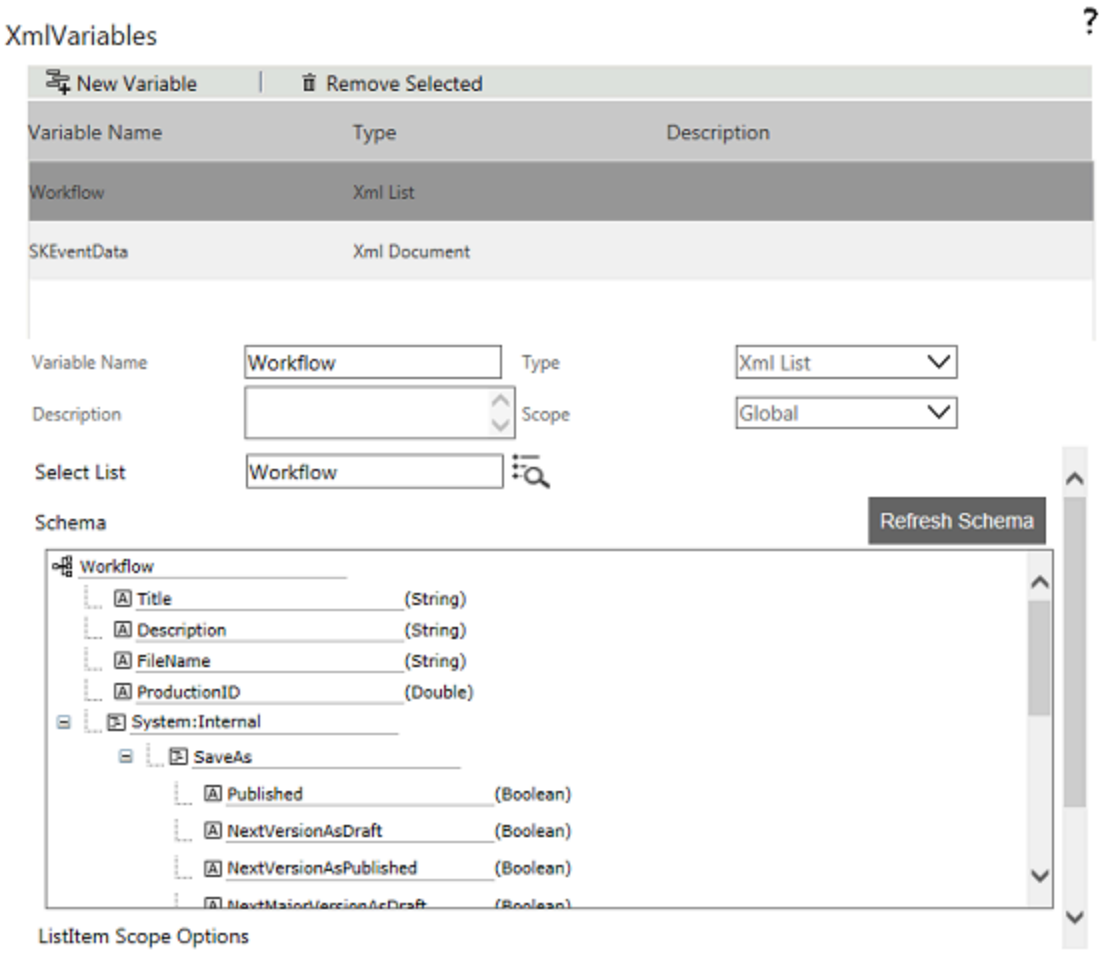
4. Click **Save & Continue**. The following screen is displayed.



5. Select the required task, and then click **Finish**. The **Process Designer** screen opens.
6. Following behavior is observed while using List Workflow.

- Workflow Creation
  When you click **Finish**, a workflow is created for the selected list. You can open this workflow either by selecting Design the Workflow now from tasks or using **Design Workflow** in the context menu of the workflow item.
- List XML Variable
  An XML variable of XML List type is also created by default and the List name is used as the Variable name. To view the XML Variable, go to the workflow designer screen. Right-click the **Start** activity, select **Activity Properties** from the options and then select the **XML Variables** icon from the **Activity Properties**.

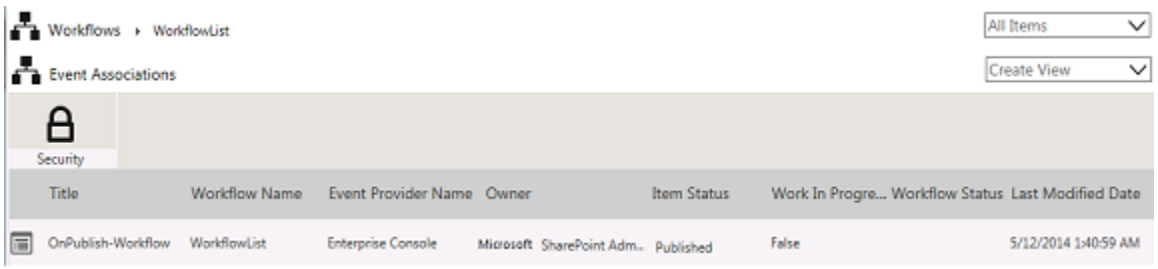
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



You can notice that, the variable is created with scope as **Current**, which means, this variable is responsible to give the information of the list item which initiates the workflow. The XML variables displayed in tree-view are the XML nodes that are mapped to the controls while creating the list.

System nodes are also created automatically for internal use. This helps you to save the list item in to another version as published or draft and also helps to save the item as folder for the list item.

- List Event Association
  This workflow is associated to the list event that you have selected at the time of execution. To view the associations, select the workflow, right click and select **View Associations** from the options.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Understand XML Node Mapping

This section explains the importance of XML Nodes while designing a List or a Form. The XML node allows binding a SQL Database to a List and the List to be mapped to a Form.

The following are the XML Nodes with an example of a library database.

- Database contains the table with columns such as, Id, Title, Author, Application.
- In the List, the Author control is bound to the DB column Author.
- In the Form, the Author Name control is mapped to the Author control in the list with the XML Node named Authorname.

A table column is mapped to a List control using the List control property 'Bound to DB Column' and the List control is mapped to Form control using 'Mapped XML Node'.

When you input data using a Form and the data is stored in the corresponding List, the List then updates the database table.

Note: When you design a List or Form, the Mapped XML Node allows mapping of a List to a Form. When you add a control for the first time, the XML Node is dynamically altered to reflect the Tag Name. Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists, and Workflows.

XML List of Type XML Variable Scenarios

Using XML List XMLVariables, you can add, update, save, and perform other functions of list items.

Add New List Item

AVEVA Work Tasks allows you to add a new list item to the list using XML List XML variables. Update Variable activity can be used to add/update an item to the list. To add an item you need to select the scope as New from Choose the List Item to Work on options. You can also use List Navigator Activity to set the scope. With the help of a scenario, we will explain this function.

Assume that you have created a list called Purchase Orders with the following fields.

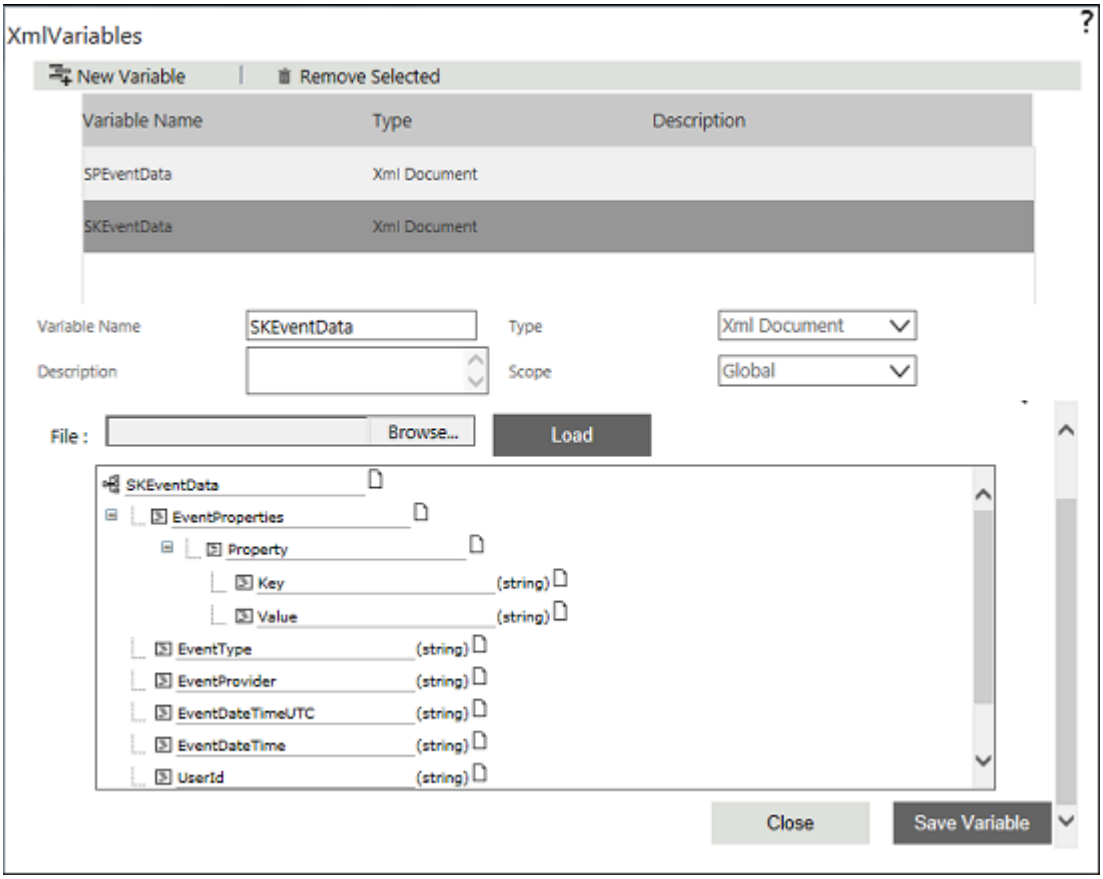
- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price
  - Item Total
- Total Purchase amount

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Now you want to add items to this list.

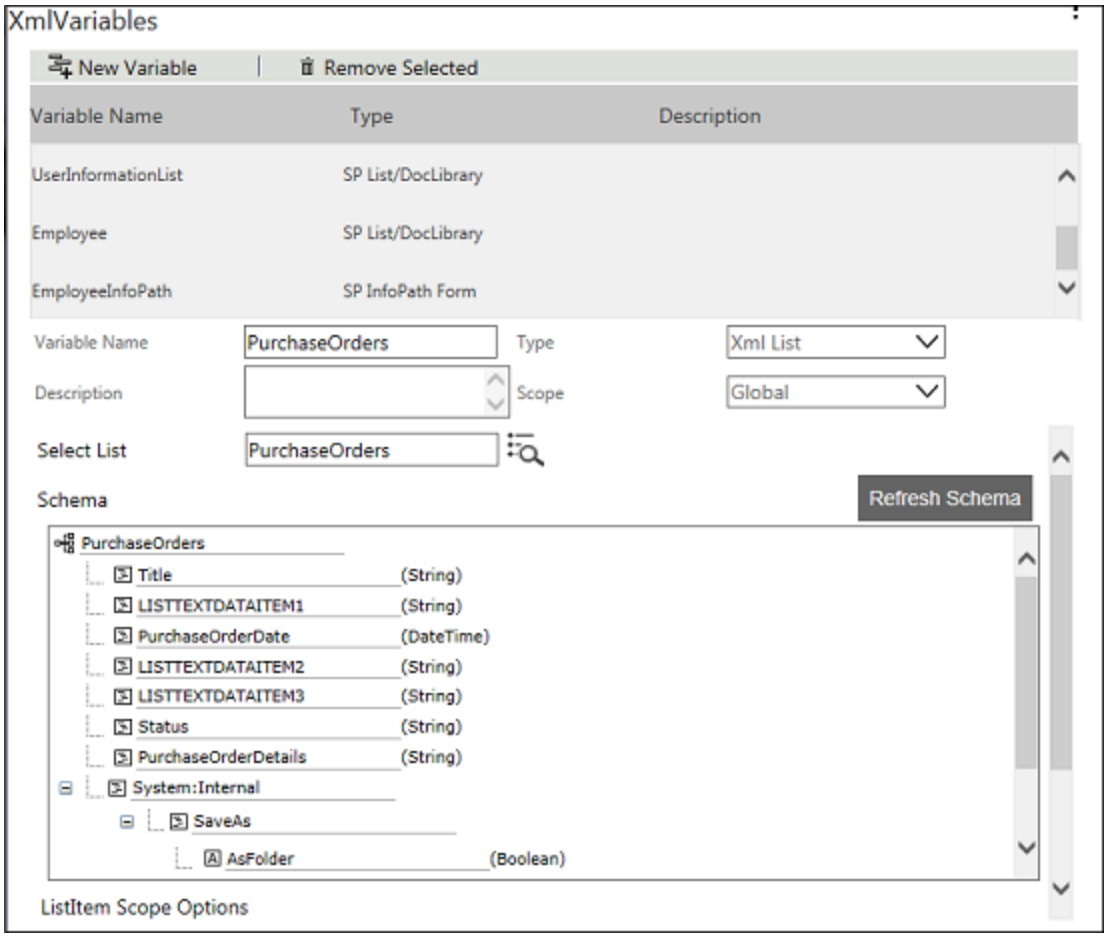
To add a new List Item

1. Login to the **Enterprise Console** and launch the **Workflow Designer** page.
2. You need to create an XML List type XML variable for the Purchase Orders list. If the variable is not created, follow the below mentioned procedure to create.
3. Click the edit icon in the **XMLVariables** property from the **Start** Activity Properties. The following screen is displayed.

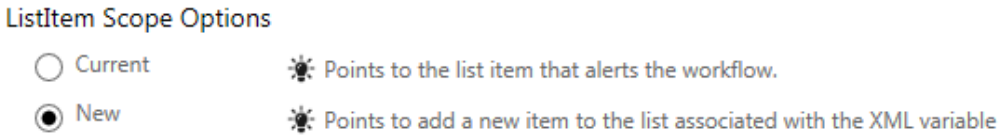


4. Click **New Variable** to add a new variable.
5. Type the variable name as 'purchaseorders'.
6. Select **XML List** from the **Type** drop-down list.
7. Type the List name in the **Select List** field or click the icon to select the list. In this case, select Purchase Orders list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

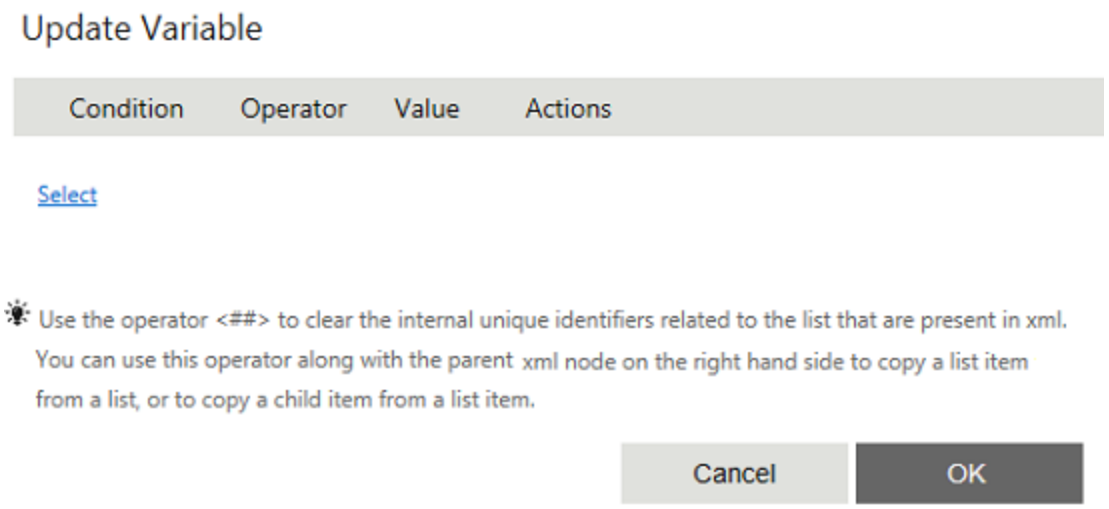


8. Select **New** as the scope from the ListItem Scope Options to point to a new item in the Purchase Orders list.

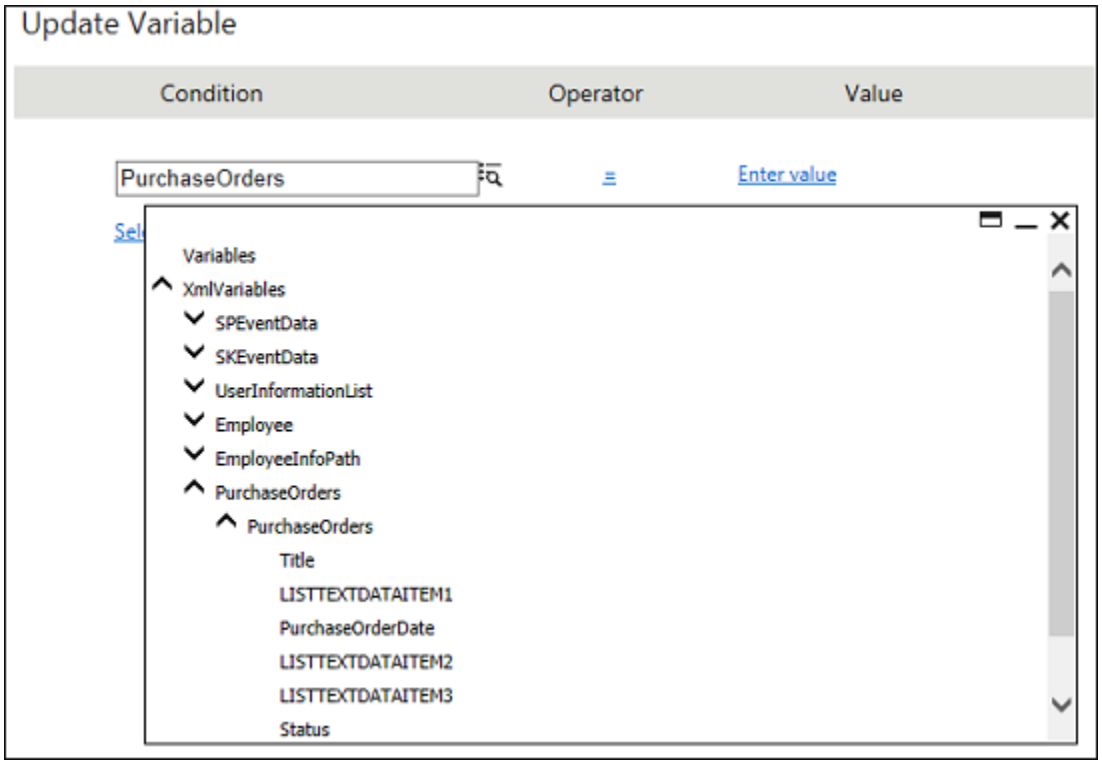


9. Now you can see the XML nodes in a tree-view that are mapped to the controls while creating the list.
10. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window.
    Note: If you set the scope as Current while creating the XML List type XML variable, you can change it in to New using List Navigator activity.
11. Drag and drop **Update Variable** activity to add an item to the list.
12. Click the edit icon in the **Set Variable** property. The following screen is displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

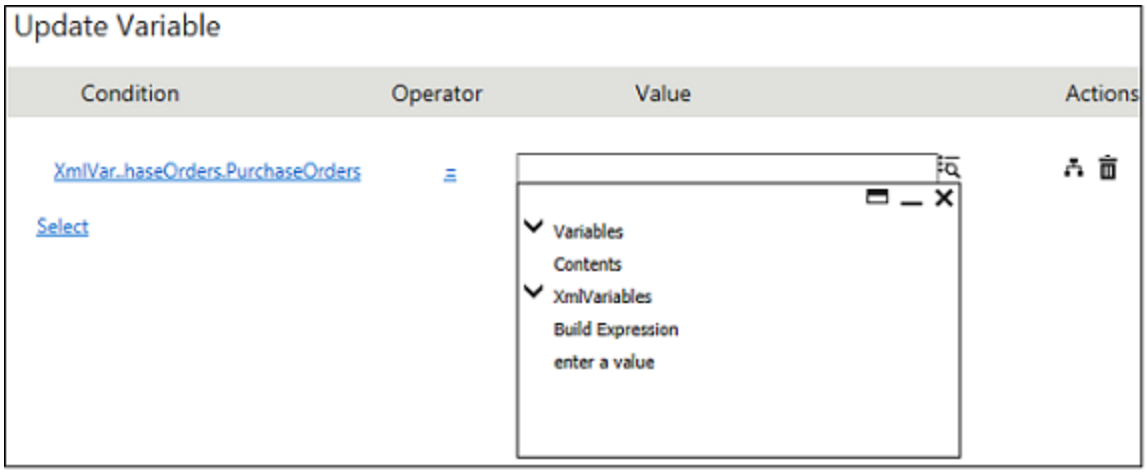


13. Click the **Select** link and select the respective field of the XML List type XML Variable created for the list (in this case purchaseorders). The fields displayed in the tree-view represent the XML nodes of the controls in the list.

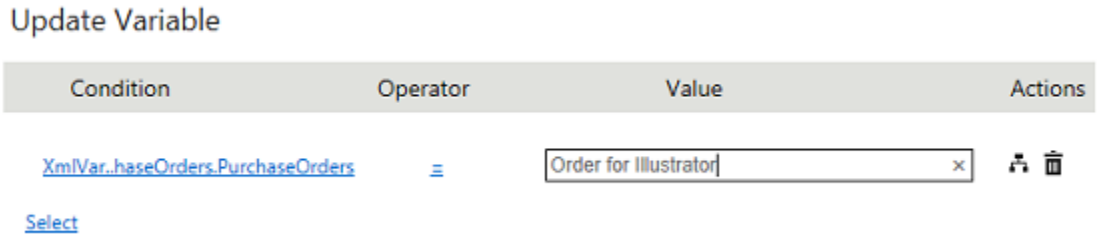


14. Select a field for which you want to set the value, from the tree-view and enter the value for the field in the right-hand side text box, or you can use the icon to set the value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



15. If you want to set the value, select **enter a value** from the tree-view, and then enter the value. Also you can select variables or contents or any XML Variable nodes directly by selecting the respective node from the tree-view. If you want to combine multiple node values, you can build an expression by selecting **Build Expression**.



16. In the same way, you can set the value for all other fields in the list for the new item. Make sure to specify the values for all mandatory fields of the list.
17. After setting the value for the entire field, click **OK**. When you execute this workflow, a new item will get added based on the value that you have set.

Update the List Item that Alerted the Workflow

AVEVA Work Tasks allows you to update the list item that alerted the workflow using XML List Type XML Variable. Update Variable activity can be used to add/update an item to the list. To update an item, you need to select the scope as Current from Choose the List Item to Work on options. You can also use List Navigator Activity to set the scope. With the help of a scenario, we will explain this function.

Assume that you have created a list called Purchase Orders with the following fields and alerted the workflow.

- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

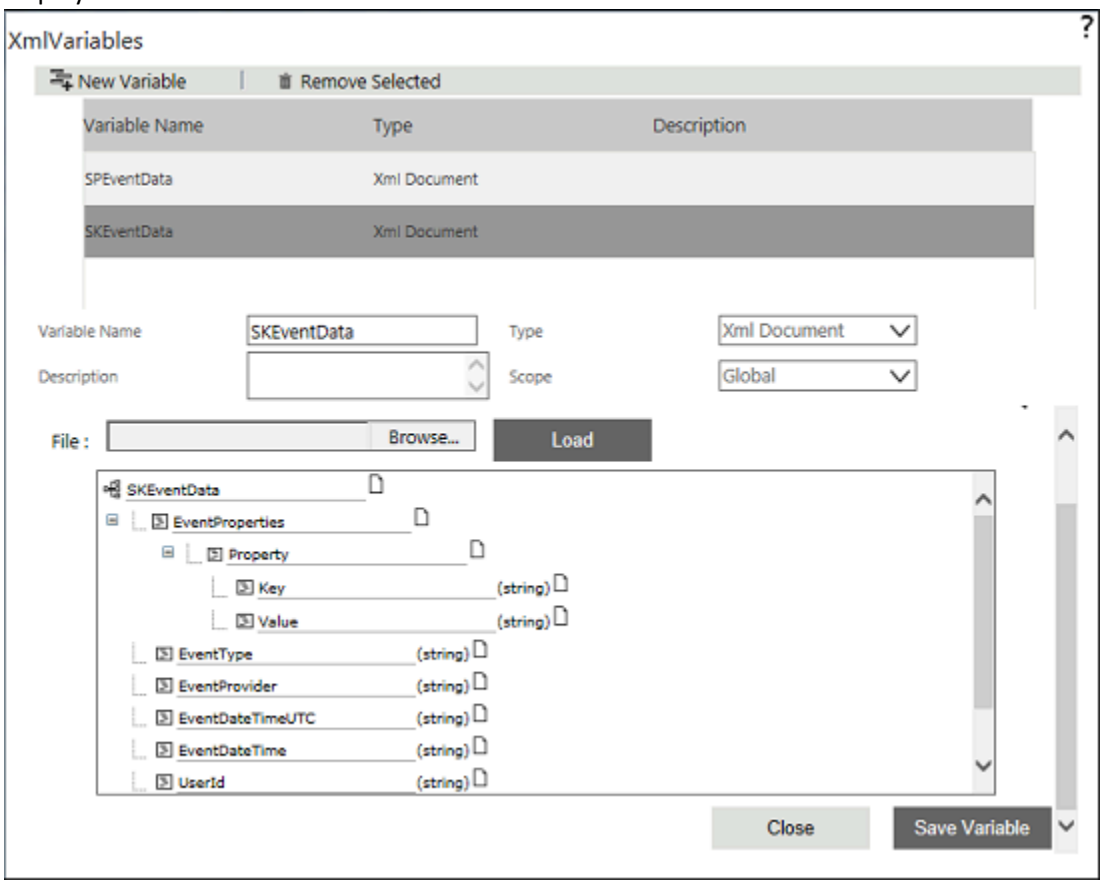
- Quantity
- Unit Price
- Item Total

- Total Purchase amount

Now you want to update an item in the same list. Follow the procedure given below.

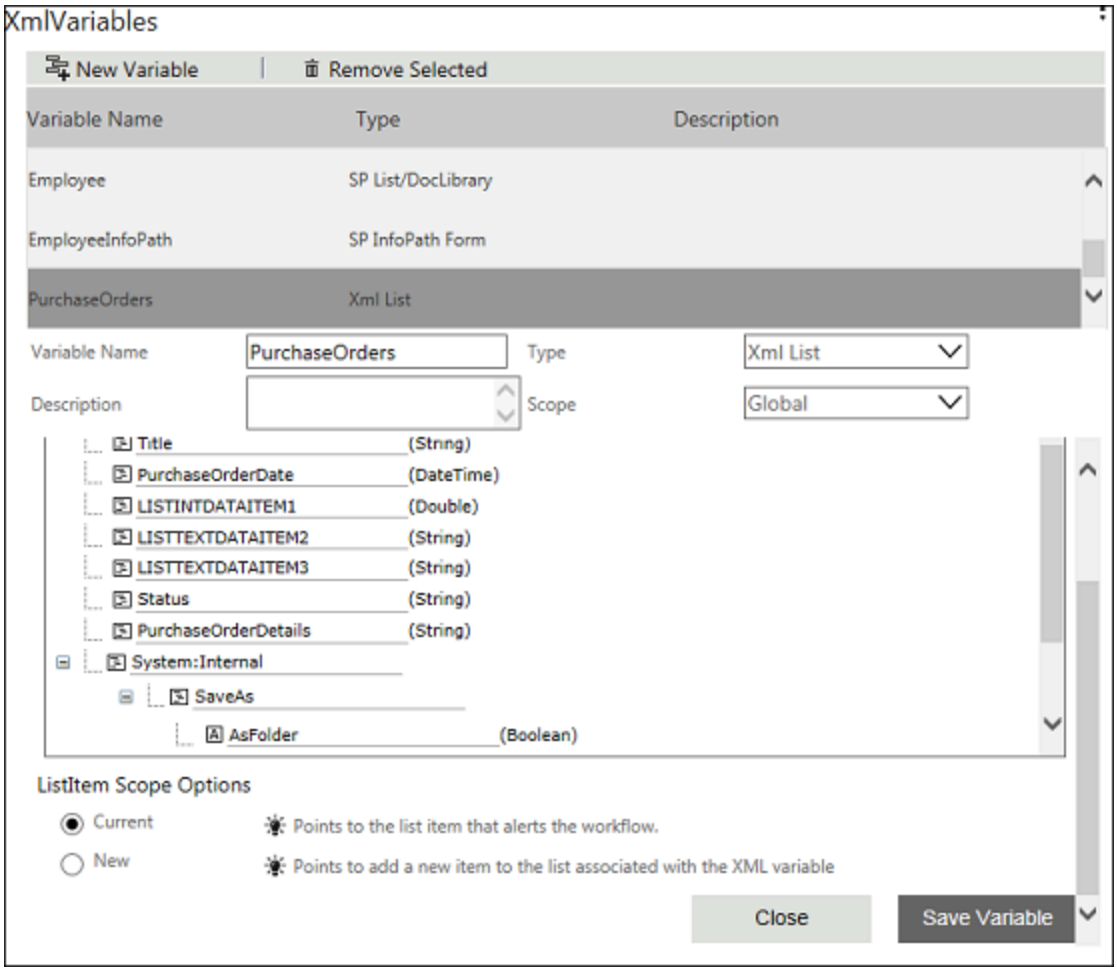
To update the list item that alerted the workflow

1. Login to the **Enterprise Console** and go to the Workflow Designer page.
2. You need to create an **XML List** type XML Variable for the Purchase Orders list. If the variable is not created, follow the below mentioned procedure to create.
3. Click the edit icon in the **XMLVariables** property from the **Start** Activity Properties. The following screen is displayed.

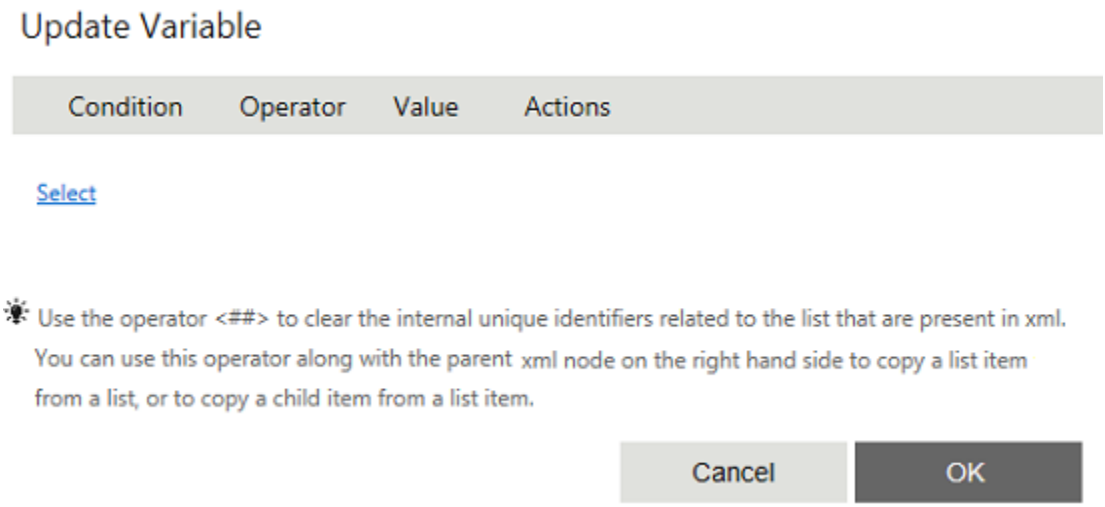


4. Click **New Variable** to add a new variable.
5. Type the variable name as 'purchaseorders'.
6. Select type as **XML List** from **Type** the drop-down list.
7. Type the List name in the **Select List** field or click the icon to select the list. In this case, select 'Purchase Orders' list.
8. Select Current as the scope from the **ListItem Scope Option** to point to an item in the Purchase Orders list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

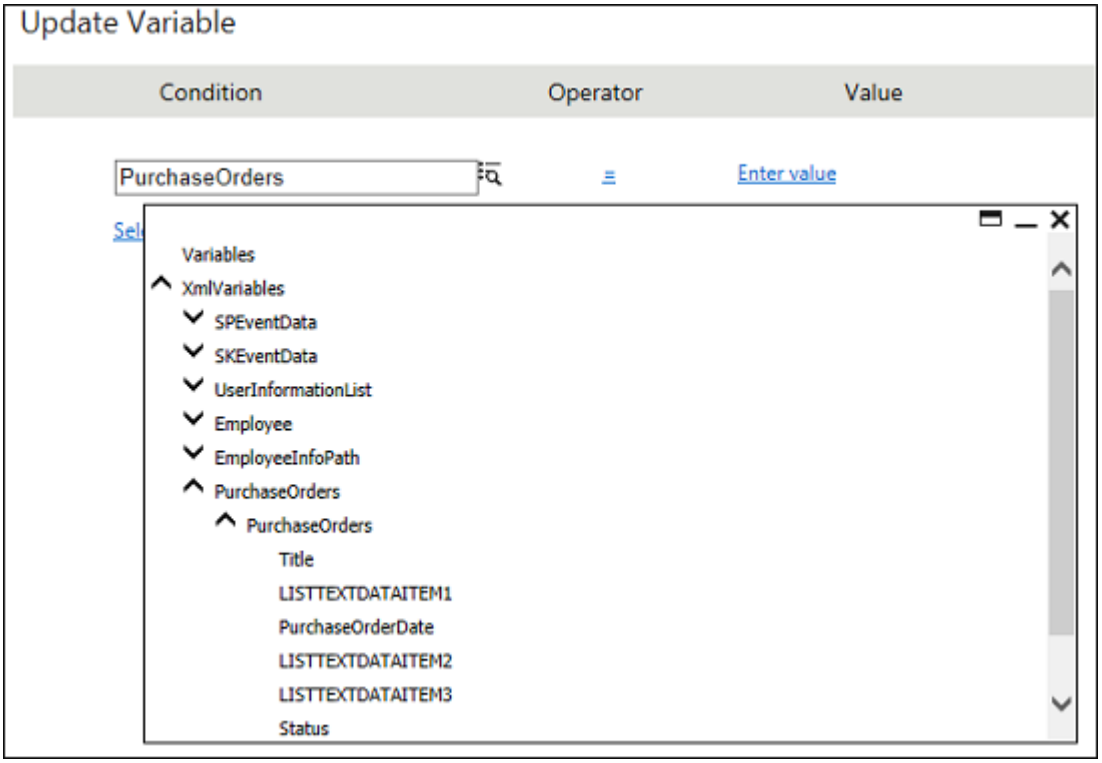


9. Now you can see the XML nodes in a tree-view that are mapped to the controls while creating the list.
10. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window.
    Note: If you set the scope as **New** while creating the **XML List** type XML variable, you can change it to Current using List Navigator activity.
11. Drag and drop **Update Variable** activity to update an item in the list.
12. Click the edit icon in the **Set Variables** property. The following screen is displayed.

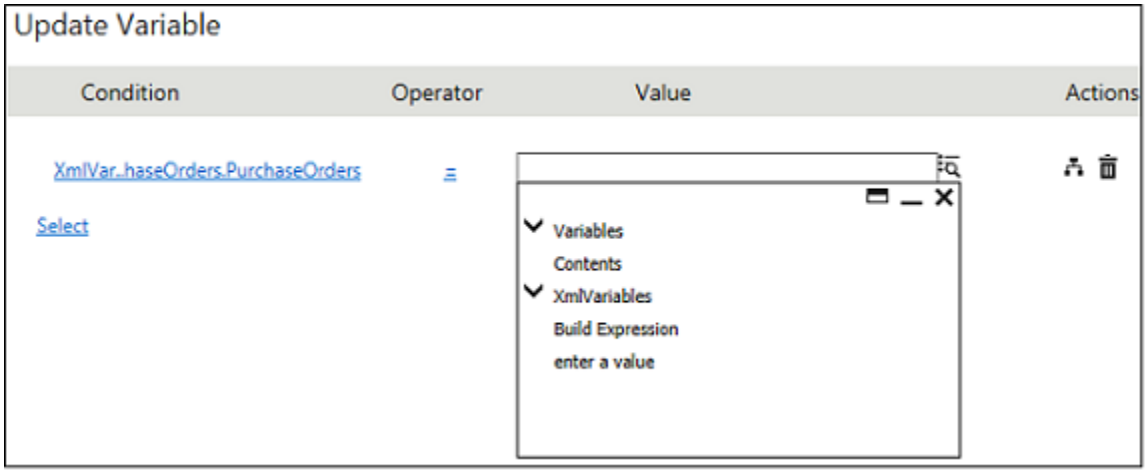


© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

13. Click the Select link and select the respective field of the XML List type XML Variable created for the list (in this case 'purchaseorders'). The fields displayed in the tree-view represent the XML nodes of the controls in the list.

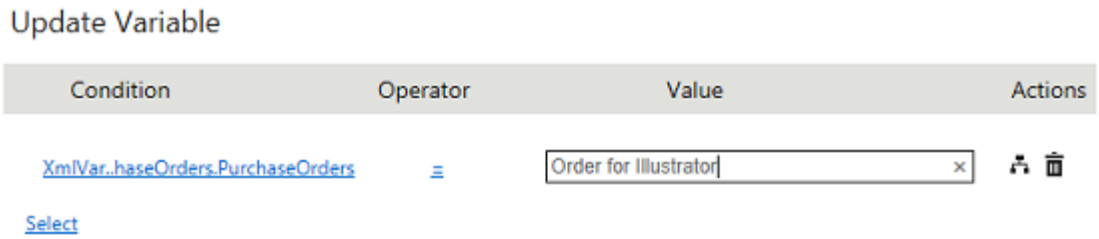


14. Select a field for which you want set the value from the tree menu. Enter the value for the field in the right-hand side text box or click the icon to set the value.



15. If you want to set the value, select 'enter a value' from the tree-view and then enter the value. Also, you can select variables or contents or any XML variables nodes directly by selecting the respective node from the tree-view. If you want to combine multiple node values, you can build an expression by selecting Build Expression.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



16. The same way, you can set the value for other fields that you want to update. In this case, you are only updating the field 'Title' which has the value 'Order for Illustrator'.
17. Click **OK**. When you execute this workflow, the item get updated based on the value that you have set.

Save the List Item as Different Version as Draft/ Published Using Xml Variable

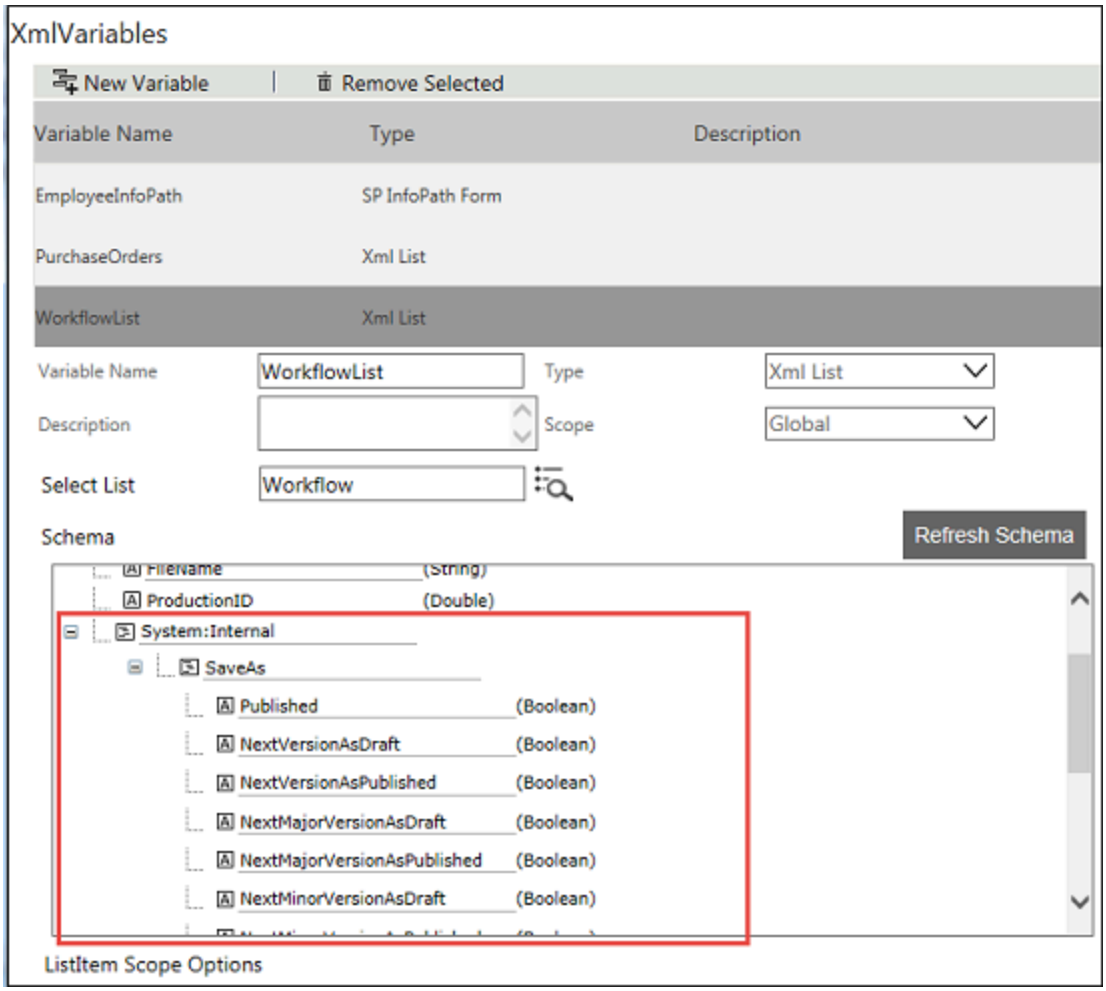
There are two kinds of list, version enabled and version not enabled. If your list is version enabled, you can save the item as a different version as published or draft using System Internal variables. This can be done at the time of setting the value for activity.

Assume that you are creating a workflow and it is attached to the out of box list workflow which is version enabled.

To save the list item as different version as draft/published using Xml Variable

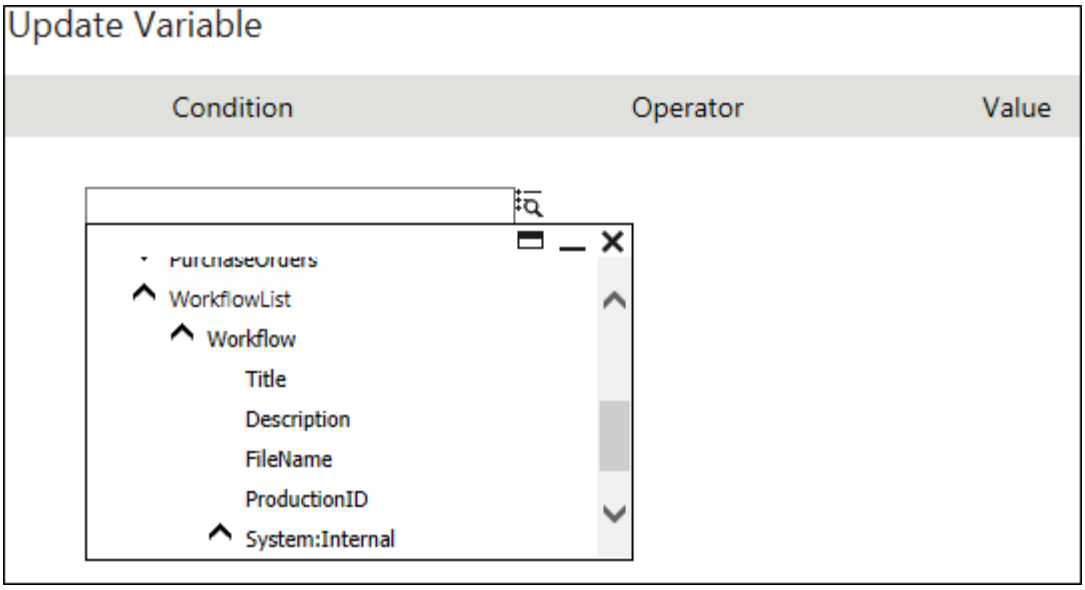
1. Right-click the **Start** activity and select **Activity Properties** from the options.
2. Click the edit icon in the field. The following screen is displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

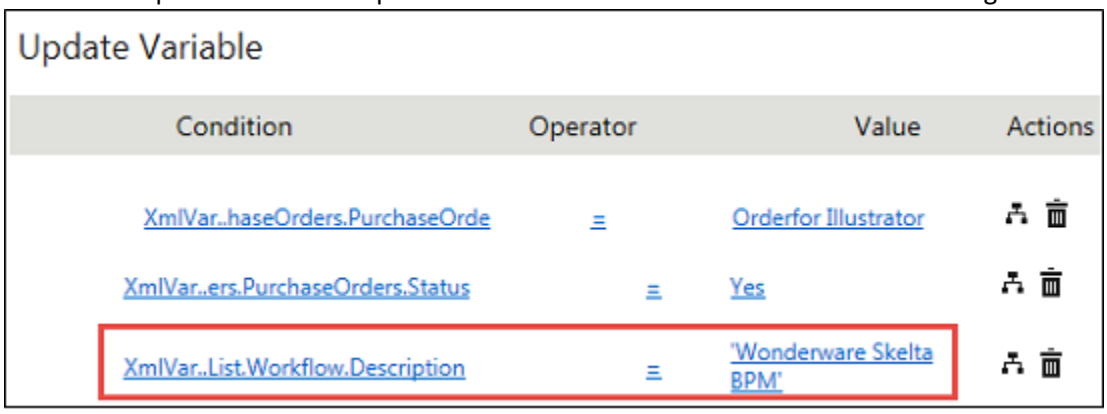


3. For the version enabled list, you can see the list of schema in a tree-view under System:Internal. Nodes include Save As > NextVersionAsDraft, NextVersionAsPublished, and so on. You can use these nodes to save an item to another version as draft or published.
4. Use **List Query** and point to an item that you want to update and save as different version. See Setting up the List Query Scope.
5. Drag and drop **Update Variable** activity. Click the edit icon in the **Set Variables** field.
6. Click the **Select** link and then click the drop-down list that appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

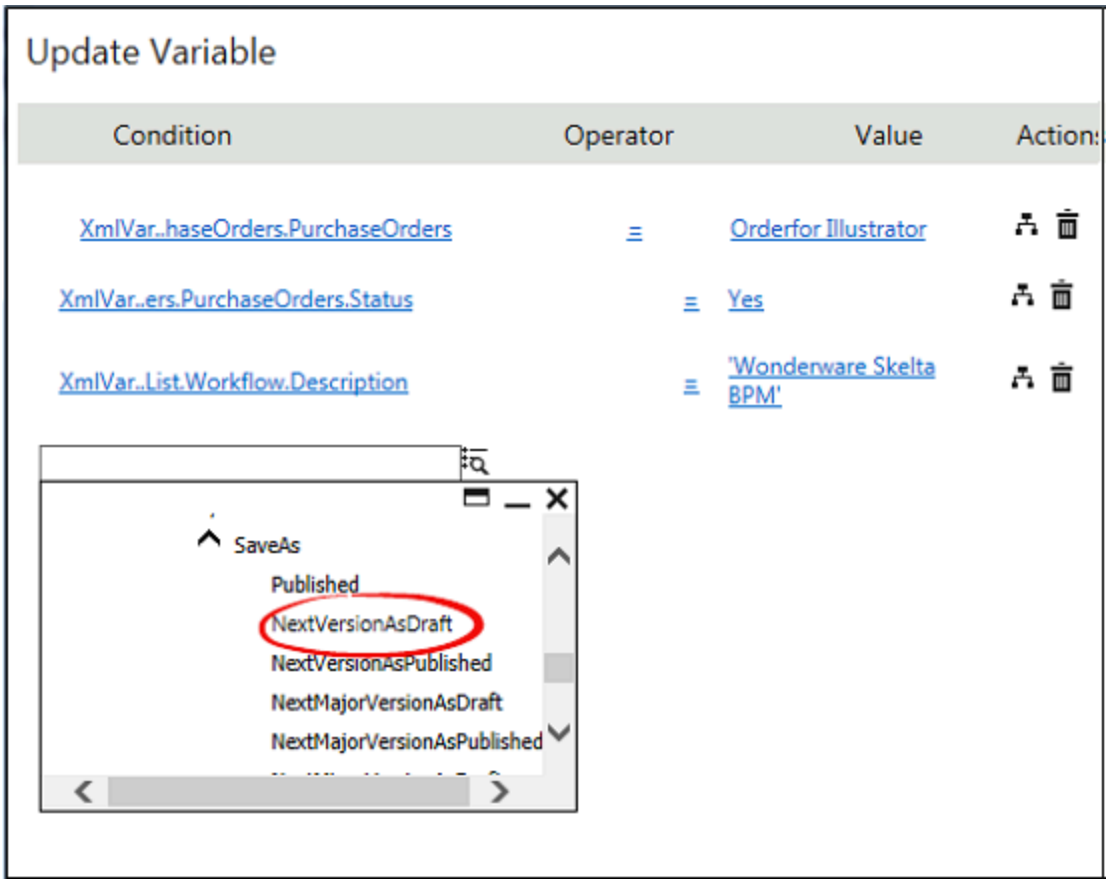


7. Assume that you want to update the item **Description** having the value 'Workflow Management'. Select Description from the drop-down list and enter the value as 'Workflow Management'.

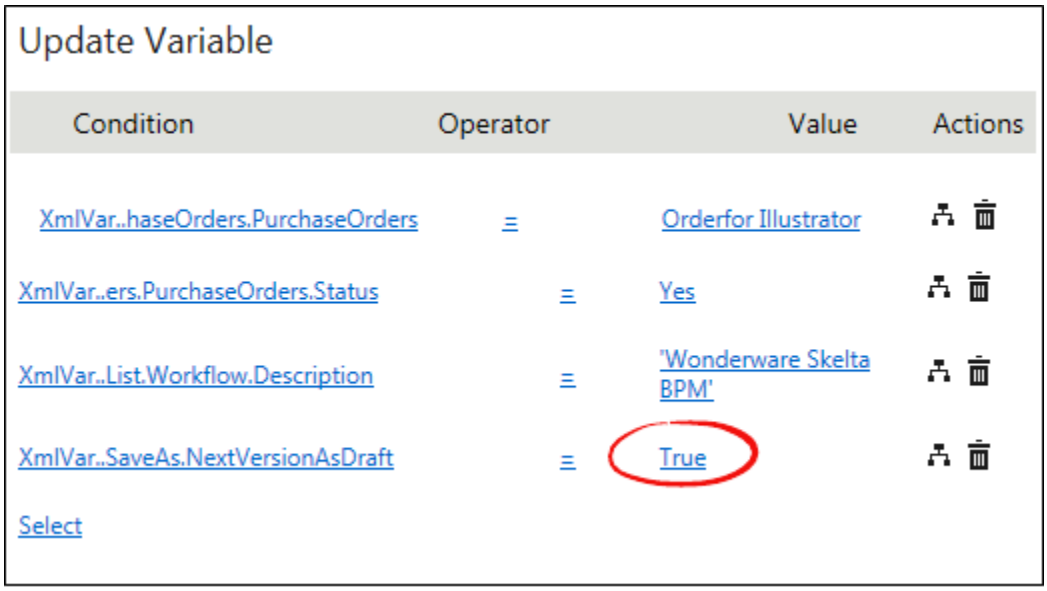


8. If you want to save this item as **Next Version > Draft**, click the drop-down.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



9. Select **NextVersionAsDraft** from the SaveAs options under **System:Internal**.
10. Enter the value as 'True'.



11. Click **OK**. This will update the item **Title** and also save the item as next version as draft.
12. In the same way, you can save any item to different versions like Next Major Version as Published/Draft, Next Minor Version as Published/Draft while updating the item.

Note: If you select multiple SaveAs conditions and set the value as **True**, the last condition will be taken into

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

account and the item will be saved based on the last condition. Also for the list that is not version enabled, the system will automatically save the item as published.

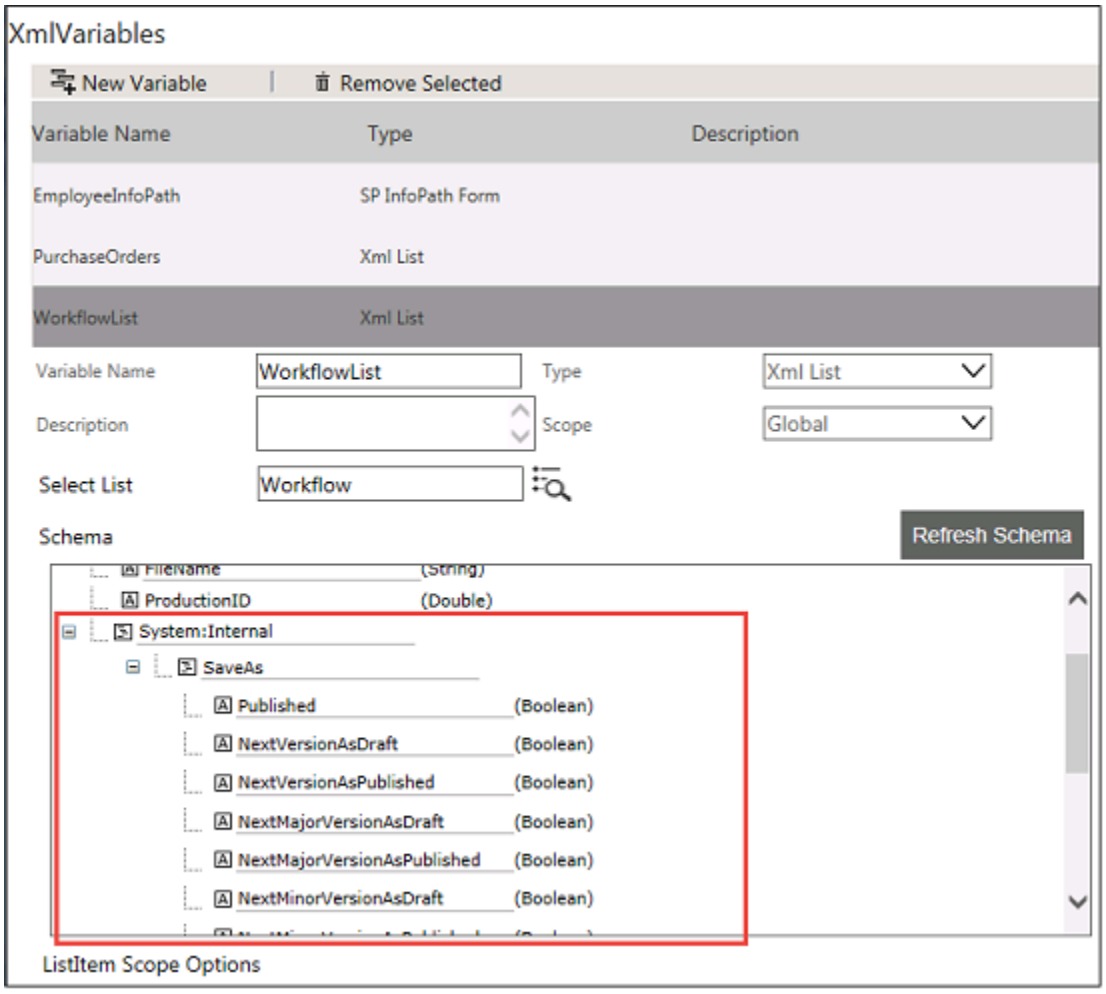
Create New Folder Using Xml Variable

If your list is version enabled, you can save the item into a folder using System Internal variables. This can be done at the time of setting the value for **Update Variable** activity.

Assume that you need to create a folder in Workflows list which is version enabled.

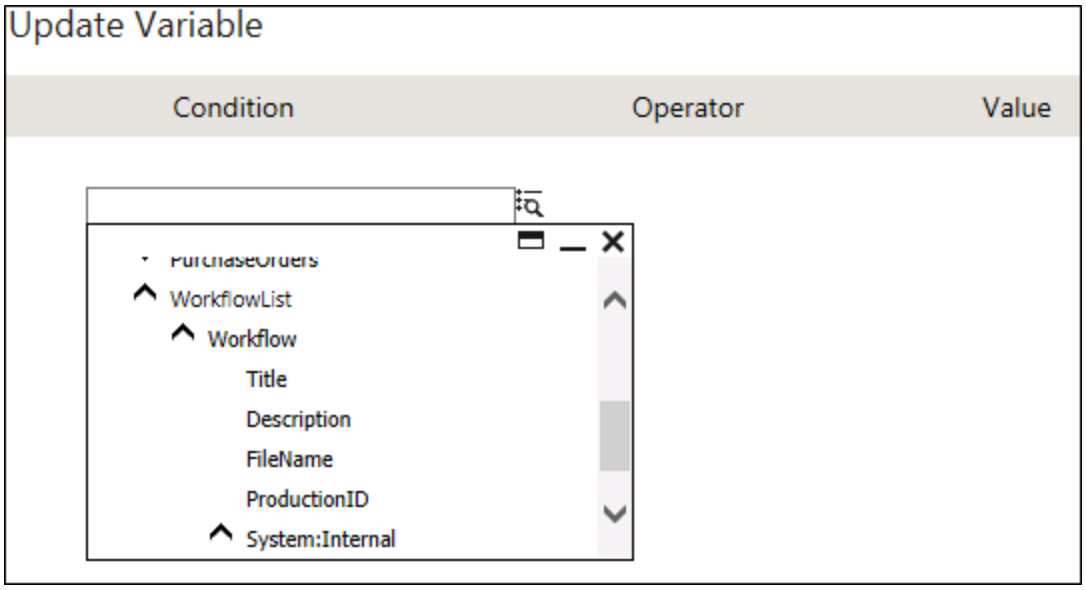
To create a new folder using XML Variable

1. Right-click the Start activity and select **Activity Properties** from the options.
2. Click the edit icon in the **XMLVariable**s field. The following screen is displayed.

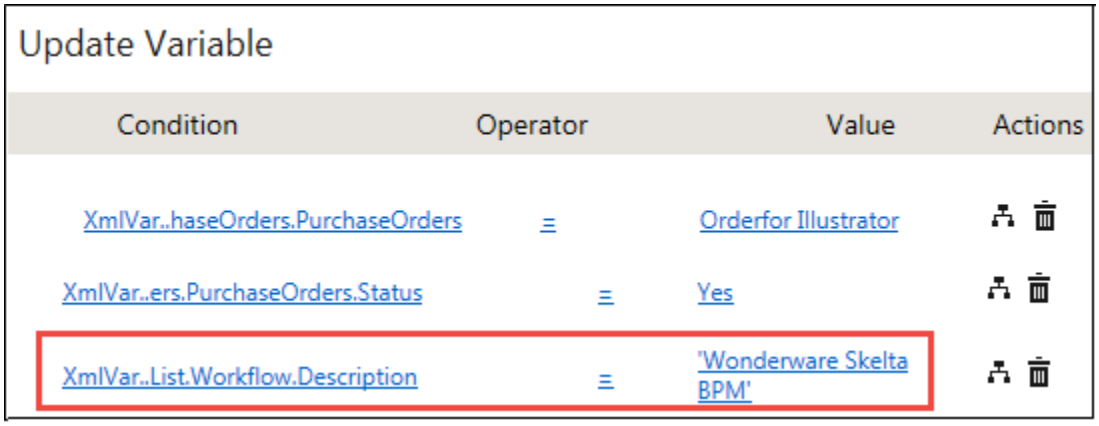


3. For the version enabled list, you can see the list of schema in a tree-view under **System:Internal**. Nodes include Save As NextVersionAsDraft, NextVersionAsPublished, **AsFolder** and so on.
4. Drag and drop **Update Variable** activity. Click the edit icon in the Set Variables field.
5. Click the **Select** link and then click the drop-down that appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

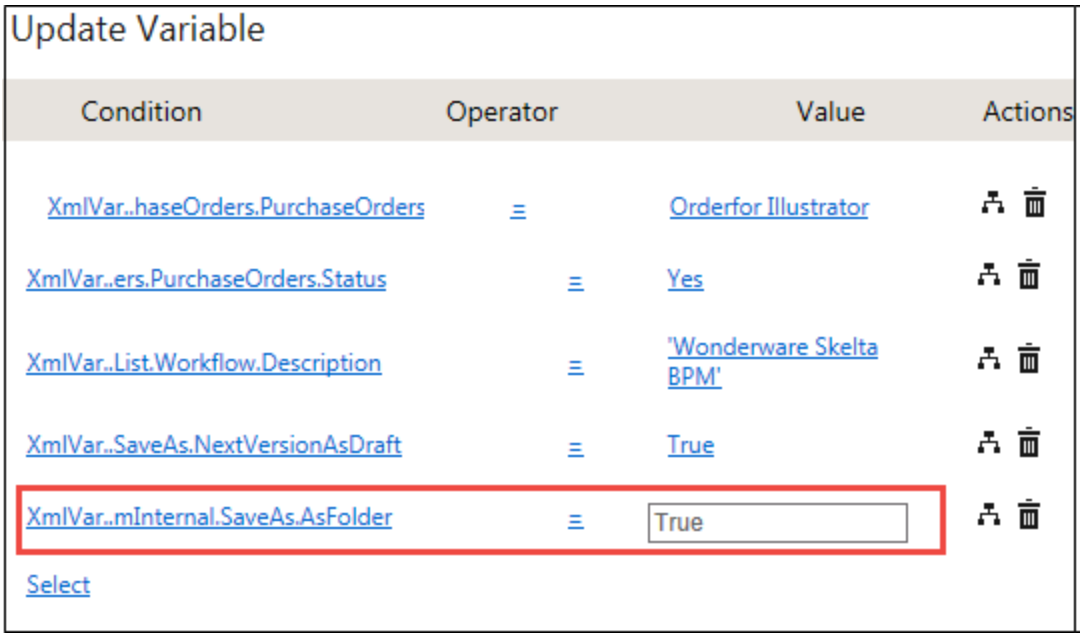


6. Assume that you want to save the item **Description** as a folder with the folder name as 'Workflow Management'. So, select **Description** from the tree-view and enter the value as 'Workflow Management' in the right-hand side text box.



7. To save this item as folder, click the drop-down.
8. Select **AsFolder** from the **SaveAs** options under System:Internal.
9. Then enter the value as 'True'

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



10. Click **OK**. This will update the item **Description** and also will create a folder with the name 'Manager'.

Note: Folder names should not have special characters like such as, :, ", \, \, " ", ', <, >, ;, |, *, /, {, }, and ?.

Add New List Item with Grid Items

AVEVA Work Tasks allows you to add list items with grid items to any list using XML variable. Update Variable activity can be used to perform this task. An XML Iterator type XML variable should be created to add the grid items (child items) and the XML Navigator activity can be used to point to the grid.

This can be explained through the scenario mentioned below.

A list, Purchase Orders, has been created with the following fields.

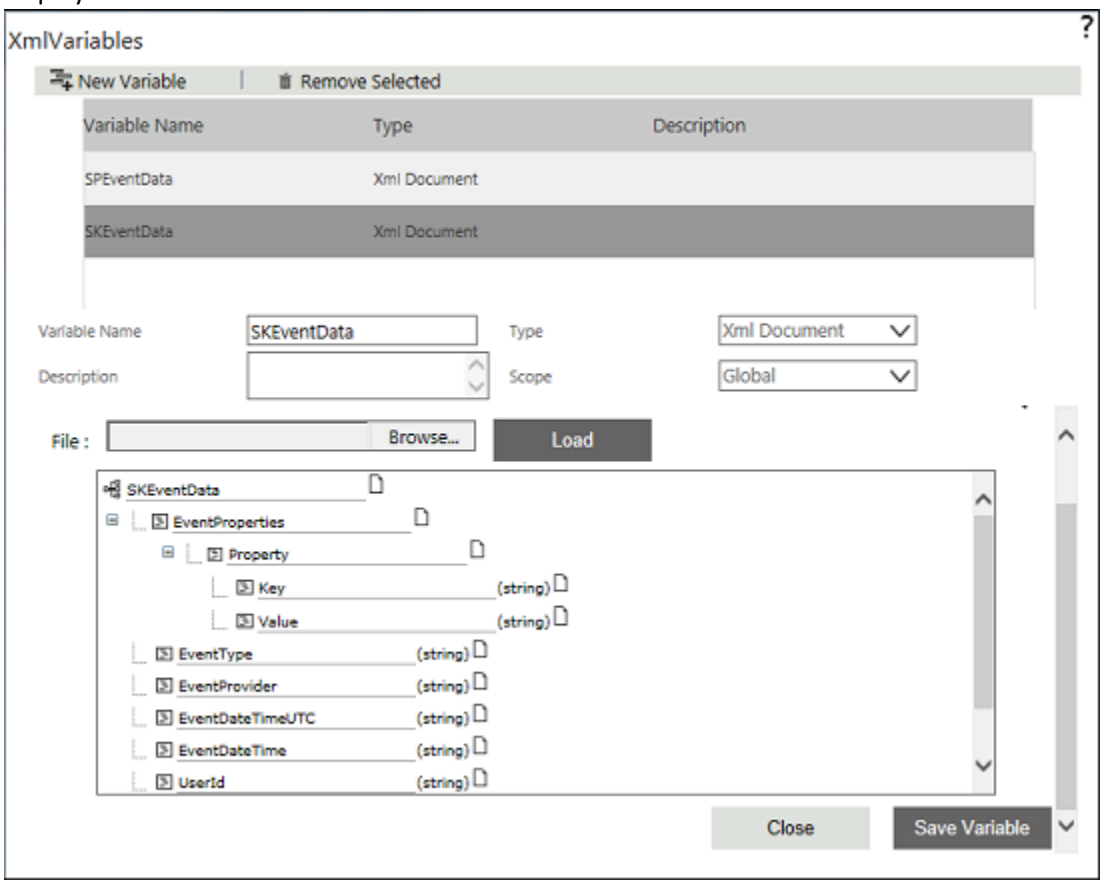
- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price
  - Item Total
- Total Purchase amount

From the workflow, you have to add an item along with the child item to the Purchase Orders list.

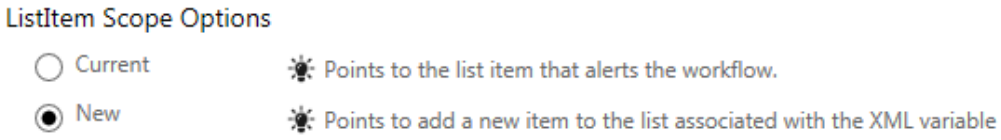
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To add a new list item with grid items

1. Log on to the **Enterprise Console** and launch the **Workflow Designer** page. You need to create an XML List type XML variable for the Purchase Orders list. If the variable is not created, follow the below mentioned procedure to create it.
2. Click the edit icon in the **XML Variables** property from the **Start** Activity Properties. The following screen is displayed.



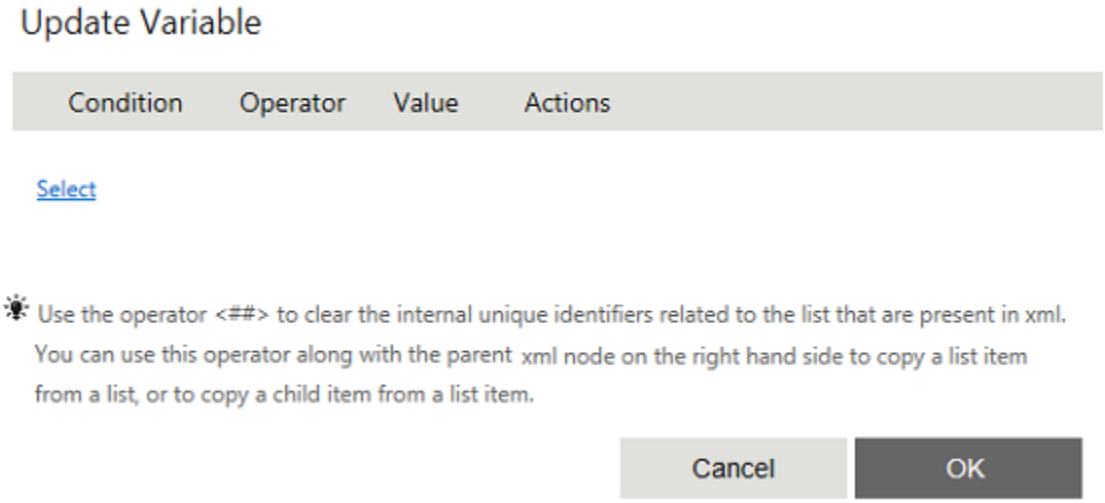
3. Click **New Variable** to add a new variable.
4. Type the variable name as 'purchaseorders'.
5. Select **XML List** from the **Type** drop-down.
6. Type the List name in the **Select List** field or click the to select the list. In this case select Purchase Orders list.
7. Select **New** as the scope from the **ListItem Scope Options** to point to a new item in the Purchase Orders list.



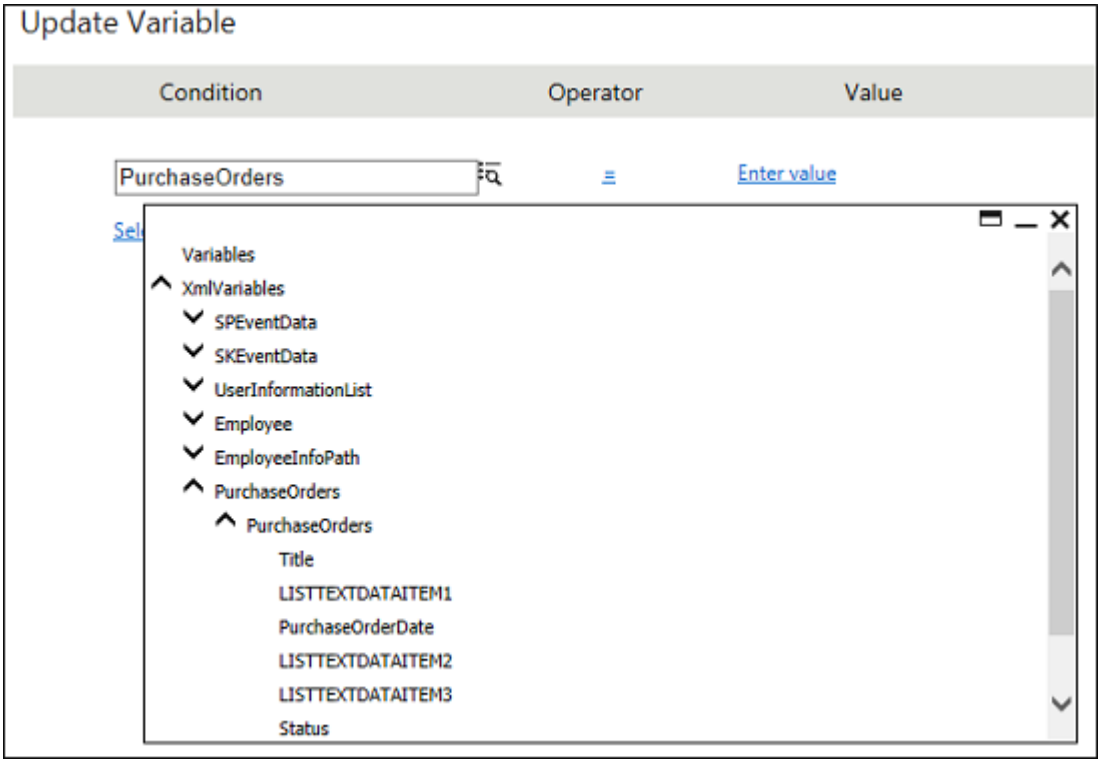
8. Now you can see the XML nodes in a tree menu that are mapped to the controls while creating the list. You can also notice that the parent XML nodes and child XML nodes (PurchaseOrderDetails grid) are also displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window.
10. Drag and drop **Update Variable** activity to add an item in the list.
11. Click the edit icon in the **Set Variables** property. The following screen is displayed.

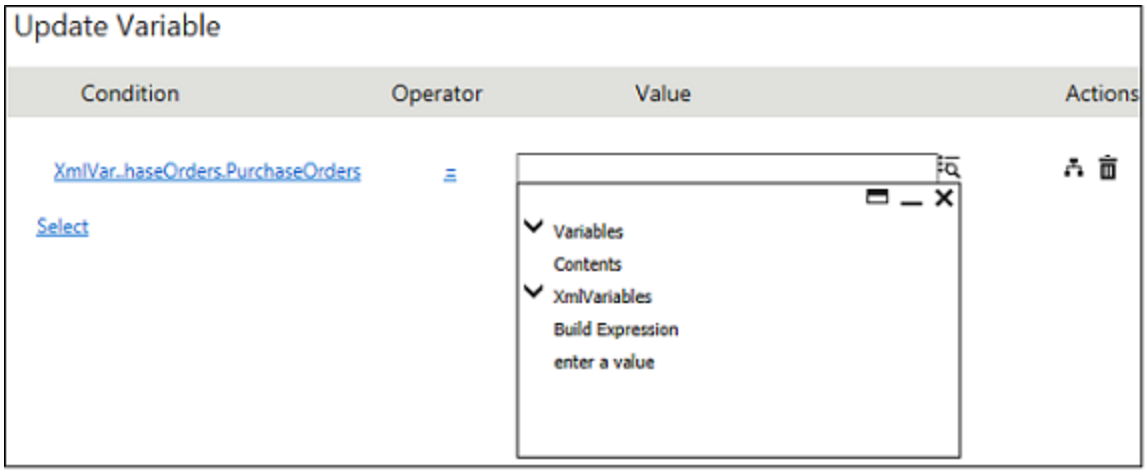


12. Click the **Select** link and select the respective field of the XML List type XML Variable created for the list (in this case 'purchaseorders'). The fields displayed in the tree-view represent the XML nodes of the controls in the list.



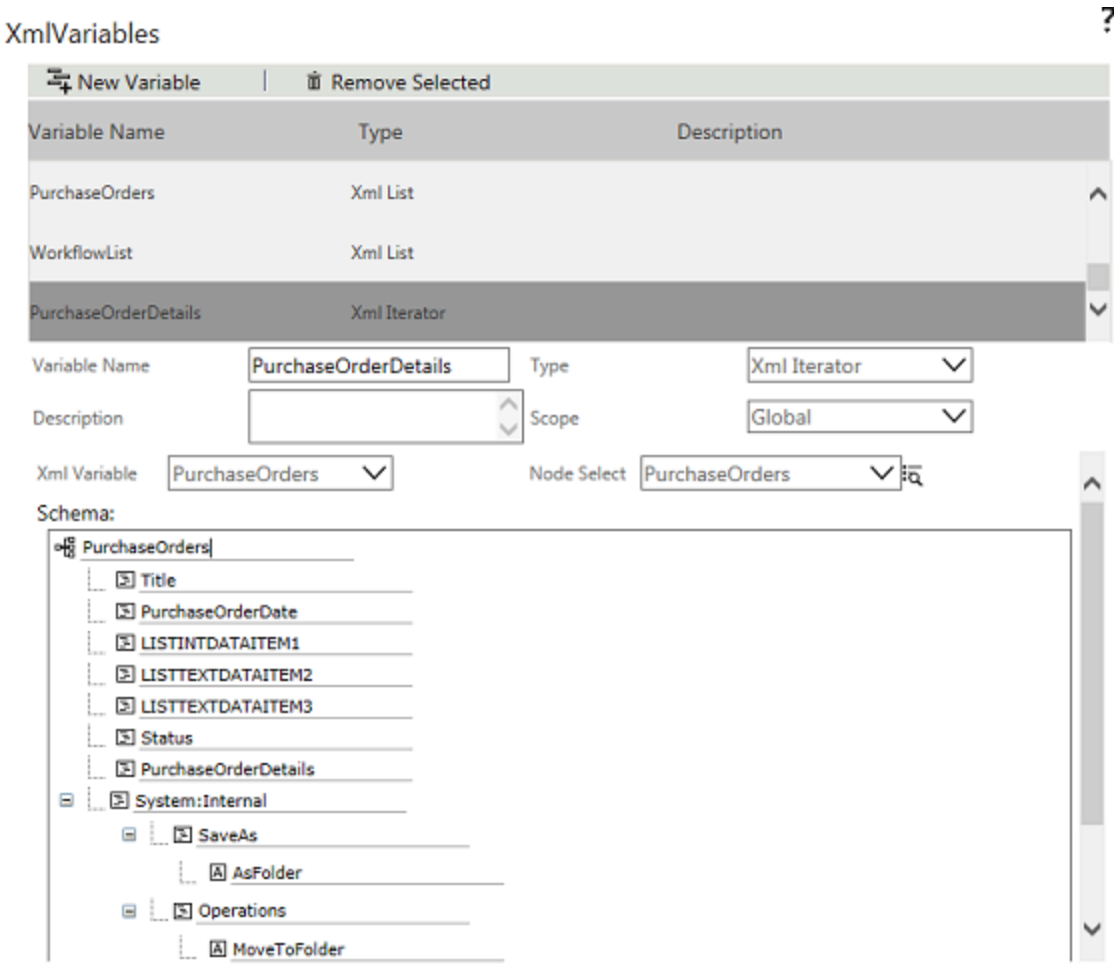
13. Select a field for which you want to set the value from the tree-view. Enter the value for the field in the right-hand side text box. If you want to set the value, select **enter a value** from the tree-view, and then type the value. Also, you can select variables or contents or any XMLvariables nodes directly by selecting the respective node from the tree-view. If you want to combine multiple node values, you can build an expression by selecting **Build Expression**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



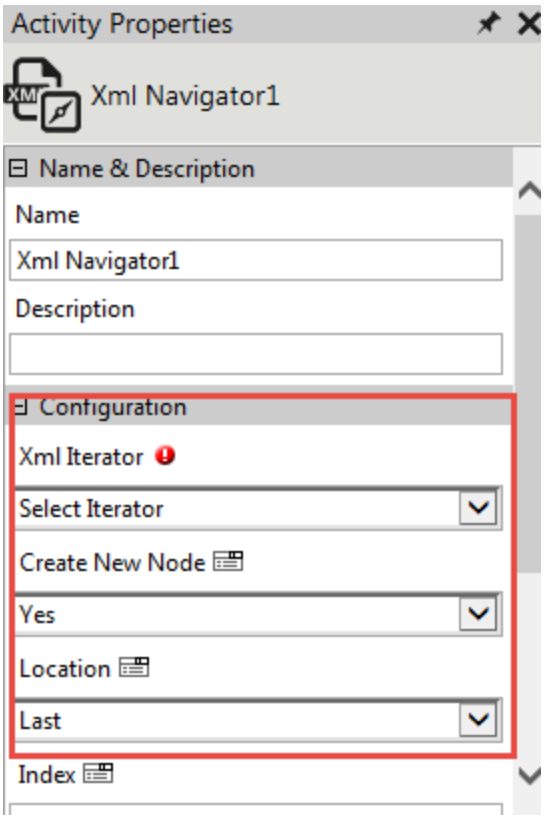
14. After setting the value for all the fields in the parent item (controls in the main form) that you want to add, click OK. This will add a list item to the list, while executing the workflow, with the fields in the main form. If you want to add an item to the grid (in this case, if you want to add a Purchase Order Details grid item), use XML Navigator activity. XML Navigator can be used to point to an item in the grid (Purchase Order Details in this case). For this purpose, you need to create an XML Iterator type XML Variable. This can be created from the Start activity XML variable property.
15. In the XML Variable screen, click **New Variable**.
16. Type the variable name as **PurchaseOrderDetails**.
17. Select **XML Iterator** from the **Type** drop-down list.
18. Select **PurchaseOrders** from the **XML Variable** drop-down list.
19. Select the repeating node (Qualification) from the **Node Select** drop-down list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

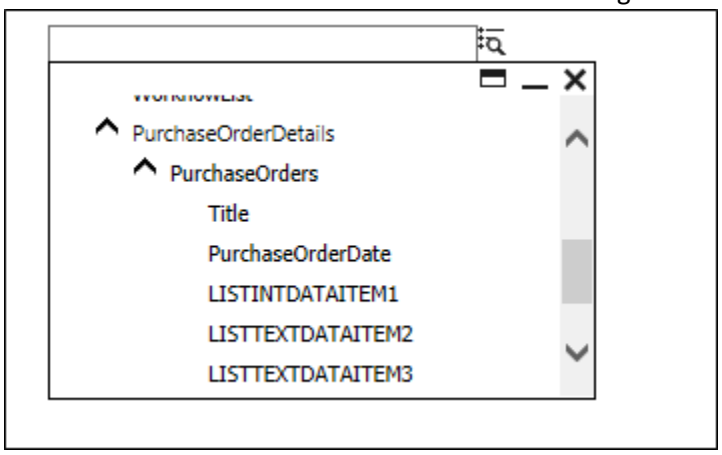


20. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window.
21. Drag and drop **XML Navigator** activity.
22. Select **PurchaseOrderDetails** from **XML Iterator** drop-down list and select **Yes** from **Create New Node** properties, to create a new node.
23. Select the appropriate **Location**. The available options are:
    - First: Navigates to the first record in the collection.
    - Last: Navigates to the last record in the collection.
    - Next: Navigates to the next record based on the current location.
    - Previous: Navigates to the previous record based on current location.
    - Index: Navigates to the record identified in the Index field.
      Since we have to append the items in Purchase Order Details grid, select the option as Last.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



24. Click **Save**.
25. Drag and drop the **Update Variable** activity to enter the grid field values.
26. Click the edit icon in the **Set Variables** property.
27. Click **Select** and choose the respective field of the XML Iterator type XML Variable created for the list (in this case PurchaseOrderDetails), that is, grid items. The fields displayed in the tree-view represent the XML nodes of the controls in the Purchase Order Details grid.



28. Select a field for which you want to set the value from the tree-view and type the value for the field in the right-hand side text box.
29. After setting the value for all the fields in the child item, click **OK**. A new item with grid will be added to the list while executing the workflow.

Note: If you want to add multiple items, use XML Navigator and Update Variable activities for Iterator within the For-Next Loop activities.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Add New Grid Item to the Existing List Item

AVEVA Work Tasks allows you to add list items with grid items to the existing list item using XML variable. The Update Variable activity can be used to perform this task. An XML Iterator type XML variable should be created to add the grid items (child items) and the XML Navigator activity can be used to point to the grid. Also, you have to use the List Query property of List Navigator activity to build the query and to point to the list item to add a new grid.

This can be explained through the scenario mentioned below.

A list, Purchase Orders, has been created with the following fields.

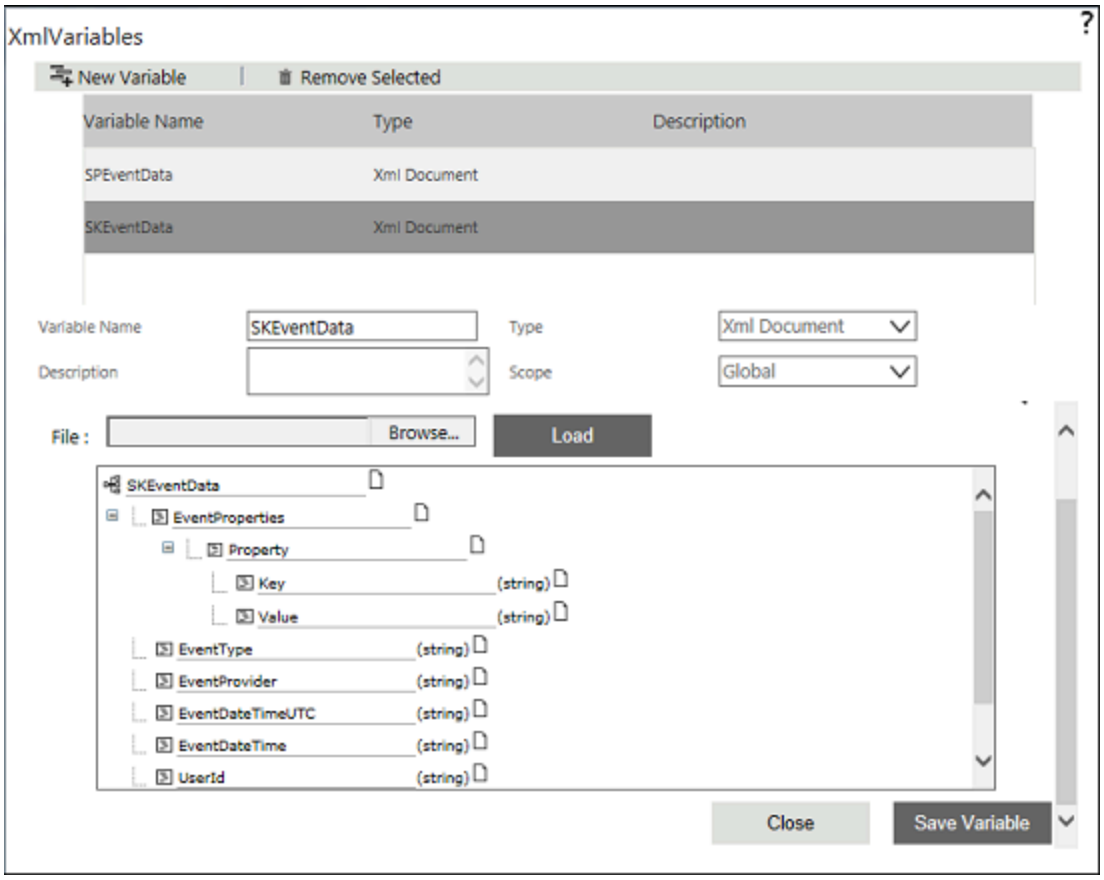
- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price
- Item Total
- Total Purchase amount

From the workflow, you have to add a grid item to a particular order say Reference No. 456789 in the Purchase Orders list.

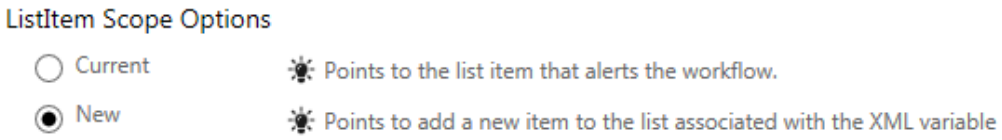
To add a new grid item to the existing list item

1. Login to the **Enterprise Console** and go to **Workflow Designer**. You need to create an XML List type XML variable for the Purchase Orders list. If the variable is not created, follow the below mentioned procedure to create it.
2. Click the edit icon in the **XMLVariables** property from the Start Activity Properties. The following screen is displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



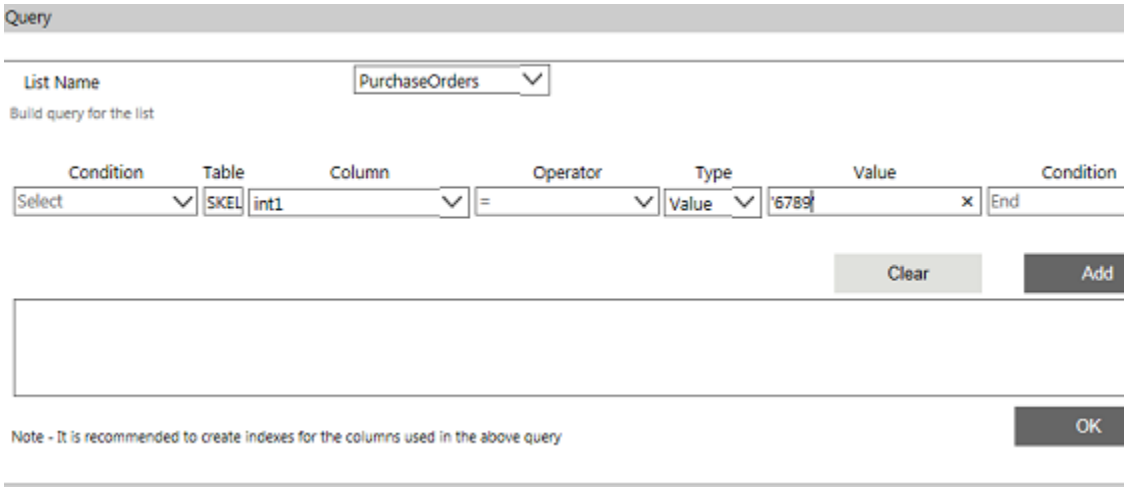
3. Click **New Variable** to add a new variable.
4. Enter the variable name as purchaseorders.
5. Select **XML List** from the **Type** drop-down list.
6. Enter the List name or click the icon to select the list. In this case, select **Purchase Orders** list.
7. Select **New** as the scope from the **ListItem Scope Options** to point to a new item in the **Purchase Orders** list.



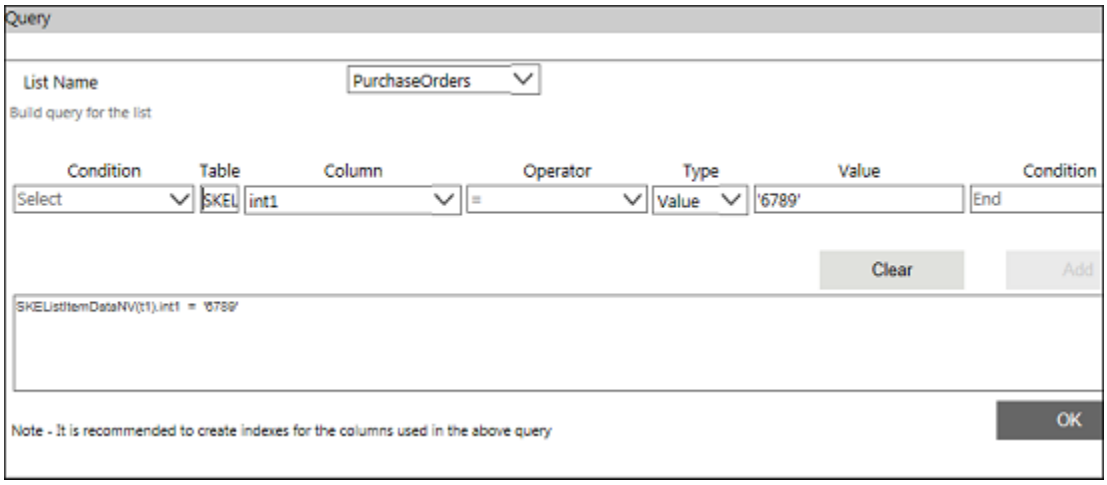
8. Now you can see the XML nodes in a tree-view that are mapped to the controls while creating the list. You can also notice that the parent XML nodes and child XML nodes (PurchaseOrderDetails grid) are also displayed.
9. Click **Save Variable**. A message is displayed saying '**Variable Updated**'. Close the window.
10. Drag and drop List Navigator activity and select **List Query** as the scope.
11. Click edit icon in the **List Query** property. The following screen is displayed.
12. Click the icon to build the query for the list **Purchase Orders**. The following screen is displayed.
13. Select **Purchase Orders** from the List Name drop-down list.
14. Click **Table** field to select the table.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

15. Select SKEListItemDataNV displayed on the top of the tree-view, since you are building a query for an item in the parent table. This will be displayed automatically in the text box.
16. Click the **Column** drop-down list.
17. Select int1 from the options as it is mapped to the Reference No field.
18. Select the operator '**=**' from the Operator drop-down list.
19. Select **Value** from the Type drop-down list.
20. Type '456789' in the text box.
21. Select **End** from the **Condition** drop-down list to end the query.

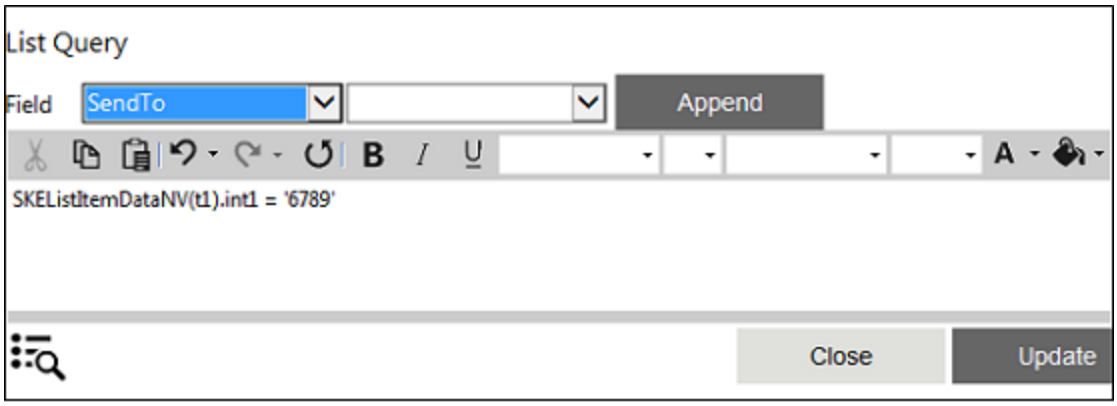


22. Click **Add** to add the condition in the text field.



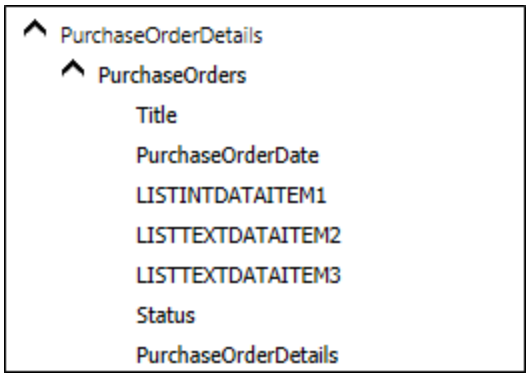
23. Here, SKEListItemDataNV is the table name where the list has been stored and t1 this the internal name that has been used for the main table. If there is any child table also, it will display as t2. Int1 this the field name to which you want to point your XML variable and 6789 is the value for that field.
24. Click **OK**, and then click **Close**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



25. Click **Update**.
26. If you want to add an item to the grid (in this case, if you want to add a Qualification grid item), use **XML Navigator** activity. XML Navigator can be used to point to an item in the grid (PurchaseOrderDetails in this case). For this purpose you need to create an XML Iterator type XML Variable. This can be created from the Start activity XML variable property.
27. On the XML Variable screen, click **New Variable**.
28. Type the variable name as `purchaseorderdetails`.
29. Select XML Iterator from the **Type** drop-down list.
30. Select purchaseorders from the XML **Variable** drop-down list. Select the repeating node (Qualification) from the **Node Select** drop-down list.
31. Click **Save Variable**. A message is displayed saying '*Variable Updated*'. Close the window.
32. Drag and drop XML Navigator activity.
33. Select purchaseorderdetails from XML Iterator drop-down list. Select Yes from Create New Node properties to create a new node.
34. Select the appropriate Location. The options available are:
    - **First:** Navigates to the first record in the collection.
    - **Last:** Navigates to the last record in the collection.
    - **Next:** Navigates to the next record based on the current location.
    - **Previous:** Navigates to the previous record based on current location.
    - **Index:** Navigates to the record identified in the Index field.
35. Since we have to append the items in **Purchase Order Details** grid, select the option as **Last**.
36. Click **Save**.
37. Drag and drop the **Update Variable** activity to enter the grid field values.
38. Click the edit icon in the **Set Variables** property.
39. Click **Select** link and choose the respective field of the XML Iterator type XML Variable created for the list (in this case purchaseorderdetails), that is grid items. The fields displayed in the tree-view represent the XML nodes of the controls in the **PurchaseOrderDetails** grid.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



40. Select a field for which you want to set the value from the tree-view. Enter the value for the field in the right-hand side text box. If you want to set the value, select 'enter a value' from the tree-view and then enter the value. Also, you can select variables or contents or any XML variables nodes directly by selecting the respective node from the tree-view. If you want to combine multiple node values, you can build an expression by selecting Build Expression.
41. After setting the value for all the fields in the child item, click **OK**. Now, a grid item will be added to the list while executing the workflow.

Note: If you want to add multiple items, use XML Navigator and Update Variable for Iterator within the For-Next Loop activities.

Copy List Item with Grid Items

AVEVA Work Tasks allows you to copy a list item with grid items to another list provide both the lists have the same schema. You can perform this task with the help of Update Variable activity. To iterate through the grid items, you need to create XML Iterator type XML variable for the grid. This feature can be explained with the help of a scenario.

"Telematics" is an organization which maintains a Purchase Orders list to keep a track of all the purchase orders raised by the company.

The purchase Orders contains details like:

- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price
  - Item Total
- Total Purchase amount

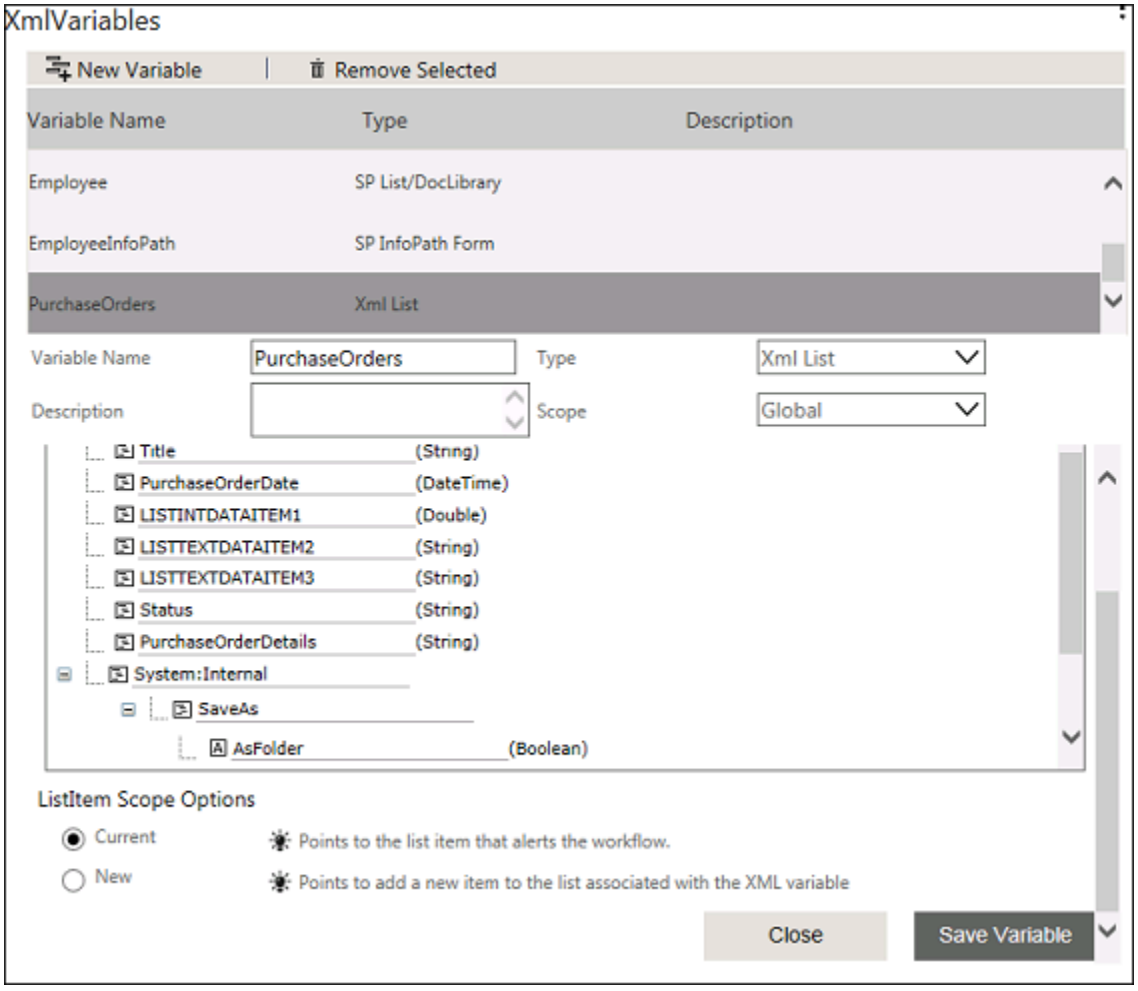
Telematics also maintains another list called Approved Purchase Order which has the same schema of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Purchase Order list. Now you want to copy an item with grid (Purchase Order approved) to the Approved Purchase Order list when the item is approved.

To copy a list item with grid items

1. Login to the Enterprise Console and go to Workflow Designer.
2. You need to create an XML List type XML variable for the Purchase Order list. If the variable is not created, follow the below mentioned procedure to create.
3. Click the edit icon in the XMLVariables property from the Start Activity Properties.
4. Here, for Purchase Order list you need to create an XML List type XML variable to work with the current purchase order item and XML Iterator type XML variable to loop through the child/grid nodes of the current purchase order item.
5. Click New Variable to add a new variable.
6. Type the variable name as 'purchaseorders'.
7. Select type as XML List from the Type drop-down list.
8. Type the List name in the Select List field or click the icon to select the list. In this case, select Purchase Orders list.
9. Select Current as the scope from the ListItem Scope to point to the an item in the list.

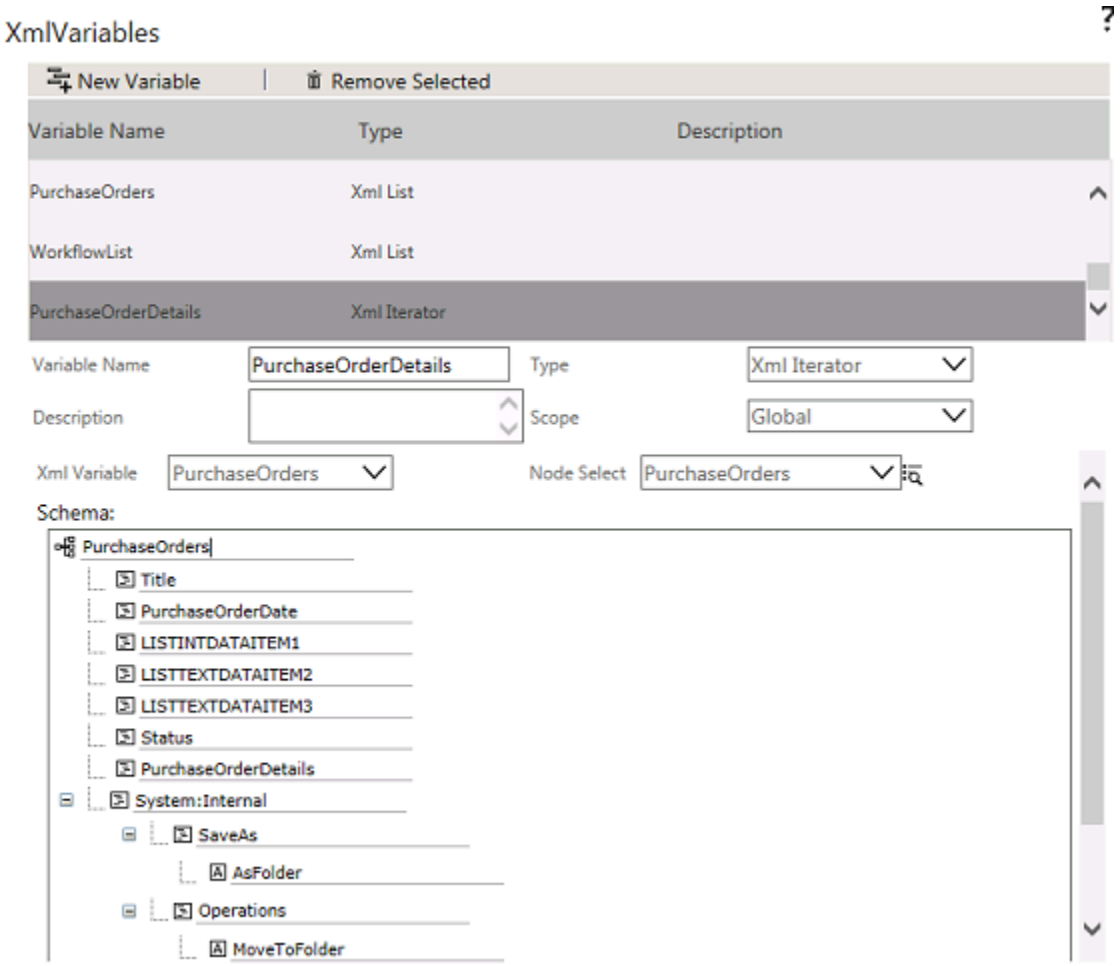


10. Now you can see the XML nodes in a tree-view that are mapped to the controls while creating the list. You

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

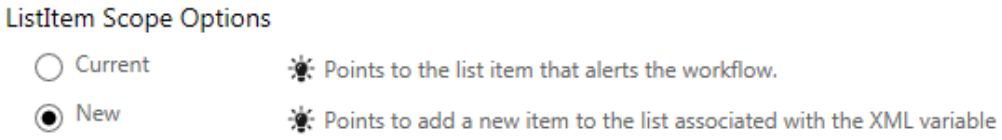
can notice that the parent XML nodes and child XML nodes (PurchaseOrderDetails grid) are also displayed.

11. Click Save Variable. A message is displayed saying 'Variable Updated'.
12. Now, you need to create XML Iterator type XML variable to iterate through the child nodes (in this case, PurchaseOrderDetails). For that, click New Variable from the XML Variable screen.
13. Enter the variable name as purchaseorderdetails.
14. Select XML Iterator as type from the Type drop-down list.
15. Select purchaseorders from the XML Variable drop-down list.
16. From the Node Select drop-down, select the repeating node, that is PurchaseOrderDetails.

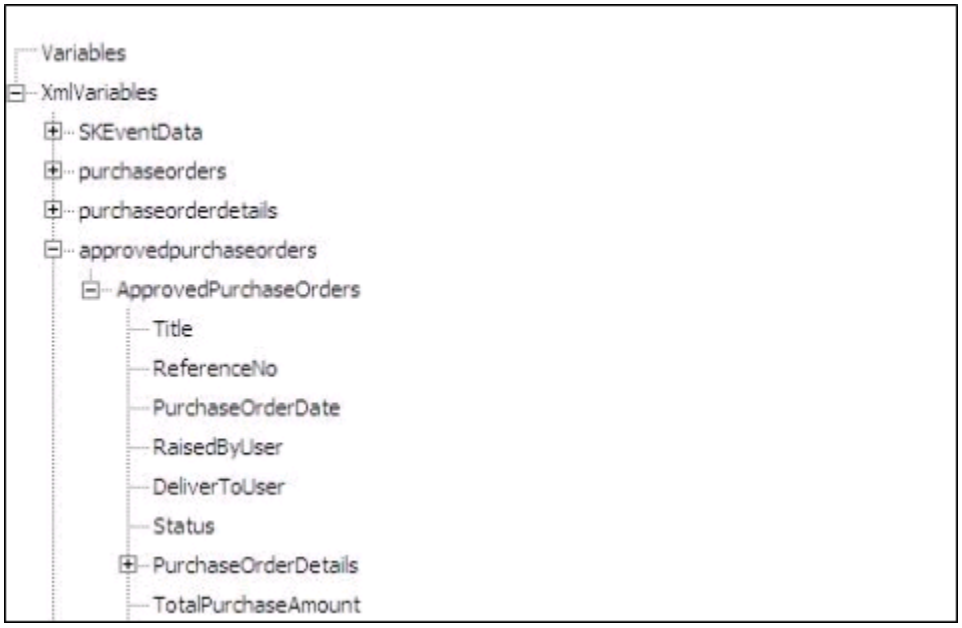


17. Click Save Variable. A message is displayed saying 'Variable Updated'.
18. In the same way you have to create XML variables for Approved Purchase Order list. For Approved Purchase Order list, you need to create XMLList type XML variable to add an item and a XML Iterator type XML variable to map to the grid items of the new purchase order item.
19. Click New Variable to add a new variable, from the XML Variable property of Start activity.
20. Enter the variable name as approvedpurchaseorders in the Variable Name field.
21. Select XML List as type from the Type drop-down list.
22. Enter the List name in the Select List field or click the icon to select the list. In this case, select Approved Purchase Orders list.
23. Select New as scope from the ListItem Scope Options to point to the new item in the list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



24. Now you can see the XML nodes in a tree-view that are mapped to the controls while creating the list. You can also notice that the parent XML nodes and child XML nodes (PurchaseOrderDetails grid) are also displayed.
25. Click Save Variable. A message is displayed saying 'Variable Updated'.
26. You need to create XML Iterator type XML variable to iterate through the child nodes. For that, click New Variable from the XML Variable screen.
27. Enter the variable name as approvedpurchaseorderdetails.
28. Select XML Iterator as type from the Type drop-down list.
29. Select approvedpurchaseorders from the XML Variable drop-down list.
30. From the Node Select drop-down, select the repeating node, that is PurchaseOrderDetails.
31. Click Save Variable. A message is displayed saying 'Variable Updated'.
32. Drag and drop Update Variable activity to set value for the parent nodes or the item.
33. Click the edit icon in the Set Variables property.
34. Click the Select link and select the respective field of the XML List type XML Variable created for the list (in this case approvedpurchaseorders). The fields displayed in the tree-view represent the XML nodes of the controls in the list.



35. Select a field for which you want set the value from the tree-view under approvedpurchaseorders. The selected variable for the field Title will be displayed in the text box as 'XMLVariables.approvedpurchaseorders.ApprovedPurchaseOrders.Title.'
36. Enter the value for the field in the right-hand side text box. Select the respective field of the purchaseorders XML variable (purchase order item to be copied). The selected XML variable for the field will be displayed as 'XMLVariables.purchaseorders.PurchaseOrders.Title'. Set the value for all other fields in the same way.
37. After setting the value for all the fields in the parent form (controls in the main form) that you want to add,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

click Ok. This will add a list item to the new list (Approved Purchase Order), while executing the workflow, with the values in the current Purchase Orders item.

38. Next, you need to iterate through the grid/child items of the current purchase order and to add the child items in the new list (Approved Purchase Order). Drag and drop ForEach-Loop activity to loop through the grid items.
39. Select XML Iterator type XML variable that you have created for the current Purchase Order from the XML Iterator drop-down list (in this case, purchaseorderdetails).
40. If you want to add an item to the grid (in this case, if you want to add a purchase order details grid item), use XML Navigator activity. XML Navigator can be used to point to an item in the grid (purchaseorderdetails in this case).
41. Drag and drop XML Navigator activity.
42. Select approvedpurchaseorderdetails from XML Iterator drop-down list and select Yes from Create New Node properties to create a new node.
43. Select the appropriate Location. The available options are:
    - First: Navigates to the first record in the collection.
    - Last: Navigates to the last record in the collection.
    - Next: Navigates to the next record based on the current location.
    - Previous: Navigates to the previous record based on current location.
    - Index: Navigates to the record identified in the Index field.

    Since we have to append the items in Purchase Order Details grid, select the option as Last.
44. Click **Save**.
45. Drag and drop **Update Variable** activity to set the value for the grid item fields of the new list item.
46. Click the edit icon in the **Set Variables** property.
47. Click the Select link and select the respective field of the XML List type XML Variable created for the list (in this case ApprovedOrderDetails). The fields displayed in the tree-view represent the XML nodes of the controls in the list.
48. Select the XML variable PurchaseOrderDetails from the tree-view. You can either select PurchaseOrderDetails node or select all child nodes separately. In this case, you are selecting the PurchaseOrderDetails node. The variable will be displayed in the text box as 'XMLVariables.approvedpurchaseorderdetails.PurchaseOrderDetails'.
49. Set the value in the right-hand side text box. Select the respective field of the purchaseorderdetails XML variable (purchase order item to be copied) from the tree-view. The selected variable will be displayed in the text box as 'XMLVariables.purchaseorderdetails.PurchaseOrderDetails'.
50. After setting the value for the fields in the child item that you want to add, click OK. This will add a grid item to the new list (Approved Purchase Order), while executing the workflow, with the values in the current (Purchase Order) list.

    Note: You can also select each node in the grid separately and set the value for each node.
51. Complete the loop with a Next activity and finish the workflow with appropriate link conditions.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Iterate through the List Item and Grid Item of the List

AVEVA Work Tasks allows you to loop through the list items and grid items of a list. A XML List Iterator type XML variable can be used to iterate through the list items, where XML Iterator type XML variable can be used to iterate through the grid items of a list. This feature can be explained with the help of a Purchase Order list scenario.

The Purchase Orders contains details like:

- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price
  - Item Total
- Total Purchase amount

Now you want to loop through the approved purchase order items (items with status as Approved) as well as its child items.

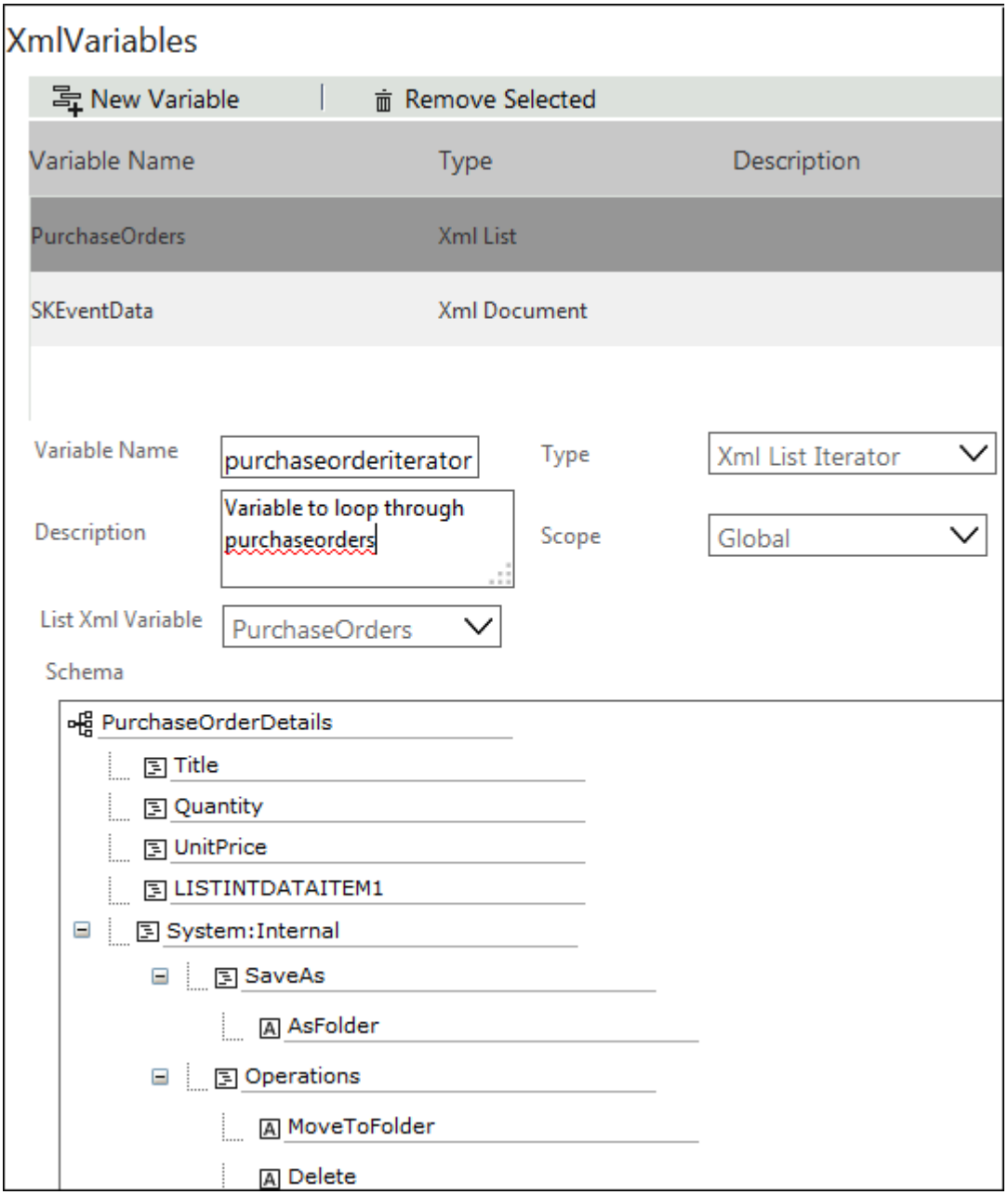
To iterate through the list item and grid item of the list

1. Log on to the **Enterprise Console** and go to **Workflow Designer**.

   Note: As a first step, you need to create an XML List type XML variable for the Purchase Orders list. If the variable is not created, follow the below mentioned procedure to create.
2. Click the edit icon in the **XMLVariables** property from the **Start** Activity Properties.
3. Click **New Variable** to add a new variable.
4. Enter the variable name as **purchaseorders**.
5. Select **XML List** from the Type drop-down list.
6. Enter the List name or click the icon to select the list. In this case, select Purchase Orders list.
7. Select the scope from the **ListItem Scope Options**. Later, you can change the scope of the variable to List Query to query the approved purchase orders using List Navigator activity.
8. Click **Save Variable**. A message is displayed saying 'Variable Updated'.
9. Now, you need to create XML List Iterator type XML variable to iterate through the approved purchase order items. Click **New Variable** from the XMLVariable screen.
10. Type the variable name as **purchaseorderiterator**.
11. Select **XML List Iterator** from the **Type** drop-down list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

12. Select **purchaseorders** from the **List XML Variable** drop-down list.



13. Click **Save Variable**. A message is displayed saying 'Variable Updated'.
14. You need to create XML Iterator type XML variable for the purchaseorderIterator (XML List Iterator type) XML variable to iterate through the child nodes of the currently iterating purchase order item. Click New Variable from the XML Variable screen.
15. Type the variable name as **purchaseorderdetails**.
16. Select **XML Iterator** from the **Type** drop-down list.
17. Select **purchaseorderiterator** from the **XML Variable** drop-down list.
18. From the **Node Select** drop-down List, select the repeating node that is, PurchaseOrderDetails.
19. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Use **List Navigator** activity to query the approved purchase order items.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

20. In this case, the following query is used to retrieve the approved purchase orders.

    SKEListItemDataNV(t1).nvarchar3 = 'Approved', where **SKEListItemDataNV** is the table used to store the list items and **nvarchar3** is the table column mapped to the Status field.

    Use ForEach-Loop and Next Loop activities to loop through the approved purchase order items.
21. Select **purchaseordersiterator** from the **XML Iterator** drop-down list of ForEach Loop activity.
22. Use any action as per the requirement to retrieve the data. You need to access the value of the parent nodes from the purchaseordersiterator XML variable.

    Use another set of ForEach-Loop and Next activities within the existing loop to iterate through the grid items (PurchaseOrderDetails).
23. Select purchaseordedetails from the XML Iterator drop-down list of ForEach Loop activity.
24. Use any action as per the requirement to retrieve the data. You need to access the value of the parent nodes from the purchaseorderdetails XML variable.

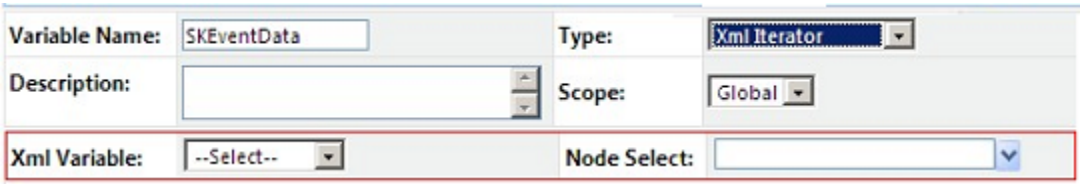
XML Variable of Type XML Iterator

XML Iterator Variables allows you to define variables for repeating XML elements and attributes. This variable type represents a collection of XML nodes at any level in the XML Variable schema. It allows you to perform actions on XML nodes such as looping through XML nodes and deriving the value of a specific node at any location.

Note: The XML Iterator type option is available only if another variable of XML Document type exists.

XML Variable of type 'XML Iterator'

When the variable type is changed to **XML Iterator**, following extra properties are displayed as seen in the image below.



Property Description

'XML Iterator' XML Variable Property	Description
XML Variable	All variables of type XML Document and XML Iterator are listed for selection. Select the parent XML Variable. The parent variable of an XML Iterator variable can be another XML Iterator Variable.
Node Select	Only nodes in the parent variable are displayed in this drop-down. Select the node that the variable represents.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

See Create XML Variable of Type XML Iterator for more information.

Create XML Variable of Type XML Iterator

To create XML Variable of type XML Iterator Variable:

- Click **New Variable** button in the XML Variable Interface.
- Enter the following information:

  Variable Name - Enter the name of the variable.

  Description - Enter a description for the variable.

  Type - This field defaults to XML Document. Select XML Iterator from the drop-down. The XML Iterator type option is available only if another variable of XML Document type exists.

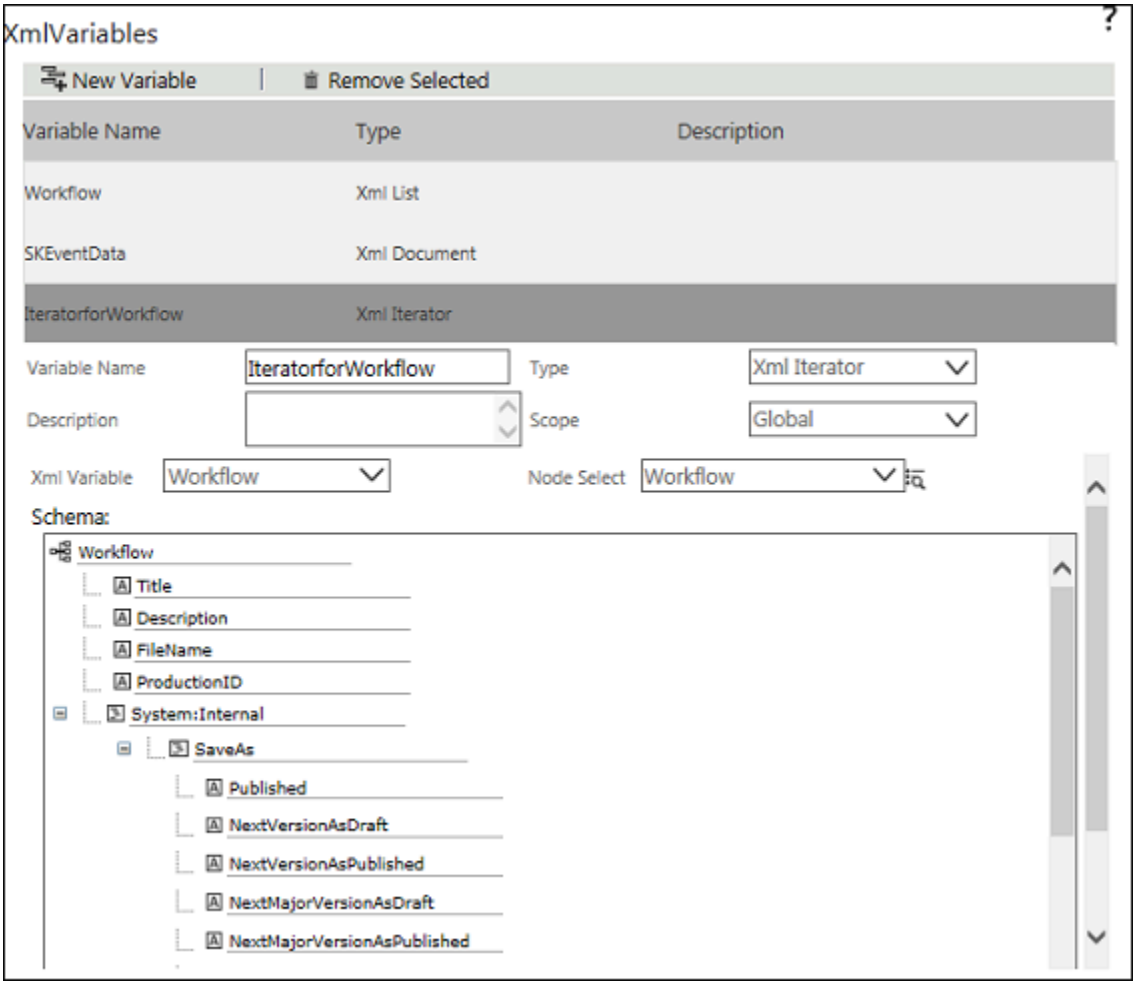
  Scope - Select the scope of the variable. The scope of the variable can be either Global or Local.
- When the variable type is changed to **XML Iterator**, two additional properties are displayed as seen in the image below.

  XML Variable - Select the parent XML Variable. All variables of type XML Document and XML Iterator are listed for selection. The parent variable of an XML Iterator variable can be another XML Iterator Variable.

  Select Node - Select the node that the variable represents. Only the nodes in the parent variable are displayed.
- Click **Save Variable**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

XML Iterator Variable Properties



Use Variables of Type XML Iterator

XML Iterator Variables can be used in any of the following ways:

- **Building conditions for setting Activity Properties** - XML Iterator Variables can be used like an XML Variable to build conditions for setting Activity Properties. However, when you select the XML Variable of XML Iterator type, only the nodes under the selected node of the parent variable are shown.
- **In ForEach Loop activity** - XML Iterator variables can be used in the ForEach Loop activity to loop through an activity till it executes for the entire collection in the XMLIterator variable. Example of XML Iterator Variable in For-Each .
- **Navigate to a specific node in the collection** - XML Iterator variables can be used in the XML Navigator Activity to go to a specific node in the collection and then perform an action. Example of XML Navigator Activity

Example of XML Iterator Variable in ForEach Loop

In this example, the XML Iterator Variable is used to get approval from all managers in the Managers.XML file

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

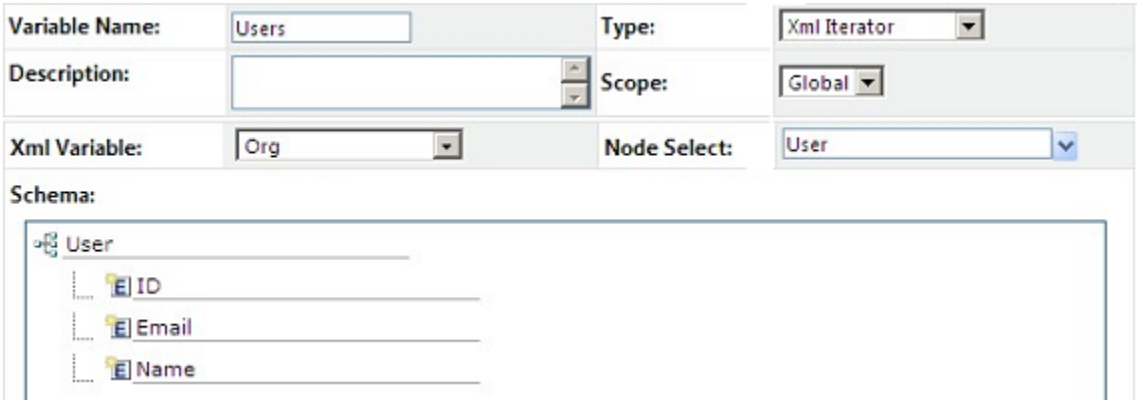
containing the list of all the users who are managers.

Managers.XML file

```
<Managers>
<Users>
<User id=1>
<Name>John</name>
</User>
<User id=2>
<Name>Jim</name>
</User>
<User id=3>
<Name>Victor</name>
</User>
<User id=4>
<Name>Andy</name>
</User>
</Users>
</ Managers >
```

Step 1: XML Iterator Variable is defined for users. The node Users holds the collection of managers.

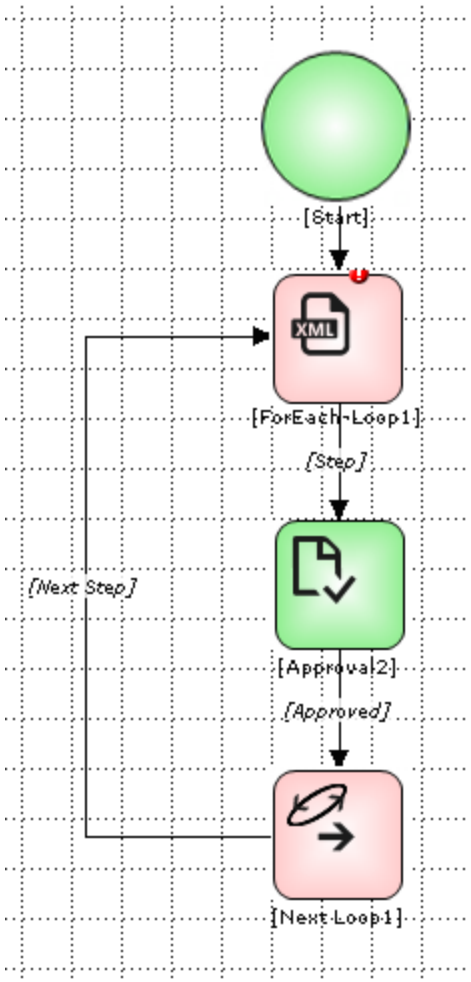
XML Iterator Variable for Managerial Users



Step 2: The approval activity is set within a ForEach Loop activity.

ForEach Loop Activity for Looping Approvals

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

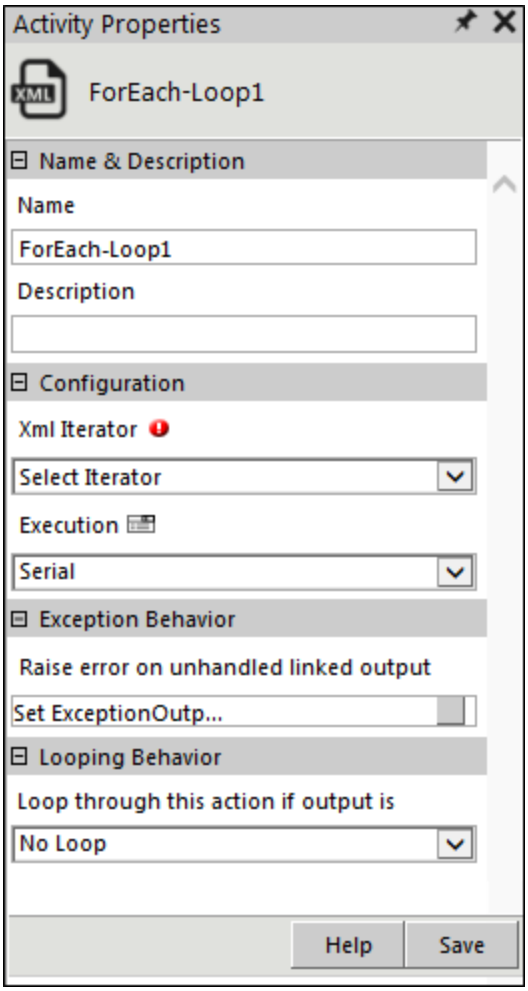


Step 3: The XML Iterator Variable 'Users' is defined in XML Iterator field of ForEach Loop properties. The Execution type is set to Serial as the approval is to be processed serially.

Note: The Execution type is set to Parallel as the approval is to be processed simultaneously by all approvers. When the Execution type is Parallel, the Next Loop activity is not required.

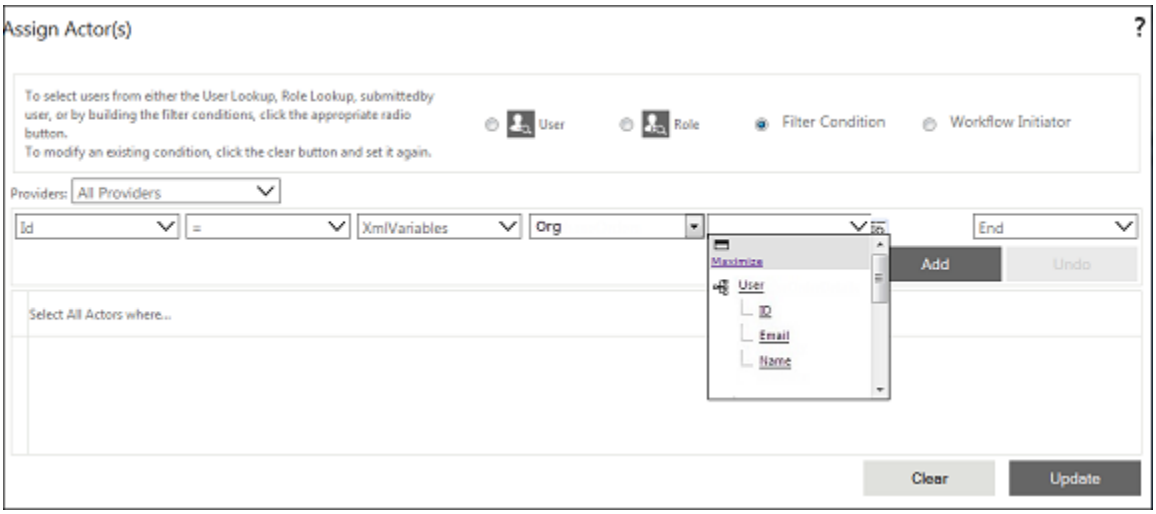
ForEach Loop Property for XML Iterator

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Step 4: The XML Iterator variable is specified in the condition for selecting actors for the approval.

Actor Selection for Approval Activity



Step 5: When the above workflow is invoked, the Workflow Engine resolves "XMLVariables.users.user.ID". It gets the ID of each manager and assigns the task to that manager.

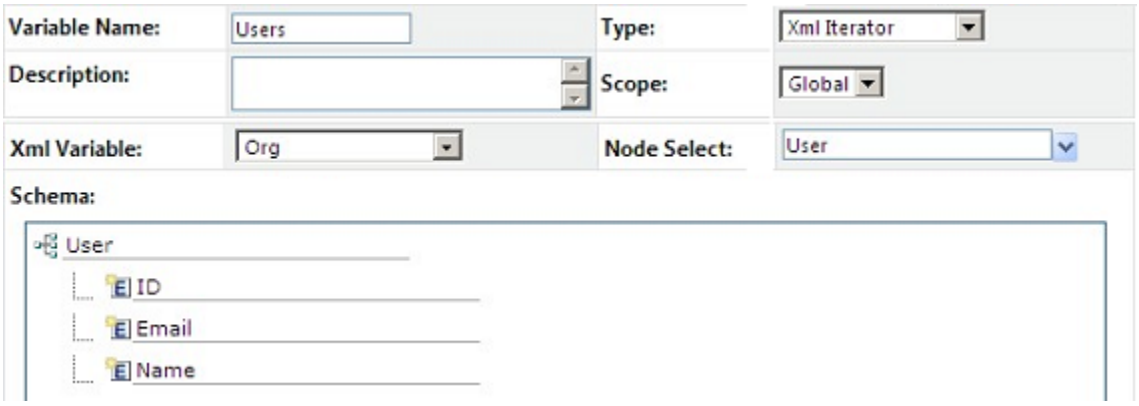
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example of XML Navigator Activity

A company allocates new laptops to employees as and when new inventories are received based on date of requisition. Allotments are to be approved by the manager of the requisitions user. Users with requisitions are added to a waiting list by the Facilities Management System. Allotments are made to the user at the top of the list. When the allotment is complete, the user is removed from the list. If a manager rejects the allotment, the allotment is made to the next person in the waiting list. This example demonstrates how the XML Navigator Activity is used to automate a process.

Step 1: A global XML Iterator variable is defined for the node users.

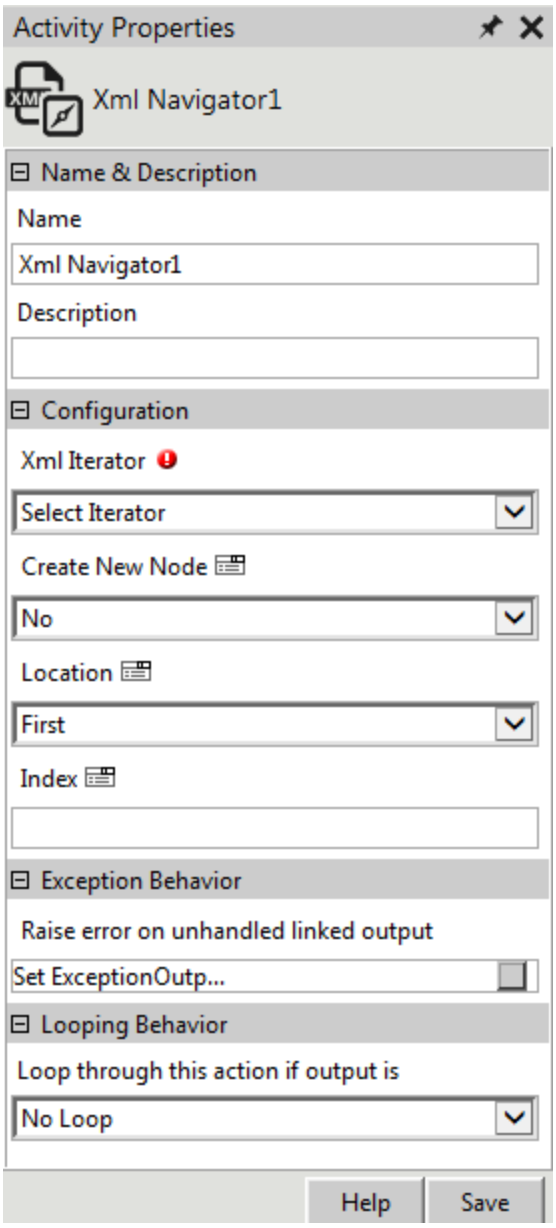
XML Iterator Variable for Users



Step 2: The XML Navigator activity is added to the workflow with **Location** set to First.

XML Navigator Properties

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

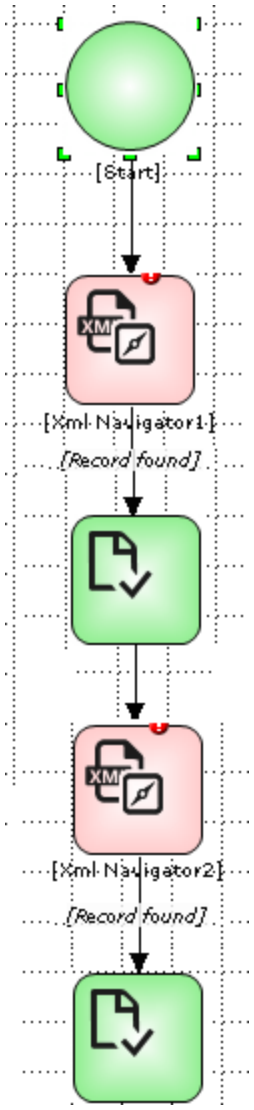


Step 3: An Approval activity is added after the XML Navigator Activity that sends an approval task to the manager of the user at the top of the list. The XMLNavigator links to Approval only if a record is found in the waiting list.

Step 4: The Approval task is linked to another XML Navigator Activity with the property set to **Next**.

Step 5: The following figure depicts the workflow after adding these two properties.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



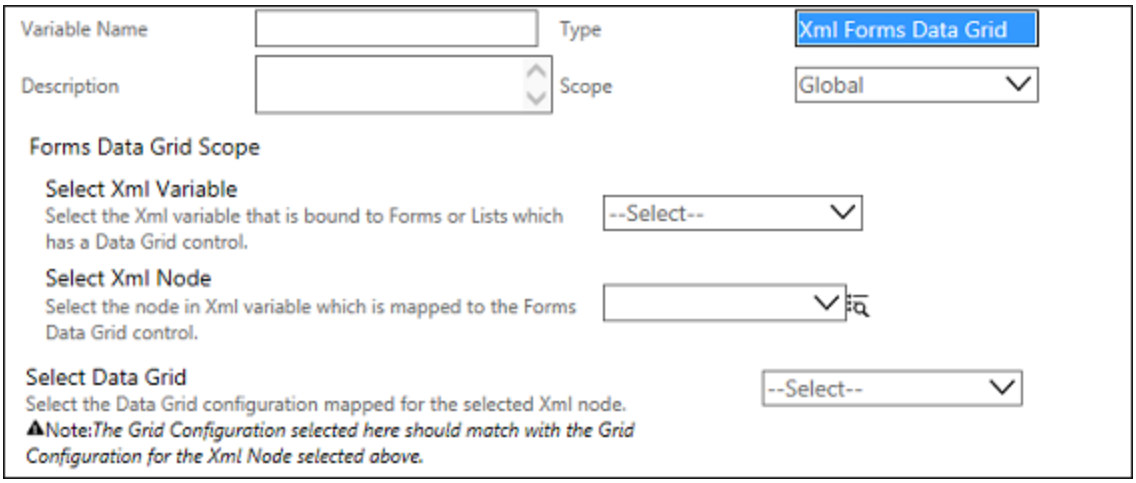
XML Variable of Type XML Forms Data Grid

Xml Forms Data Grid variable is used to read or modify the selected values of a Data Grid control used in AVEVA Work Tasks Forms or Lists.

XML Variable of type 'XML Forms Data Grid'

When the variable type is selected as XML Forms Data Grid, following properties are displayed as seen in the image given below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Property Description:

'XML Forms Data Grid' XML Variable Property	Description
Select Xml Variable	Xml variables of type Xml Document and Xml list are listed here. Select the parent Xml Variable that is mapped to Forms or Lists which have a Data Grid control.
Select Xml Node	Only nodes of the parent variable selected above are displayed here. Select the Xml Node which is mapped to the Forms Data Grid control.
Select Data Grid	Lists the Grid Configurations created. Select the Data Grid Configuration mapped for the selected Xml node. The Grid Configuration selected here should match with the Grid Configuration for the Xml node selected above.

See Creating XML Variable of Type 'XML Forms Data Grid' for more information.

Create XML Variable of Type XML Forms Data Grid

To create XML Variable of type XML Forms Data Grid:

1. Click **New Variable** button in the XML Variable Interface.
2. Enter the following information:

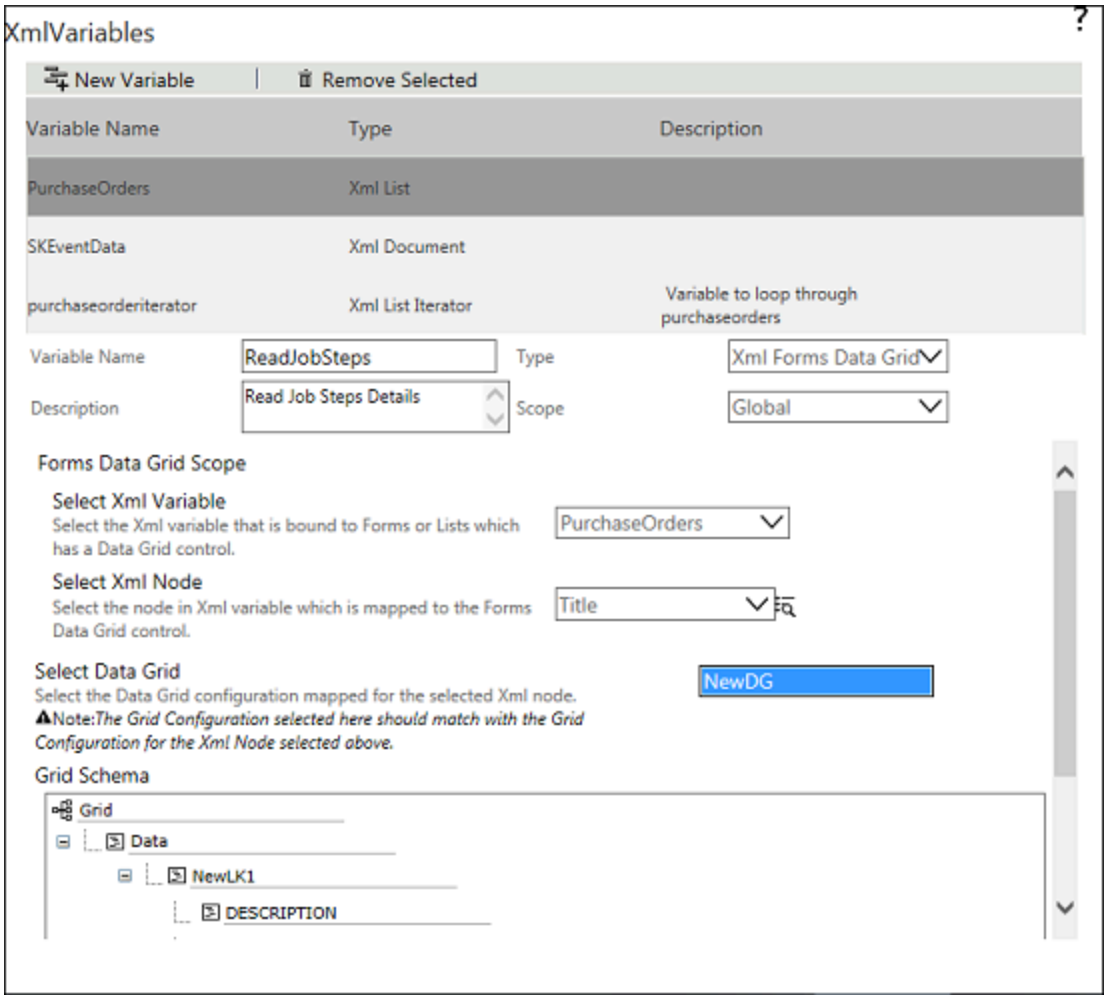
   Variable Name: Enter the name of the variable.

   Description: Enter a description for the variable.

   Type: This field defaults to XML Document. Select XML Forms Data Grid from the drop-down.

   Scope: Select the scope of the variable. The scope of the variable can be either Global or Local.
3. When the variable type is changed to XML Forms Data Grid, following additional properties are displayed as seen in the following image.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



a. **Select Xml Variable** - Xml variables of type Xml Document and Xml list are listed here. Select the parent Xml Variable that is mapped to Forms or Lists which have a Data Grid control.

b. **Select Xml Node** - Only nodes of the parent variable selected above are displayed here. Select the Xml Node which is mapped to the Forms Data Grid control.

c. **Select Data Grid** - Lists the Grid Configurations created. Select the Data Grid Configuration mapped for the selected Xml node. The Grid Configuration selected here should match with the Grid Configuration for the Xml node selected above.

4. Click **Save Variable**.

Use XML Variable of Type XML Forms Data Grid

The XML Forms Data Grid variable usage can be demonstrated using the following two scenarios:

1. Create "Xml Forms Data Grid" type Xml Variable and use it in a workflow to iterate through row selections done in the Data Grid.
2. Use "Xml Forms Data Grid" type Xml Variable to change/add row selections in a Data Grid control using Workflow.

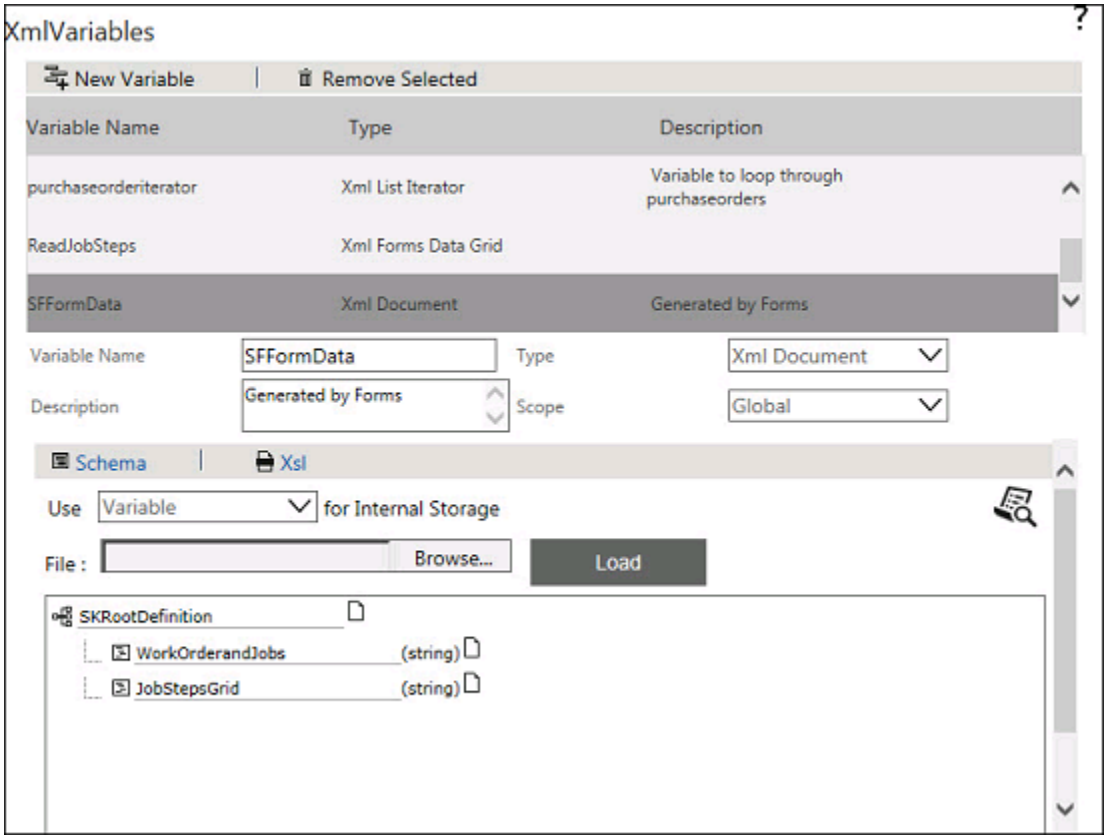
Let us consider a typical **Work Order – Job** manufacturing process scenario. In a manufacturing process, a **work order** is a collection of jobs that produce an item. A **job** is a list of **steps** or procedures that is executed to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

produce an item or a version of an item. The list of steps can be grouped together into **Step Groups**. Multiple jobs can be performed to produce a single item.

Prerequisites

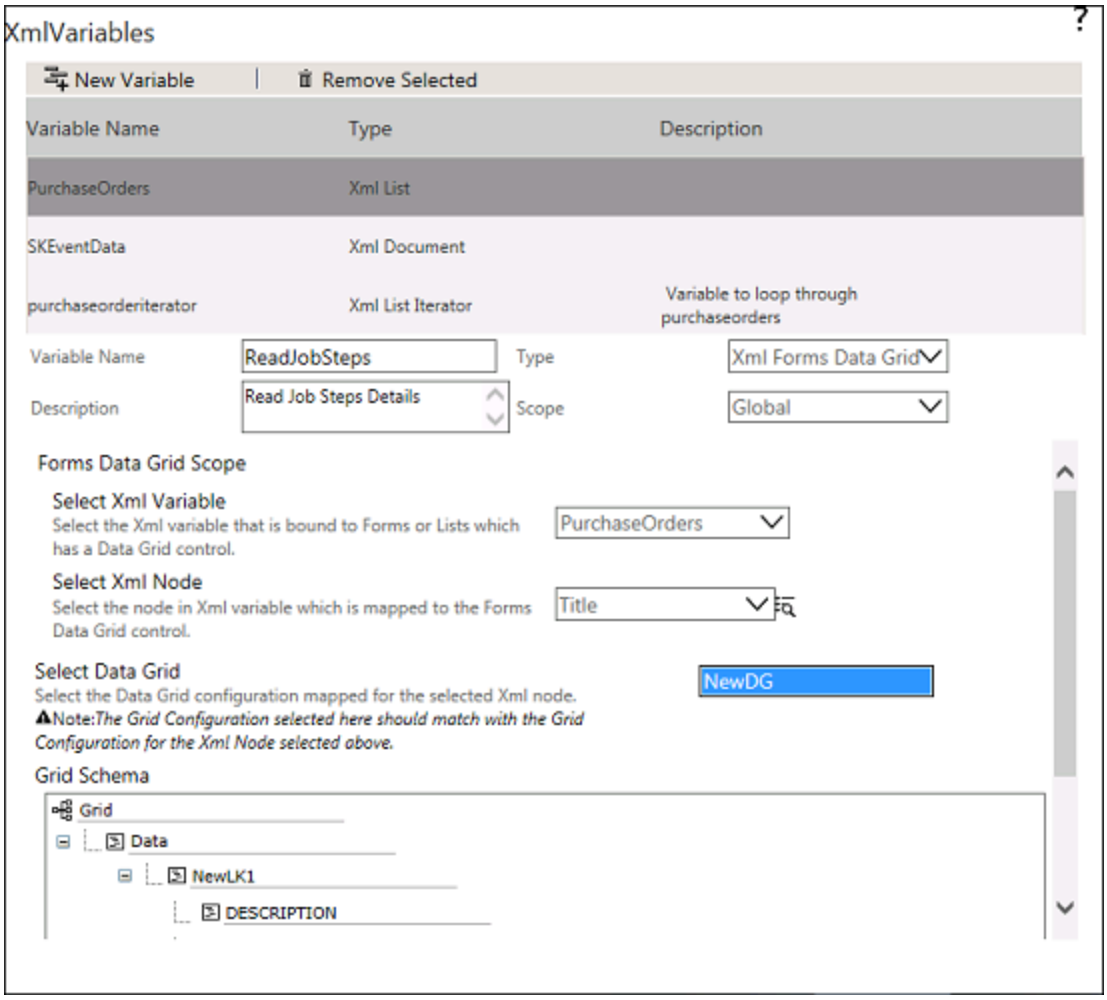
1. AVEVA Work Tasks Form with two Data Grids. One Data Grid will be showing the Work Order and Jobs details and the other would show the Job Step details. When the user selects a Job in the Work Order and Jobs grid, the Job Steps grid will be populated with the relevant Job Steps.
   The grid selection information is stored in the XML nodes mapped to the Grid Control. To extract the required values and use it within workflows, we need to create XML variables of type XML Forms Data Grid.
2. This form is associated to a new Workflow. The XML variable SFFormData will be added to the XML Variable collection. (Form XML variable can also be created without association, i.e. by selecting the corresponding Form from the XML Variable interface).



Scenario 1: Create Xml Forms Data Grid type Xml Variable and use it in a workflow to iterate through selections done in the Data Grid

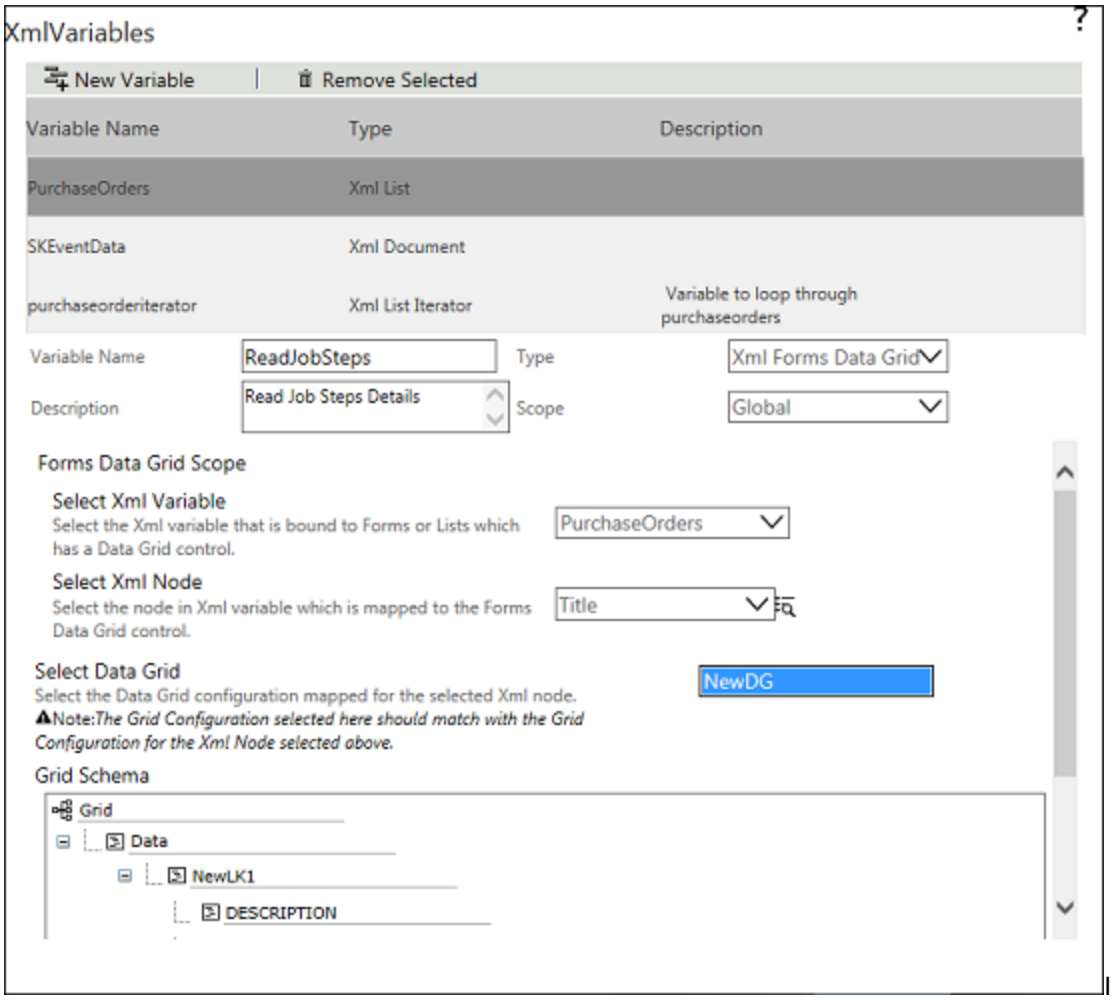
1. Create an XML variable of type XML Forms Data Grid as shown below for Job Steps Grid. Select JobStepsGrid XML Node which is mapped to the Job Steps Grid Configuration. (This will contain the complete selection information of the Data Grid once the form is submitted.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



2. As the selected job can have multiple steps, we need an XML Iterator variable to iterate through the multiple steps selected in the Job Steps Grid. Create an XML Iterator variable 'LoopJobSteps' with repeating node(Node Select) as JobStepLookup.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

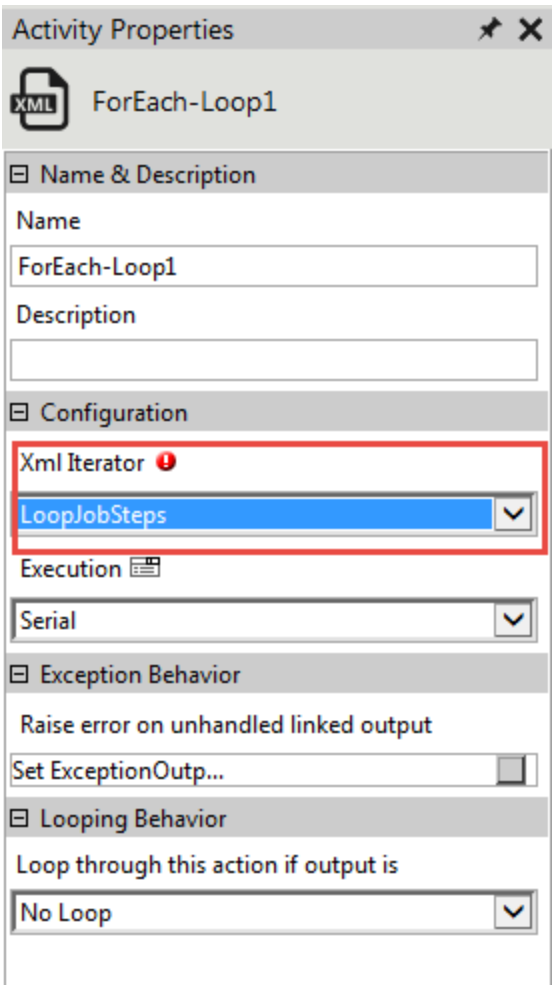


Note: The XML Iterator variable is created to iterate through the multiple selections. To get more information on the selections in the parent hierarchy levels, an XML Iterator needs to be created for the parent node (which is repeating) also.

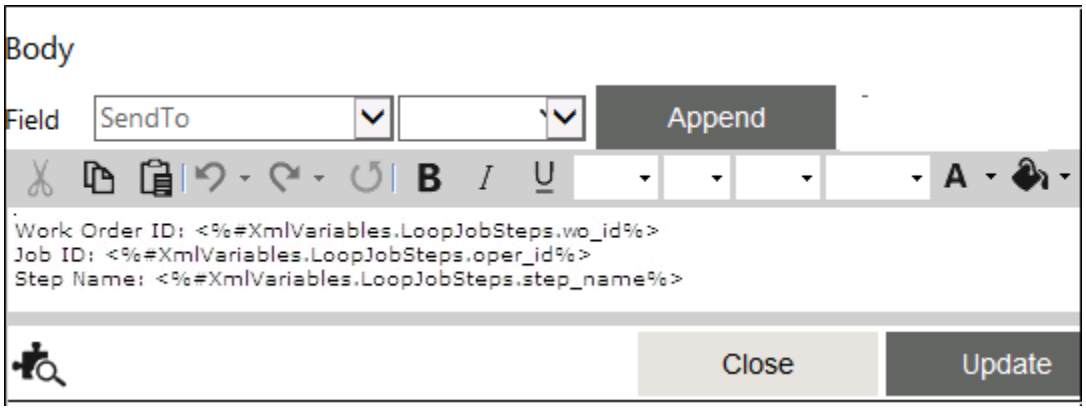
For example, if we have to loop through the multiple selections of Work Order we need to create Iterator variable with the repeating node as 'WorkOrderLookup'.

3. Add a ForEach Loop activity. Set its XML Iterator property with the Iterator variable created above.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

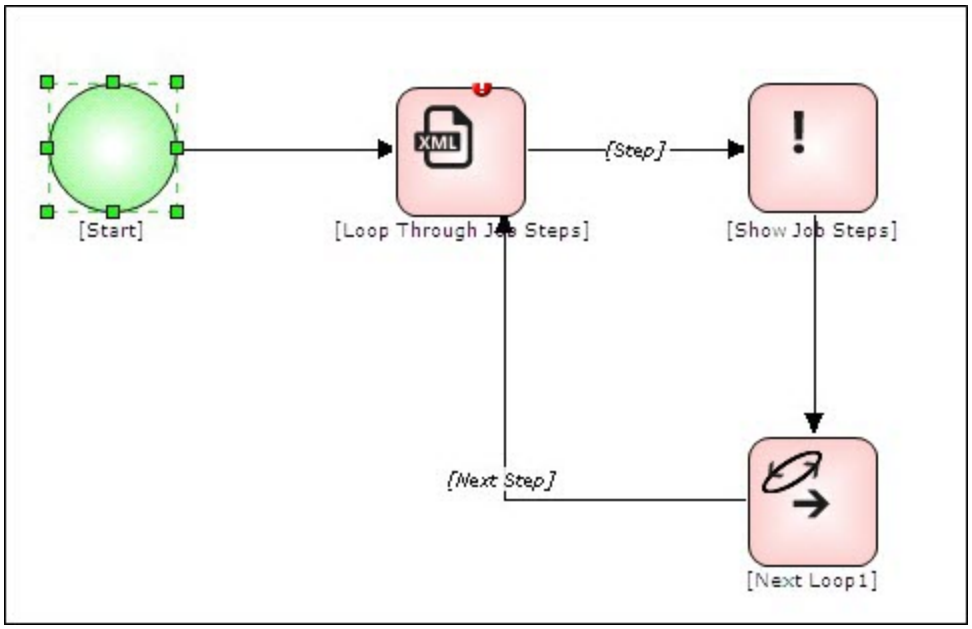


4. Add an Information activity to display the selections made. Set its Body property as shown below:

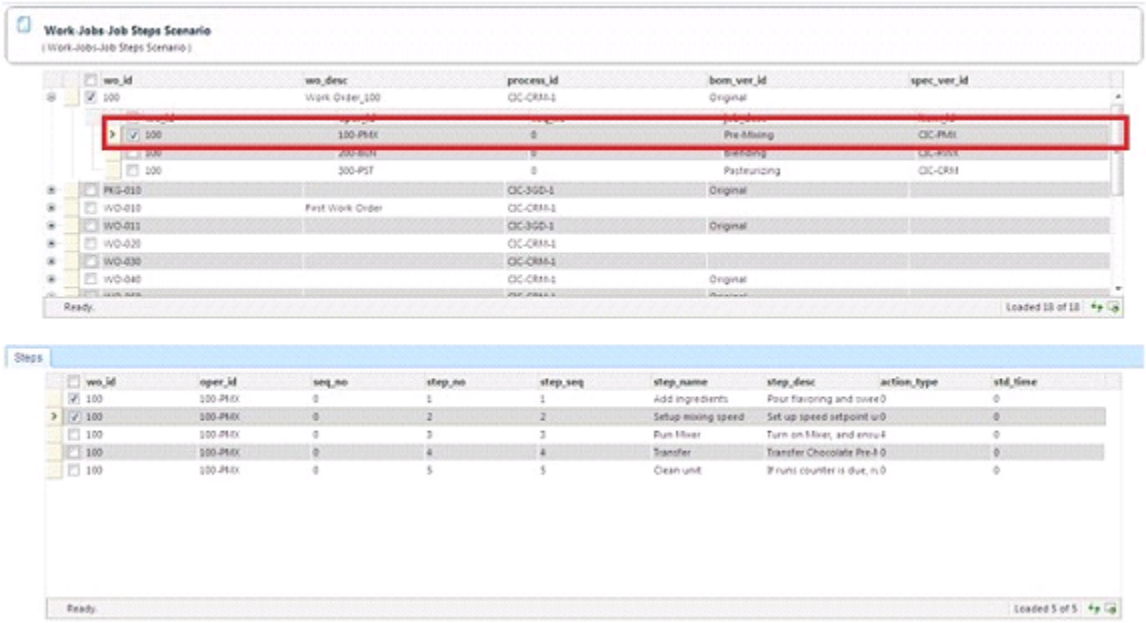


5. Design the workflow as shown below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

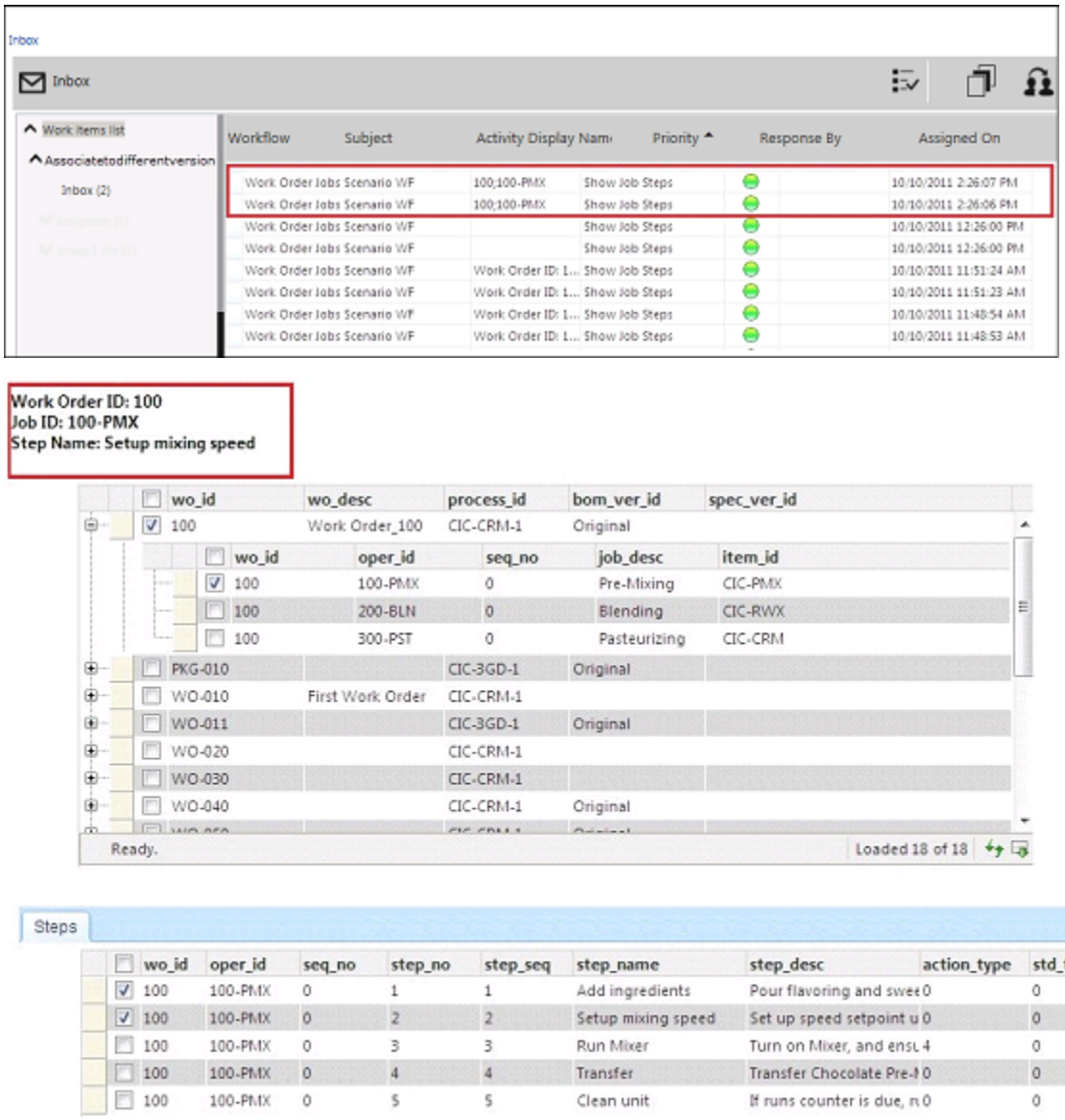


6. Publish and Test Run the workflow.
7. Select rows from Work Order Jobs Grid in the form. The Job Steps Grid gets populated with the steps for the selected job. Select the Steps and then click Finish.



8. Information is sent with Step details of the selected job.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

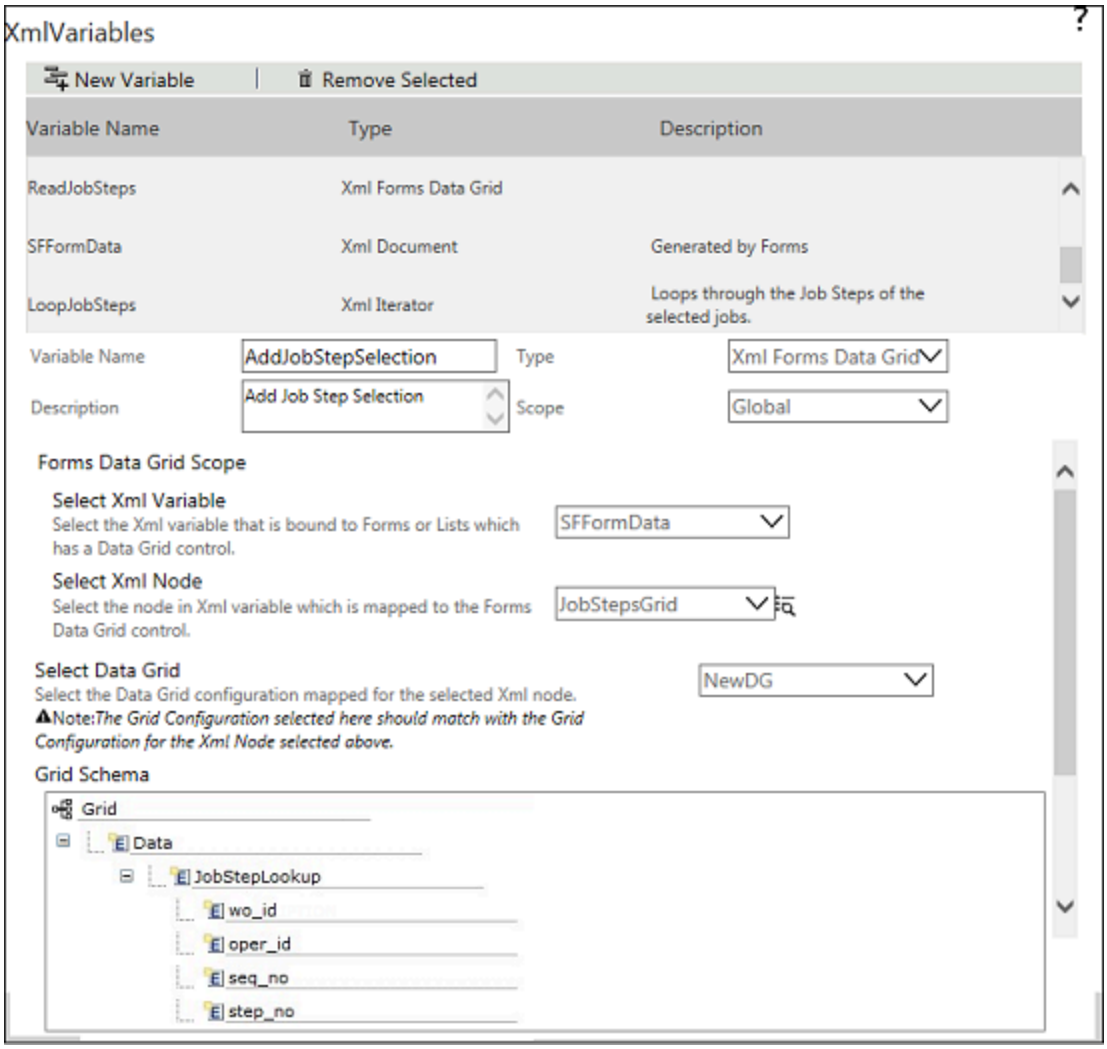


Scenario 2: Use Xml Forms Data Grid type Xml Variable to change/add row selections in a Data Grid control using Workflow.

The Forms Data Grid XML variable can also be used to add/modify the row selections done in a Data Grid control. The below scenario explains how to add a row selection to the existing row selections using workflows.

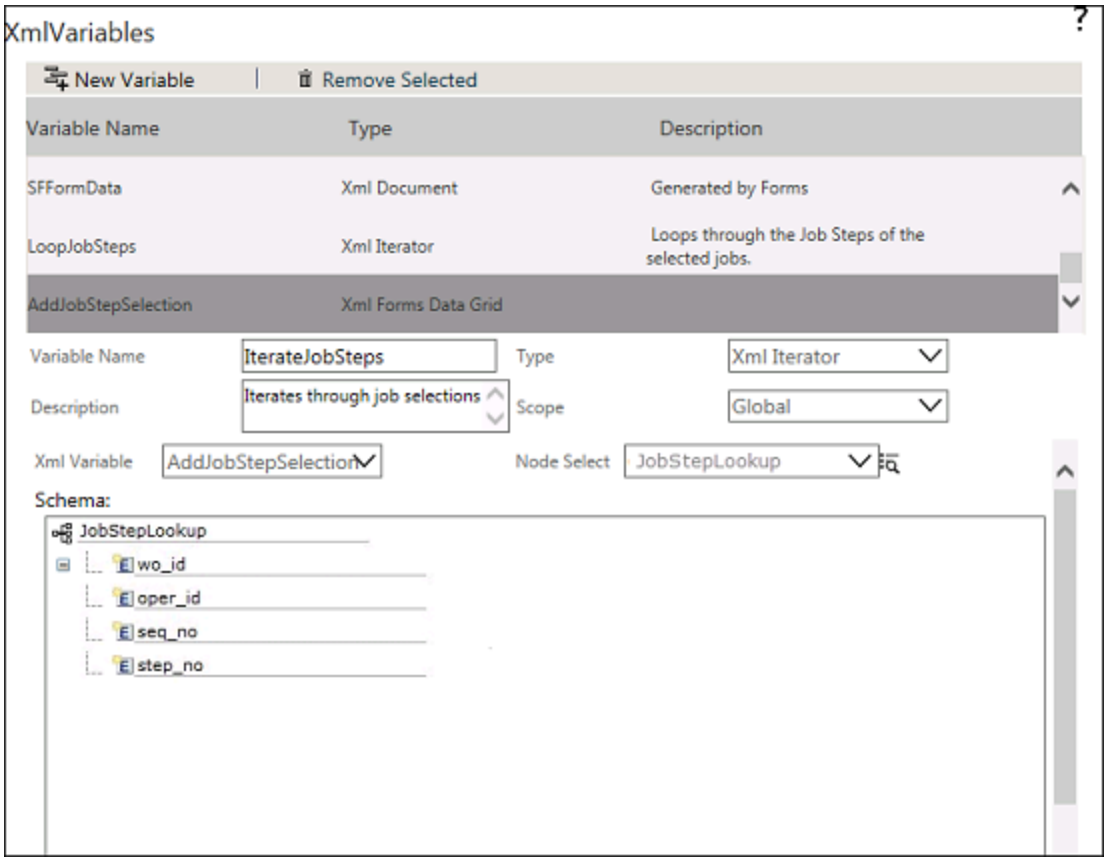
1. Create an XML Forms Data Grid Variable for the Jobs Step Grid Configuration as shown below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

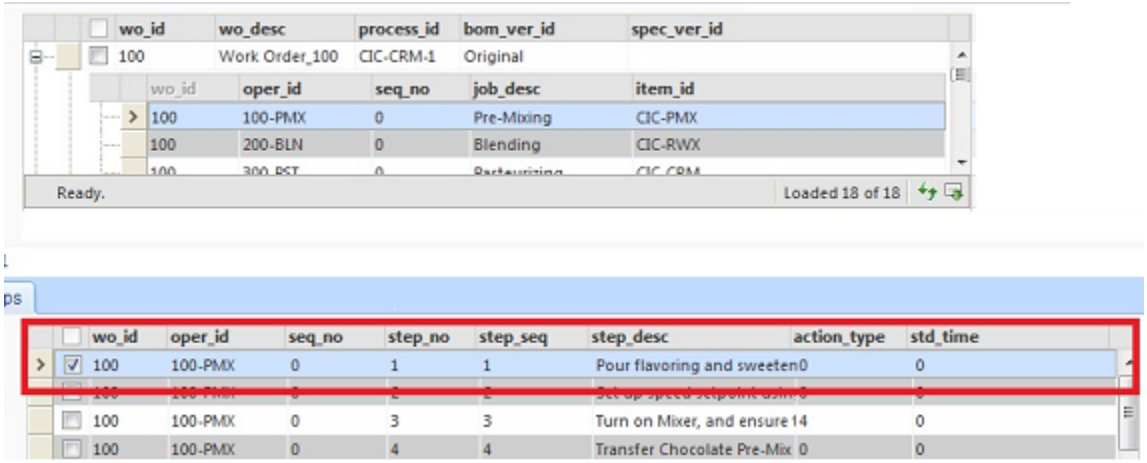


2. Create an XML Iterator variable to iterate through the Jobs Steps for the selected job.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The form instance XML generated for the form with a Job Step selected will be as given below. [However, note that the XML generated will be in the encoded format in the Form instance XML].




```
<Grid>
<Data>
<JobStepsLookup>
<wo_id><![CDATA[100]]></wo_id>
<oper_id><![CDATA[100-PMX]]></oper_id>
<seq_no><![CDATA[0]]></seq_no>
<step_no><![CDATA[1]]></step_no>
<step_seq><![CDATA[1]]></step_seq>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<step_name><![CDATA[Add ingredients]]></step_name>
</JobStepsLookup>
</Data>
</Grid>
```

To add another row selection using workflows, another XML node has to be created for the JobStepsLookup node (which is the repeating node). If another node is added, the XML will be generated as shown below:

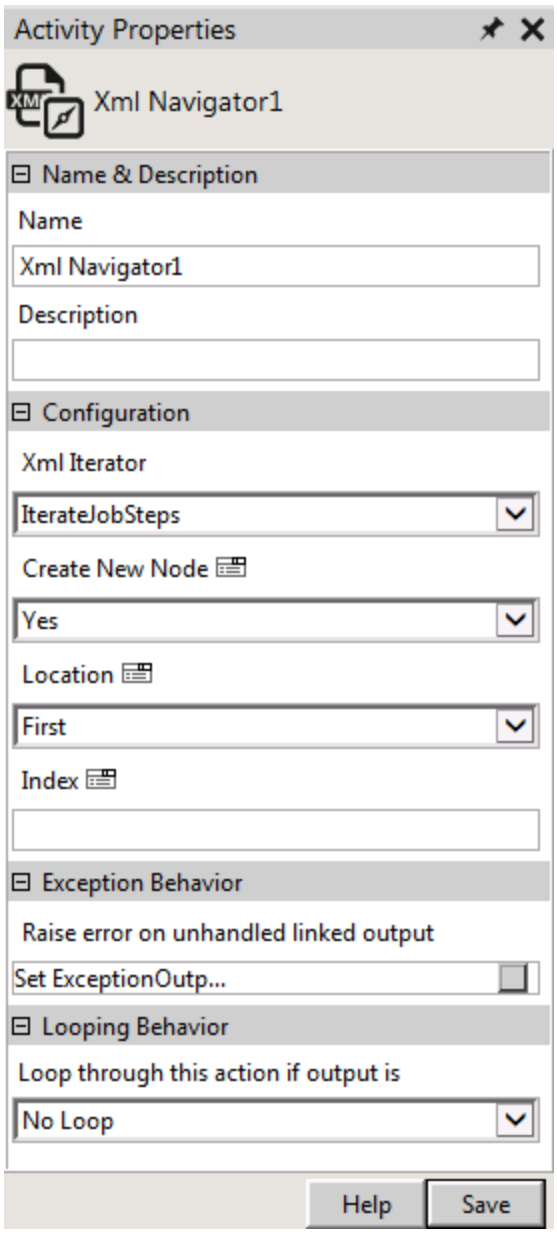
```
<Grid>
<Data>
<JobStepsLookup>
<wo_id><![CDATA[100]]></wo_id>
<oper_id><![CDATA[100-PMX]]></oper_id>
<seq_no><![CDATA[0]]></seq_no>
<step_no><![CDATA[1]]></step_no>
<step_seq><![CDATA[1]]></step_seq>
<step_name><![CDATA[Add ingredients]]></step_name>
</JobStepsLookup>
<JobStepsLookup>
<wo_id><![CDATA[100]]></wo_id>
<oper_id><![CDATA[100-PMX]]></oper_id>
<seq_no><![CDATA[0]]></seq_no>
<step_no><![CDATA[3]]></step_no>
<step_seq><![CDATA[3]]></step_seq>
<step_name><![CDATA[Run Mixer]]></step_name>
</JobStepsLookup>
</Data>
</Grid>
```

wo_id	oper_id	seq_no	step_no	step_seq	step_desc	action_type	std_time
100	100-PMX	0	1	1	Pour flavoring and sweeter	0	0
100	100-PMX	0	2	2	Set up speed setpoint usin	0	0
100	100-PMX	0	3	3	Turn on Mixer, and ensure	4	0
100	100-PMX	0	4	4	Transfer Chocolate Pre-Mix	0	0

To achieve this, XML Navigator activity is being used in this scenario.

1. Add an XML Navigator activity to add a new row selection to the JobSteps grid. Set the properties as shown below. Please note that Create New Node property is set as Yes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

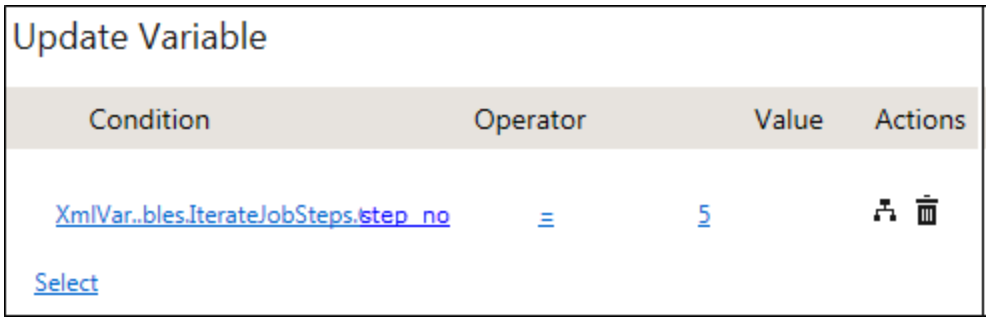


Note: To add selections at multiple hierarchy levels, an Iterator variable has to be created at the parent levels with their corresponding repeating nodes.

2. To set the value for the row selection an Update Variable activity is used. This activity needs to update the value to the Value field of the lookup data source of the Grid Configuration.

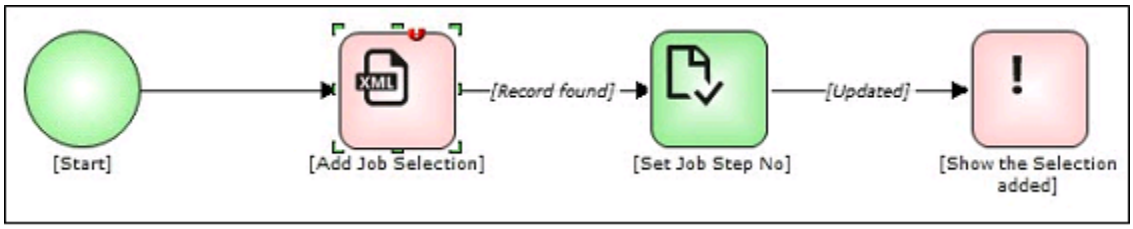
   Here the Value column for the Jobs Step Grid Configuration is: step_no.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

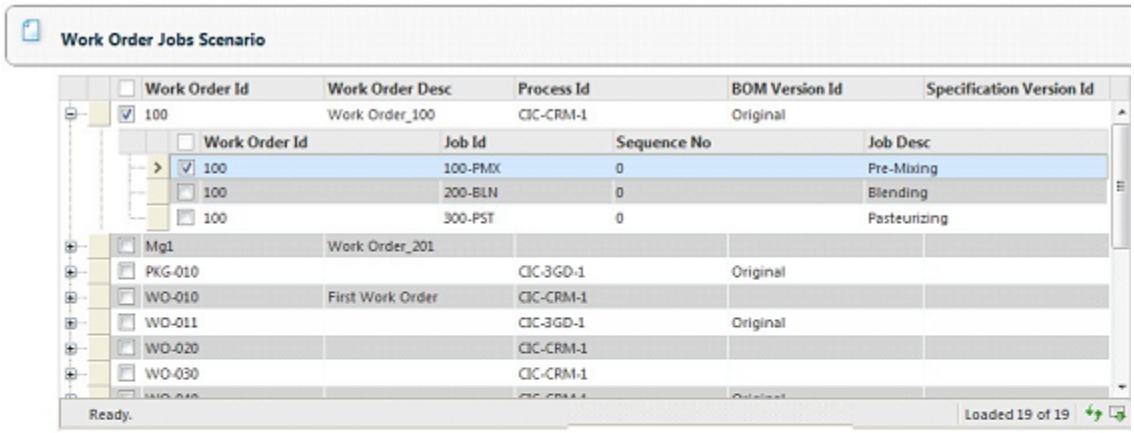


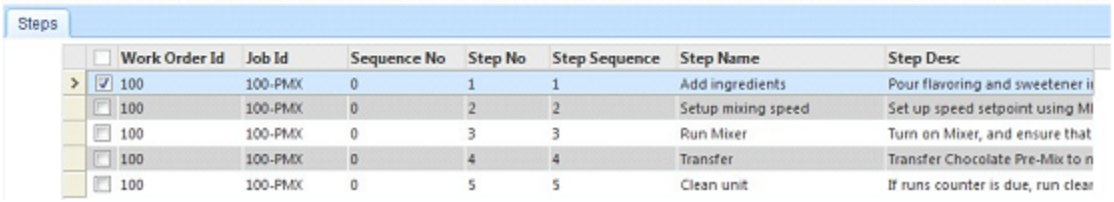
Note: This update does not update or modify any data to the data source lookup configured for the Data Grid Control.

3. Design the workflow as shown below:



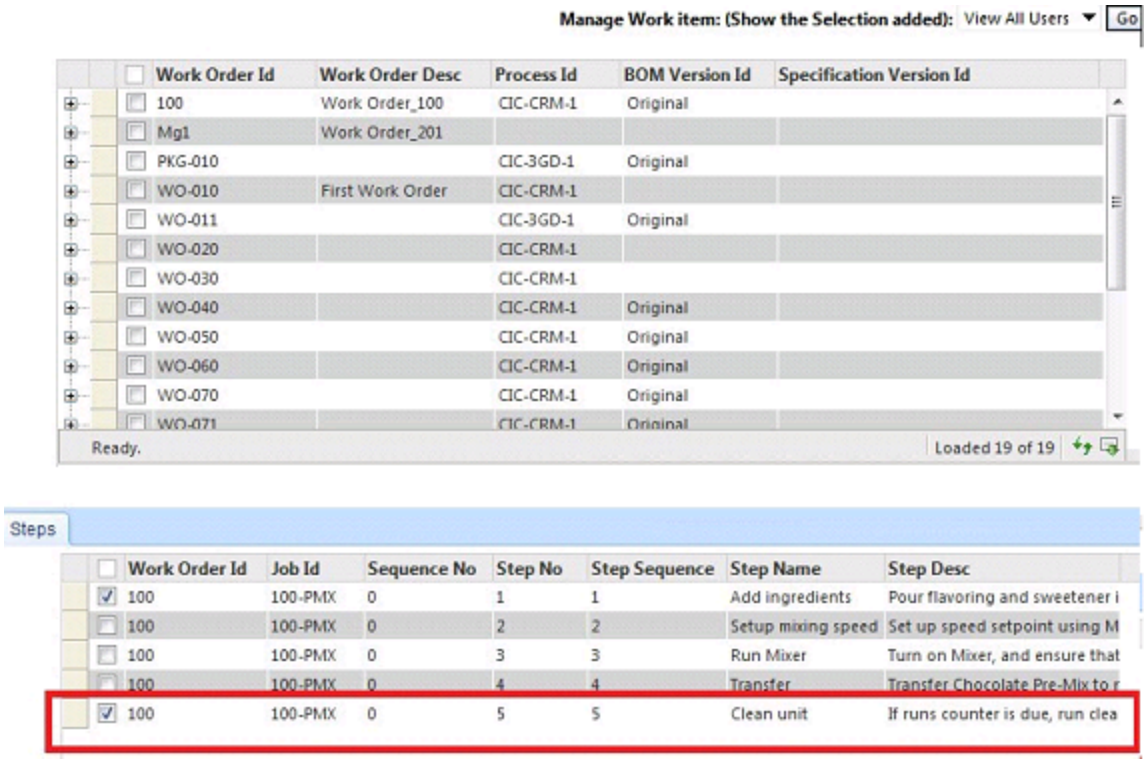
4. Publish and Test Run the workflow.
5. Select rows from Work Order Jobs Grid in the form. The Job Steps Grid gets populated with the steps for the selected job. Select the Steps and then click Finish.





6. Once the workflow execution is complete, information is sent with the updated Job Step selection details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

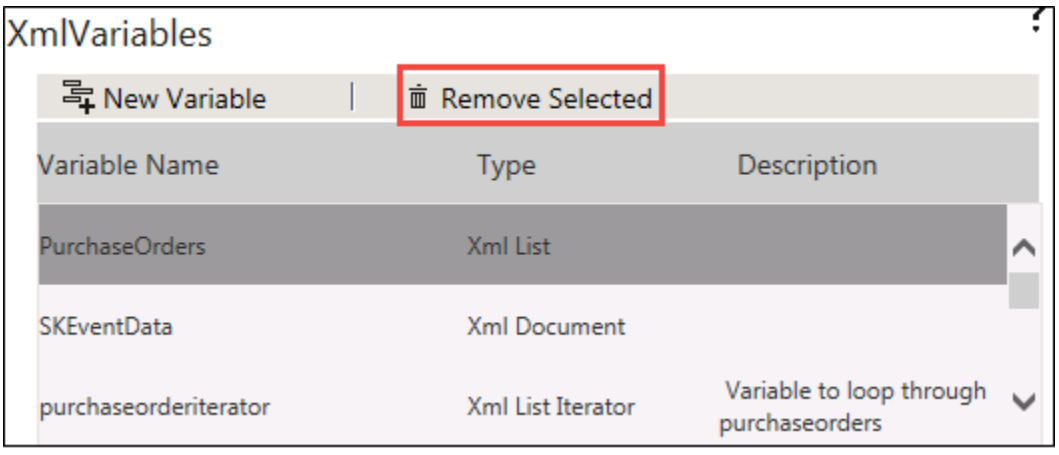


Note: To modify an existing selection, go to the specific node and update the Value field of the lookup data source. If a definite node is not specified for updating the selection, then by default the first selection is modified and updated.

Delete XML Variables

To delete an XML Variable,

- Select the variables to be removed.
- Click **Remove Selected** in the **XMLVariables** window.

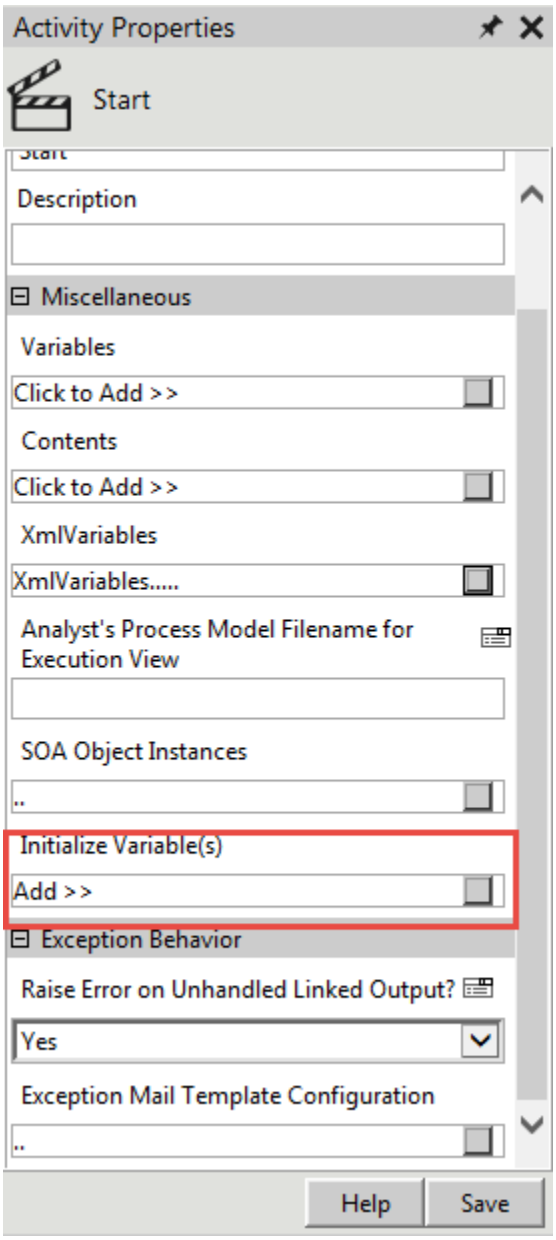


Note: You cannot delete a variable that is the parent of another **XML Iterator** type variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Initialize Variables

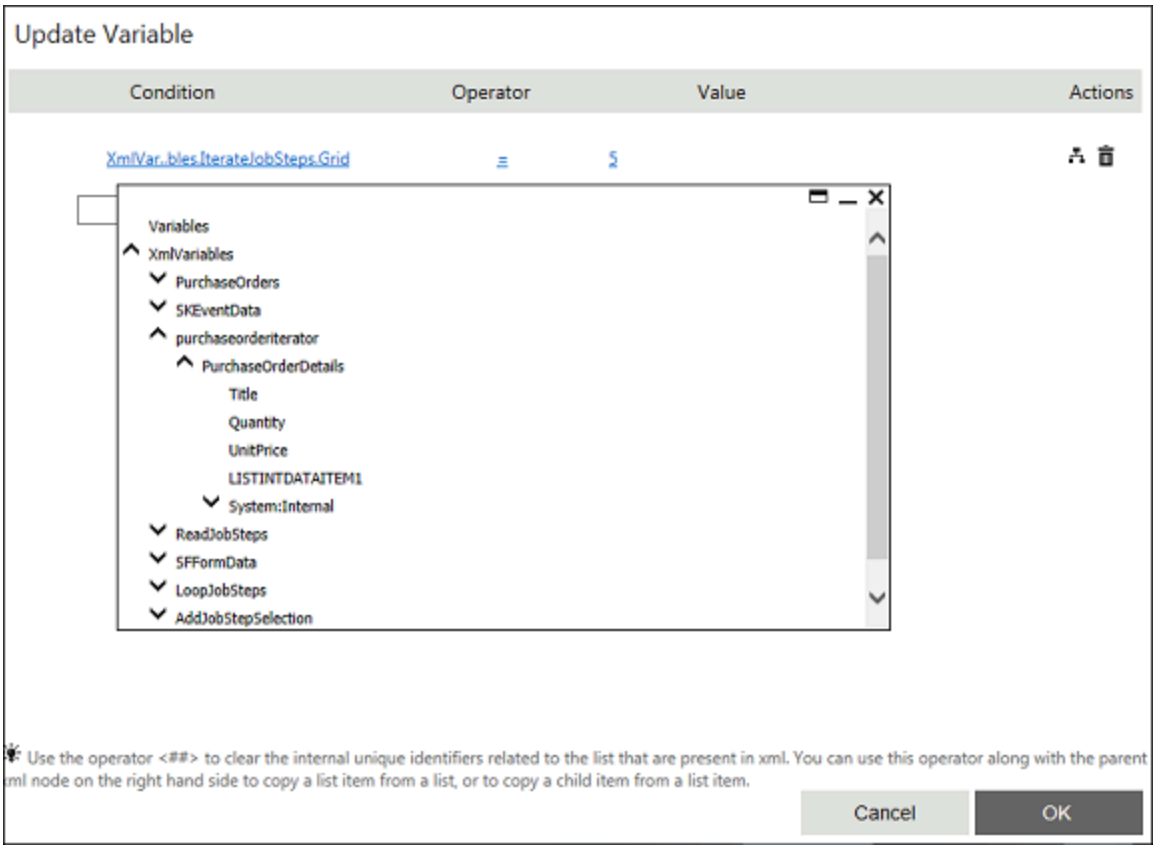
As soon as you create a variable, you have the option to initialize it.



You can update the variable values using the Update Variable activity, if needed. The supported types for initialization are:

- Variables
- Content variables
- XML variables

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Note:
You can also use the Expression Editor to initialize variables.
For number data types, the minimum and maximum values are not validated. Enter only valid numeric values for number data types.
For DateTime, Date, or Time data types, ensure that the date or time input format is based on the culture set. You can enter the input or use the Date and Time picker to enter the required date and time.
Date.CurrentDate is not supported. Use the tree view option to pick the required date and time.
The time format supports long data type to enable you to enter seconds.

Mapped Error Outputs

Each activity has the Raise Error on Unhandled Linked Output property clicking on which, all the default configured mapped error outputs in red with the respective check boxes selected will be displayed.

Mapped Error Outputs

The Mapped Error Outputs list contains the error outputs (shown in red) that cause the workflow to complete abruptly. This can happen if the error outputs are not linked to the next activity in the workflow. To change the mapped error outputs, select or de-select the relevant options and click OK.

Workflow Behavior for Error Outputs

If the error outputs are not linked to the next activity in the workflow, and if the 'Stop with Error' option is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

selected, the workflow will complete with errors. Similarly, if the 'Exit Activity with Error Output' option is selected, the workflow will not finish with errors, but will give an Error Encountered output.

Note: The changes made to this property will take effect only if you set the value for **Raise Error** on **Unhandled Linked Output** to **Yes** in the **Start Activity**.

See *Exception Behavior* in Start Activity topic, for detailed information on configuring **Exception Behavior** properties.

Human Activities

The following activities are found under Human Activities:

- **Approval Activity**: The Approval Action is used in the AVEVA Work Tasks Workflow process design in cases where an approval request has to be approved by an actor or set of actors.
- **Checklist Activity:** The Checklist Activity is incorporated in the AVEVA Work Tasks Workflow process design to send questions that need to be answered by actors in a workflow.
- **Choice Activity**: The Choice Activity is used in the AVEVA Work Tasks workflow process design in cases where a task needs to be assigned with multiple options for an action.
- **Information Activity**: The Information Activity is included in the AVEVA Work Tasks workflow process designs when there is a need to provide some information to a set of actors.
- **Resource Set Activity**: The Resource Set Activity is used in the AVEVA Work Tasks workflow process design when the actor for executing the subsequent activity needs to be selected by another actor in the previous activity of the workflow.
- **Task Activity**: The Task Activity is used in a AVEVA Work Tasks workflow process design in order to assign a task to an actor or set of actors.
- **Work Activity**: The Work Activity is used in the AVEVA Work Tasks workflow process design in cases where custom activities are required.
- **Invoke Form Activity**: The Invoke Form Activity is basically an activity inside workflows, wherein the workflow actually invokes the form.

Approval Activity

Activity Description

The Approval Action is used in the AVEVA Work Tasks Workflow process design in cases where an approval request has to be approved by an actor or set of actors.

Application Scenarios

Typical application scenarios include the Leave Approval Workflow Process, which will have an Approval activity by a Manager, and the Purchase Requisition Workflow, which will have an Approval activity by an Accounts Manager.

Workflow Variables for Approval:

Steps to define variables for use in the Approval activity:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

This section describes how to define variables for use in Approval activities. It gives the syntax for creating the variables and typical examples for different cases.

Variables for capturing WorkItemIds

The Activityids for the activity can be retrieved using a variable of type array, with the following syntax:

'^'+CurrentActivity.Name+'WorkItemIds'

For example, ^ManagerApprovalWorkItemIds

Variables for capturing All Actors' Details

The details of the actor(s) to whom the approval work item gets assigned can be accessed by declaring a variable of type array with the following syntax:

'^'+CurrentActivity.Name+'ActingUsers'

For example, **^ManagerApprovalActingUsers**.

Variables for capturing Responding Actors' Details

The details of the actor submitting the approval/rejection response can be accessed by declaring a variable of type resource with the following syntax:

'^'+HWSActivity.ActivityName+'RespondingUser'

For example, ^ManagerApprovalRespondingUser.

This variable stores the responded user information. In case of multiple users, it will retrieve the last user details.

Note: For single user, you can declare this variable as Resource type. If task is assigned to multiple users, to get all responded users information, then declare this variable as **Array**.

Variables for capturing Approving/Rejecting Actor Details

The details of the actor(s) submitting either approval or rejection response can be accessed by declaring a variable of type Array with the following syntax:

'^'+ activityName+'ApprovingUsers' (or)

'^'+ activityName+'RejectingUsers'

For example, **^ManagerApprovalApprovingUsers or ^ManagerApprovalRejectingUsers**.

Variables for capturing Comments

To capture the comments entered by an actor, add a variable with a name prefixed by '^', with the following syntax:

'^'+_WorkItem.HWSActivity.ActivityName+'CommentApproved'/'CommentRejected'.

For example, if a comment entered along with an approval submission has to be retrieved from an activity named 'ManagerApproval', add a variable with **^ManagerApprovalCommentApproved** as its name. If a comment entered along with a rejection submission has to be retrieved from the same activity, i.e., 'ManagerApproval', add a variable with ^ManagerApprovalCommentRejected as its name. The variable gets initialized with the comment entered by the actor, after the activity response is submitted.

Variables for assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the Start activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To assign multiple queues, specify the queue-ids with semicolon (;) as a separator.
Skelta.HWS.Queue.QueueCollection class can be used to retrieve queue information.

Variable for Queue Acting Users

You can declare an Array type variable to get updated information about the name of the actor to whom an activity gets assigned from a queue.

In the Workflow Variables, define the variable in the following format:

^+CurrentActivityName+'QueueActingUsers'

Ex: ^HRFormQueueActingUsers

HRForm is the activity name.

For example the value of the user who acts on the HRForm activity will be stored in the variable. The Approval1 task will be assigned to the same user.

Variable for storing timeout warning value

Suppose if you want to store the timeout warning value for an Approval activity in your workflow. The naming convention for the system variable to be declared is:

^<ActivityName>ResponseByWarningTimeout

The value returned by the variable will be in UTC format. For multiple user scenarios the highest timeout value calculated for a user will be available in the variable.

Note: Multiple timeout warnings cannot be tracked using this feature.

Every time you input a value for a property, ensure to save the data. Click the **Save** button on the **Activity Properties** window. You need to save the property so that the subsequent properties in the activity can inherit the required information.

Activity Properties

The **Approval** activity has to be configured by specifying appropriate values for the different properties in the Action Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Approval activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Display Name - This property specifies the name to be displayed for the activity in the Web Work Item List. The **maxlength** attribute for type String is added in **Action.XML** and set the default value as **100**. You cannot add more than 100 characters.

If you want to increase the Action name length then you have to manually update it in **Action.XML** file as well as they have to increase the column size in DB.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Action.XML file changes

<property category="cat_name;01" name="Action Display Name" displayname="Display Name" type="string" maxlength="100" helpstring="Display name for the action in activity list."></property>

Database Changes

You need to modify the following tables to increase the column size for action name to more than 100:

1. SWExecutionDetails - Change *ActionName* column.
2. SKHWSActivity - Change *ActivityName column.*
3. *SKWorkItem - Change* ActivityDisplayName column.

Property Type: Optional (This property need not be set if the display name is already configured in the Actions.XML. The name entered here overrides any display name that is configured for this activity in the Actions.XML file.)

Configuration

You can use these properties to configure the functionality of the activity.

Delivery Channels - This property is used to specify the delivery channels to deliver this activity to the assigned actor(s). Multiple channels can be specified giving the actor(s) different options to act on the work item.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Delivery Channel Property

1. To select the Delivery Channels, click on the button for the 'Delivery Channels' property, the Proper window will be displayed.
2. In this window, the Delivery Channels that have been configured are listed on the left along with a check-box against each channel. Click and select the check-box corresponding to the channel you wish to select. You can select multiple channels by selecting the respective check-box.
3. Click on Save Channels button.

See Approval - Delivery Channels for a detailed description of the Delivery Channels property window.

Work Item Fields - This property is used to set the Work Item Fields to be used to display application data as well as value in variables and content in the Work Item list when this activity is executed.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Work Item Fields property

1. To select the Work Item Fields, click on the button for the 'Work Item Fields' property to open the property window.
2. In this window, the Work Item Fields that have been set are listed in a grid. Click on a row to select the corresponding Work Item Field.
3. Right click on the row, and select the "SetWorkItemField" option from the context menu to set it.

See Approval - Work Item Fields for a detailed description of the Work Item Fields property window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Actors

The properties in the Actors group have been described below. You can use these properties to specify the actors who will work on the activity and related properties.

Assign Actor(s) - This property is used to specify the actor(s) who has to take up the Approval activity. It is tagged with the Ignore Unavailable Actors property which in turn is tagged with the Alternate Actor Allowed property. This will be explained in detail later.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any actor(s).)

Steps to set the Assign Actor(s) property

1. To specify the actors, click on the button for the 'Assign Actor(s)' property to open the property window.
2. In this window build an expression to select the actors using the resource properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking the **Add** button. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking the **Undo** button
5. You can also clear an existing expression if necessary by clicking the **Clear** button. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click the **Update** button to save it.

See Approval - Assign Actor(s) for a detailed description of the Assign Actor(s) property window.

See the Using XML Variables section in the About the XML Variable User Interface topic for details on how to use the XMLVariables to build the expression.

Assign Queue(s) - This property specifies the queue(s) to which the Approval activity is to be assigned.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any Queue(s).)

Steps to set the Assign Queue(s) property

1. To specify the queue(s), click on the button for the 'Assign Queue(s)' property to open the property window.
2. In this window select the queue(s) from the **Available Queues** list. You can select multiple queues by selecting all the required queues while having the <Ctrl> key pressed.
3. After selecting the required queue(s) click on the [>] button to move it to the Selected Queues list. You can click on the [>>] button to select all the queue(s) and move them to the **Selected Queues** list.
4. You can remove any queue from the selected list by selecting the required queue and clicking on the [<] button. You can remove all the queues by clicking on the [<<] button.
5. After adding the required queue(s), click the **Assign** button to assign the activity to the selection
6. Click the Close button to close the window without saving the selection.

See Approval - Assign Queue(s) for a detailed description of the Assign Queue(s) property window.

Min. Ownerships [% or Number] - This property, which can be specified as a number or percentage, is the minimum number of actors who should take ownership of the Approval request. This property is related to the 'Timeout warning for Ownership' and 'Timeout for Ownership' properties.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- To illustrate, when 'Timeout for Ownership' is reached, the activity first checks if the ownership count is less than Min.Ownerships (if it is set).
- If it is, the work item will be removed from all Work Item lists.
- If the 'Min.Ownerships' has already been satisfied, then the work item will be removed from the Work Item lists of those who have not taken ownership.
- If the Min.Ownerships value is 0, then the activity checks if the ownership count is less than Max.Ownerships (if it is set).
- If it is, the work item will be removed from all Work Item lists. Else the work item will be removed from the Work Item lists of those who have not taken ownership.
- If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work Item lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Max. Ownerships [% or Number] - This property (referred to as Max.Ack), which can be specified as a number or percentage, is the maximum number of actors who should take ownership of the Approval request.

- If 'Timeout for Ownership' is specified and achieved, and the 'Min.Ownerships' is also satisfied, then the activity waits for action from those who have taken ownership.
- If the Min.Ownerships value is set to 0, then it checks if the ownership count is less than Max.Ownerships. If it is, the work item will be removed from all Work Item lists. Else the work item will be removed from the Work Item lists of those who have not taken ownership.
- If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work Item lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Enable Take Ownership? - If this property is set to **Yes**, then the actor is provided with a **Take Ownership** option in the Work items list to take ownership of the task. If you are not required to take **Take Ownership** of the Work items list, then set this property as **No**.

Note: By default, this property is set as **Yes**.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Enable Take Ownership? property

- Select **Yes** or **No** option from the **Enable Take Ownership?** drop-down list.

Re-Assign Allowed? - If this property is set then the actor is provided with a Re-Assign option. Using this the approver can re-assign the work item to another resource.

Property Type: Optional (This property needs to be set only if necessary.)

Ignore Unavailable Actors? - This property provides the following options - **Yes**, **Yes** - **holidays excluded** and **No**.

- If this property is set to 'No', when the Approval activity is assigned to the actor's Work Item list, a check is first done to see whether the actor is available. If the actor is not available then a check is made for an alternate actor assigned to the unavailable actor. Simultaneously the value of the property 'Alternate Actor Allowed' is checked. If that is also set, then the Approval work item is assigned to this actor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- If this property is set to **Yes - holidays excluded**, the Workflow engine will ignore the fact that actors are not available if the day is an organization holiday and the output Not Enough Resources will not be generated.

Property Type: Optional (This property needs to be set only if necessary.)

Alternate Actor Allowed? - This property is set to assign a particular task to an alternate actor in the absence or unavailability of any actor.

Property Type: Optional (This property needs to be set only if necessary.)

Notification Contents

The properties in the Notification Contents group have been described below. You can use these properties to specify the content used to notify actors about the activity.

Subject - This is the subject of the Approval request. The subject can be customized for each actor.

Property Type: Optional (This property need not be set. If it is not set, the default subject line from the Actions.XML file is displayed.)

Steps to set the Subject property

1. To specify the subject, click the button in the 'Subject' property to open the property window.
2. In this window enter the subject text directly in the text editor.
3. If you want to customize the subject, you can do so using the **Field** controls to add resource property values. For example, to add a subject that includes the email id of the actor submitting the leave application, you can do as follows: First type in the text "Submitted By". Next select "Variable" and "Variable.SubmittedBy.Email" from the two drop downs for **Field** and click **Append**. This will add the placeholder for the variable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the email id.
4. You can edit the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes and zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the subject, click **Update** to save it.
7. Click **Close** to close the window without saving the subject.

See Approval - Subject for a detailed description of the Subject property window.

Body - This is the actual message body of the Approval request sent to the actor(s). The message can be customized for each actor. There is an Enterprise Console Explorer(plug-in) edit icon in the Body property window to get the Work item details URL. See Enterprise Console Control (EC Plug-in) for more information.

Property Type: Optional (This property need not be set.)

Steps to set the Body property

1. To specify the message body, click the button for the 'Body' property to open the property window.
2. In this window, enter the body text directly in the text editor.
3. If you want to customize the body you can do so using the **Field** controls to add resource property values. For example, to add a body that includes the name of the actor submitting the leave application, you can do as follows: Type in the text "Hi, Please approve the leave application submitted by". Select "Variable" and

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

"Variable.SubmittedBy.name" from the two drop downs for Field and click Append. This will add the placeholder for the variable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the name.

4. You can edit the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command and also zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the body, click Update to save it.
7. Click **Close** to close the window without saving the body.

See Approval - Body for a detailed description of the Body property window.

From Email Address? - This property is used to specify the email address from which notification messages will be sent for the activity.

Property Type: Optional (This property needs to be set only if necessary. If it is not set, the default value in Actions.XML will be used.)

Hide Responses? - If this property is set to 'No', then actors can see the responses of other actors in a group approval scenario. To view these responses i.e., ownerships, approvals or rejections, select the 'Take Ownership' option. This will open a new pop up with the responses of other actors who have taken ownership.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Show Custom Document View Window - If this property is set to 'Yes', then the Custom Document View Window will be shown to actors in their Work Item Lists.

Property Type: Optional (This property needs to be set only if necessary.)

Priority - This property is set to indicate the priority of the approval request. Enter the values 0-33 to set as Low, 34-66 for Medium, or 67-99 for High. The default value is set to 34.

Property Type: Optional (This property needs to be set only if necessary.)

Send Notification Email - If this property is set to 'Yes', then notification emails will be sent to the actors. If it is set to 'Yes-HTML', then the emails will be sent in HTML format.

Property Type: Optional (This property needs to be set only if necessary.)

Notify if alternate actor is considered - If this property is set to 'Yes', then notification emails will be sent to the actors when an alternate actor is considered.

Property Type: Optional (This property needs to be set only if necessary.)

Redirect URL - This property is used to get the URL of the web page to redirect the client to, in the case of synchronous execution of the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Redirect URL property

1. To specify the Redirect URL, click on the button for the 'Redirect URL' property to open the property window.
2. In this window, enter the Redirect URL directly in the text editor.
3. If you want to customize the Redirect URL you can do so using the Field controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for Field, and

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

clicking **Append**. This will add the placeholder for the variable in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the URL.

4. You can edit the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command and also zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Redirect URL, click **Update** to save it.
7. Click **Close** to close the window without saving the Redirect URL.

See Approval - Redirect URL for a detailed description of the Redirect URL property window.

Custom Document View URL - This property is used to get the URL of the page to which the custom document dlls does a post. The data posted to the page are application, workflow, executionid, executiondetailsid, and workitemid. The posted data can be retrieved as Request.Form["workitemid"].

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Custom Document View URL property

1. To specify the Custom Document View URL, click the button for the 'Custom Document View URL' property to open the property window.
2. In this window, enter the Custom Document View URL directly in the text editor.
3. If you want to customize the Custom Document View URL you can do so using the Field controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for **Field**, and clicking **Append**. This will add the placeholder for the variable in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the URL.
4. You can edit the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command and also zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Custom Document View URL, click **Update** to save it.
7. Click **Close** to close the window without saving the Custom Document View URL.

See Approval - Custom Document View URL for a detailed description of the Custom Document View URL property window.

Mail Channel

The properties in the Mail Channel group have been described below. You can use these properties to customize the approval activity(activities) with different mail channel templates.

Use html template from property - If this property is set to **'No'**, then it will take the mail channel template from the physical path

'[AVEVA Work Tasks Installed Path]\WORKFLOW MANAGEMENT.NET\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\AprovalEmail.html.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If this property is set to **'Yes'** and the "**HTML Template**" property is empty, then it will take the mail channel template from the physical path

'[AVEVA Work Tasks Installed Path]\BPM.NET\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\AprovalEmail.html.

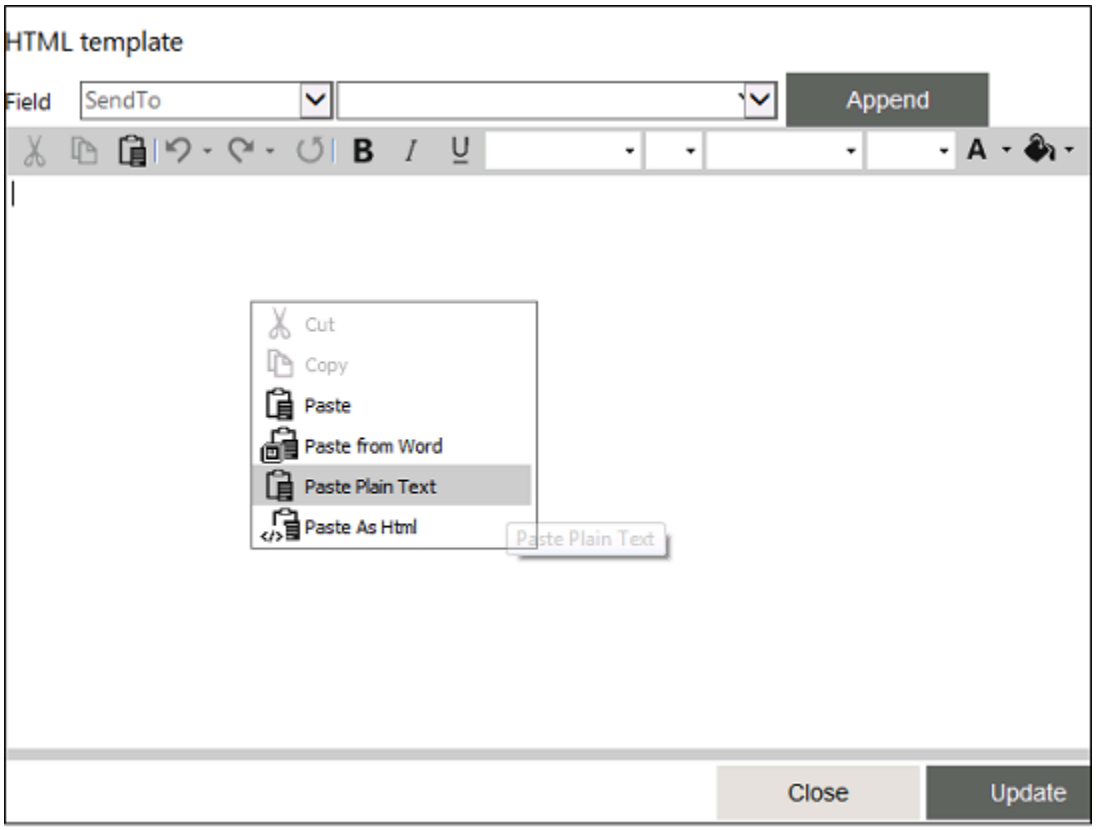
Property Type: Optional (This property needs to be set only if necessary.)

HTML template - This property is used to define the html template for mail channel.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the HTML template

1. To specify the HTML template, click the edit icon to open the property window.
2. In the **HTML template** window, provide template which should be in html format. If you are pasting a copied html content, then right-click the **HTML template** window and paste it as **"Paste Plain Text"**.



3. Click **Update**.

See Approval - HTML Template template for a detailed description of the property HTML template window.

Attachment

This property is used to configure Report to be sent as an Email attachment.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Attachment property

1. To specify the **Attachment** property, click the edit icon to open the property window.
2. In the **Configuration of Mail Attachment** window, select the required attachment type by clicking the radio

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

button.

3. In the **Attach Report** property, you can attach a report by selecting the **Report Reference Variable** through the Expression Editor. Alternatively, you can get the **Report** by clicking the Report icon and selecting the required report from the Reports list screen. To attach multiple reports, add new rows using the **Add** link.
4. Click **Submit**.

See Approval - Attach Reports for a detailed description of this property.

Decision Influencers

The properties in the Decision Influencers group have been described below. You can use these properties to specify some parameters that influence the outcome of the activity.

Affirmative Answer for Approval [% or Number] - This property is used to check what percentage or number of persons should take action for approval. Decision on the approval is taken only after the specified number or percentage of actor(s) have taken the action. The work item waits for the actors' action until the 'Timeout for activity' is reached.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Action Limit [% or Number] - This property specifies the minimum number of approvers to act on the Approval before the 'Timeout Warning for Action'. If the activity limit is reached before the 'Timeout Warning for Action' then the warning message is not shown.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Wait For All The Participants To Act -

- If this property is set to 'Yes', then the engine will wait for all the approvers to act on the approval activity. Once the activity count (no. of approvals and rejections) reaches the 'Max.Ownerships' limit value, a decision for approval is taken if the 'decision count' (no. of approvals) is >= 'Affirmative Answer for Approval'. Otherwise, the activity is rejected.
- If both 'Affirmative Answer for Approval' and 'Action Limit' are set to 0, the activity is considered 'approved' only if 'decision count' >= 'rejection count'.
- If this property is set to 'No', the engine will take the decision for the activity based on the values for 'Affirmative Answer for Approval(AAA)', 'Max.Ownerships', 'Min.Ownerships' and 'Action Limit (AL)'
  - If AAA is set and 'decision count' >= AAA, the activity is approved.
  - If AAA is set and 'decision count' < AAA and Max.Acks = AAA and if at least one rejected, the activity is considered rejected.
  - If AAA = 0, and AL is set, if Max.Acks = AL, and if at least one rejected, the activity is rejected.
  - If AAA = 0, and AL is set, and if 'decision count' >= AL, the activity is approved.
  - If AAA is set and 'rejection count' > AAA/2, the activity is rejected.
  - If AAA = 0 and AL is set and 'rejection count' > AL/2, the activity is rejected.
  - If 'activity count' = Max.Ownerships, a decision for approval is taken if the 'decision count' >= AAA. Otherwise, the activity is rejected.
  - If both AAA and AL are set to 0, the activity is considered 'approved' only if 'decision count' >= 'rejection count'.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Escalation & Timeout

The properties in the Escalation & Timeout group have been described below. You can use these properties to specify escalation and timeout settings for the activity.

Set Calendar for Timeout - This property is used to specify the calendar to be used to determine activity timeouts.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to Set Calendar for Timeout

1. To specify the calendar for timeout, click the button for the 'Set Calendar for Timeout' property to open the property window.
2. In this window, select the required calendar by clicking the calendar name in the tree structure.
3. Click **Set Calendar** to set the selected calendar.
4. You can specify that a resource calendar should be used, if available, by clicking to select the corresponding check box.

See Approval - Set Calendar for Timeout for a detailed description of the Set Calendar for Timeout property window.

Minimum available time for Activity - This property is used to check the availability of an actor to whom a task must be assigned while executing the activity. The availability check for the actor is done based on the Calendar settings (Global or Resource Calendar). If the time taken to complete the task is within the value set for the **Minimum available time for Activity** property, then the task will be assigned to that actor. However, if the time taken to complete the task exceeds the value set for the **Minimum available time for Activity** property, then the task will be assigned based on the values set for the **Ignore Unavailable Actors?** and **Alternate Actor Allowed?** properties.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout for Ownership - This property specifies the time until which actors are allowed to take ownership, i.e. the time until which the Approval activity will wait for actors to take ownership.

- If Min.Ownerships is set and is not reached, the work item is removed from all Work Item lists.
- If Min.Ownerships is 0, it will check if ownership count is less than Max.Ownerships. If it is, the work item is removed from all Work Item lists. Else the work item will be removed from the Work Item lists of those who have not taken ownership. The work item will remain in the Work Item list of those who have taken ownership and wait for action from them.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout Warning for Ownership - This property specifies the time at which a warning is provided to an actor to take ownership of the work item assigned to him/her. If the actor does not take ownership within this time the engine is notified with a warning.

- Thus if 'Min.Ownerships' is set, and it is not satisfied before the given time then a warning is sent.
- If Min.Ownerships is 0, it will check for Max.Ownerships. If the ownership count does not reach the Max.Ownerships value within the set time, a warning is sent.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- If both Min.Ownerships and Max.Ownerships are not set, this property is ignored.

Property Type: Optional (This property needs to be set only if necessary.)

Understand how Ownership Properties relate to each other:

- Approval activity is assigned to the approvers in the approval list by the Workflow Engine. Approval activity has a default property set that at least one person has to make to Take Ownership of the Approval activity.
- A drop down interface has been provided for the actor to Take Ownership. However, the actor can act on the Approval without taking ownership first, in which case ownership will be considered to be done, i.e. if the actor acts upon the Approval activity without acknowledging, then it is considered as an ownership with action taken.
- After the actor has taken ownership, a link is provided to the actor, which opens a pop up with the details of the other approvers. This is again tagged with the 'Hide Responses?' property. The link will be provided only If 'Hide Responses?' is set as 'No'. The pop up will have details like Approver name, Ownership Status, Action taken, Comments.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number], then the 'Timeout Warning For Ownership' property is not considered.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] and the 'Time Out For Ownership' is also reached, then the work item is removed from other approvers' Work Item list and the work item waits for action from those who have taken ownership.
- If the 'Min.Ownerships' limit is not reached for the Approval activity, and the 'Timeout Warning For Ownership' has been exceeded then the actor is notified with a warning by including an email or an information activity, with the Timeout Warning for Ownership link to it.
- If 'Timeout For Ownership' is set, if the Approval activity has not reached the minimum limit (Min.Ownerships) and the 'Time Out For Ownership' has been reached, then the work item is removed from the Work Item lists of all the approvers.
- If 'Min.Ownerships' = 0 and if 'Max.Ownerships' is set, the activity waits for actors to take ownership or action until the 'Time Out For Ownership' is reached (if this is set).
- If none of the ownership properties are set then the work item waits until the actor takes ownership or acts upon it.

Timeout Warning for Activity - This property specifies when a reminder has to be provided for the approvers to act on the Approval request, if the activity count has not been reached. If properties such as Max.Ownerships, Affirmative Answer for Approval (AAA), and Action Limit (AL) are set, once the property values are reached then a warning is sent to the actor. When the timeout is reached, the following checks are made: If AL is set, when the activity count i.e. the no. of approvals or rejections (AC) < AL, then a warning is sent; if AL = 0 and AAA is set, and when AC < AAA, then a warning is sent to the participant.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Timeout for Activity - This property's value is used to decide until what time the work item should wait for user action.

If properties such as Max.Ownerships, AAA, AL are set, once the property values are reached the work item is removed from the Work Item lists of all the approvers.

- When the timeout is reached, the following checks are made:
  - If AL is set, when the activity count i.e. the no. of approvals or rejections (AC) < AL, then the timeout for activity is performed and the activity is removed from all Work Item lists.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- If AC > AL, it checks with the no. of approvals or 'decision count' (DC) and AAA. If DC >= AAA (if it is set), the work item is considered approved; otherwise, it is rejected.
- If AAA = 0 and if DC >= AL, the work item is considered approved; otherwise, it is rejected.
- If AL = 0 and AC < AAA, then the timeout for activity is performed and the work item is removed from all Work Item lists. If AL = 0 and AC > AAA, it will check with DC.
- If DC > AAA it is approved; otherwise, it is rejected. If AL = 0 and AAA = 0, then if DC is at least 1, then it is considered approved; otherwise, it is rejected.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Multiple TimeOut Warning - This is used to set recurring timeout warnings that will be sent to the actors who have to take action. To set multiple timeout warnings, click the 'Multiple TimeOut Warning' button in the Properties pane for the Approval activity. A new window pops up. In this window, set the interval for the recurring warning as well as the output message. You can also set the Recurring Time Interval. This can be used to set a different recurring time out after the first timeout happens. For example, if 'Time Interval' for the property is set as 7 days and 'Recurring Time Interval' is set as 2 days, then the first time out will happen after 7 days and there will be a recurring time out every second day after the seventh day (first) time out.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Steps to set the Multiple TimeOut Warning property

1. To set the multiple timeout warning, click the button for the 'Multiple TimeOut Warning' property to open the property window.
2. Enter the time interval for the timeout in the **Time Interval** field
3. Select "Yes" from the drop down for the **Recurring**? field if you want the timeout warning to recur. Select "No" otherwise.
4. Enter the output message corresponding to the timeout warning in the **Output** field.
5. Enter the time interval for the recurring timeout in the **Recurring Time Interval** field.
6. After entering the content details, click **Add Row** to add this timeout setting and display a new row for entering another timeout setting.
7. Remove rows if necessary by selecting them and clicking **Remove Selected**.
8. After specifying the timeout settings, click **Update**.

See Approval - Multiple Timeout Warning for a detailed description of the Multiple TimeOut Warning property window.

Miscellaneous Top

Redirection Enabled? - This property enables redirection. When this property is set to Yes, the workflow redirects the user to the URL in the Redirect URL property in the next activity.

Activity outputs:

The Approval Activity returns the following values -

Approved: Displayed when the request is approved.

Rejected: Displayed when the request is rejected.

Not Enough Resources to Acknowledge: Displayed when the filter condition did not yield enough actors to whom the request can be sent.

Timeout - Acknowledge: Displayed when the specified minimum number of actors did not take ownership of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

request in the given time.

Timeout - Action: Displayed when some or all of the specified actors did not perform the required activity in the given time.

Timeout Warning - Acknowledge: Displayed when a warning is generated that the specified minimum actors have not taken ownership of the request in the given time.

Timeout Warning - Action: Displayed when a warning for timeout is generated because some or all the actors did not perform the required activity in the given time.

Note: The following outputs will return only if you assign tasks to a queue user.

Resource Unavailable for Queue - If a queue is assigned to a task and if there are no resources available in the queue then this output will get triggered.

TimeOut for Queue Task - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting the completion of that task.

Timeout Warning -Queue - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting in sending a warning out put. However, the task will still be sleeping.

Timeout Warning -Queue Participant - This output will get triggered when the task in the queue is picked by an user but still not acted on that task within the queue escalation time out settings resulting in triggering a warning output. However, the task will still be sleeping.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Looping Behavior

You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Approval - Delivery Channels

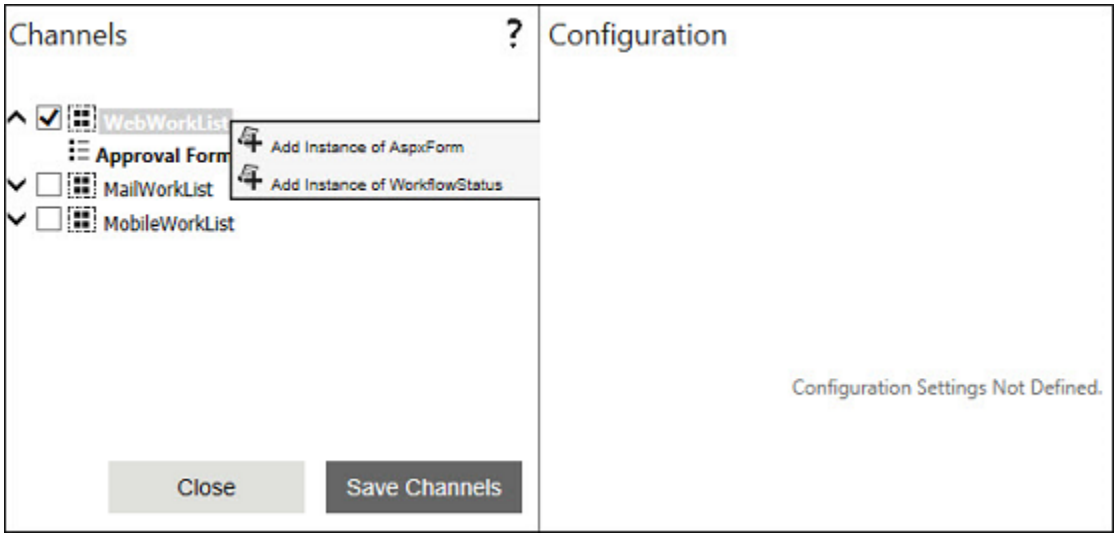
Purpose

The purpose of this chapter is to describe the feature of WorkflowStatus Form of Delivery Channels property under the category Configuration of any human activity available in AVEVA Work Tasks.

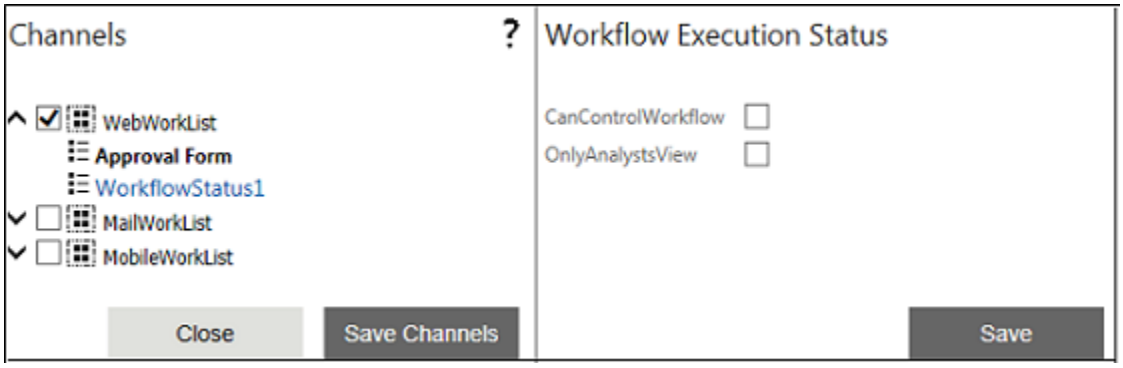
Description: This feature enables us to view the status of the workflow in the work item list in the graphical form. **[Optional]**

1. Select the **Approval Activity**.
2. Click the **Delivery Channels Property**.
3. This will open a window like the one shown below.

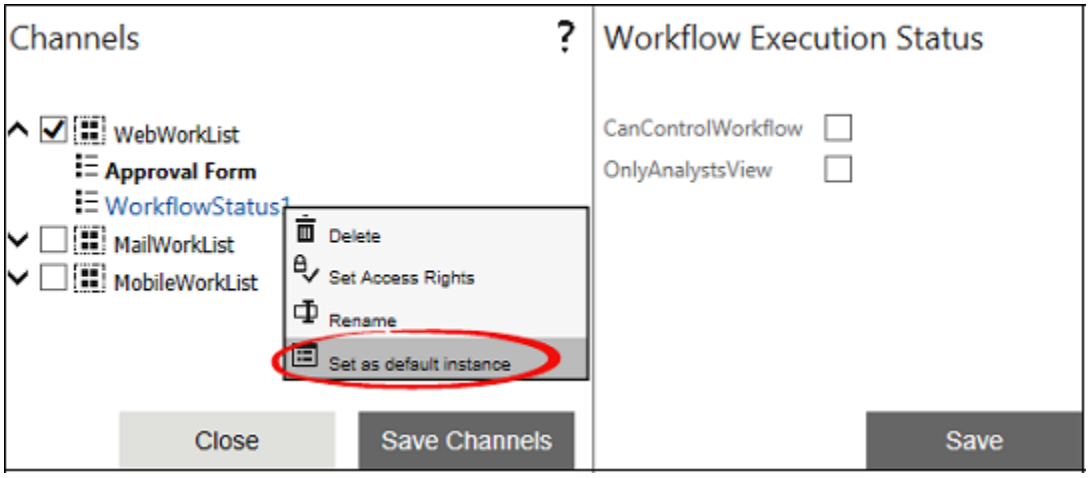
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



4. Right-click the WebWorkList to display the context menu with the Add Instance of Workflowstatus option as shown below.



5. Set the Properties for the Execution View Status Graphical view. Click **Save**.
6. If you want to Set the workflowstatus form as the default instance you can do it by right-clicking workflowstatus1 and selecting the **Set as default instance** option from the context menu.



7. Click the **Save** and **Save Channels** buttons.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Output

Now, when you click on any row in the Work Item List, you will find this Execution view page being displayed as the detailed view, if you set it as a default instance.

Otherwise Workflow status form will appear in context menu of each row in Work items list and you can open the execution view by selecting it from the context menu.

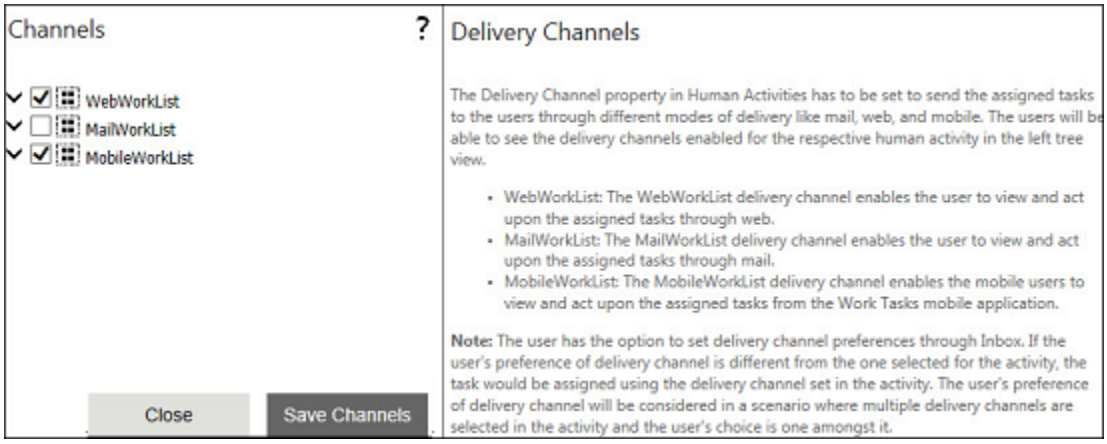
Delivery Channels - MobileWorkList

To deliver the work items to the mobile device, the **MobileWorkList** option in the Delivery Channel property needs to be selected. This allows mobile users to view and act upon the tasks from the Inbox that is rendered on their mobile devices.

This topic describes the steps required to configure the **MobileWorkList** option in the **Delivery Channel** property of the Approval activity.

To configure 'MobileWorkList' option in the Delivery Channels Property

1. In the **Process Designer**, drag and drop the **Approval** activity on to the design area. The activity properties pane of the selected activity appears on the right side of the design area.
2. In the **Activity Properties** pane, expand the **Configuration** node and select the **Delivery Channels** property.
3. In the **Delivery Channels** window, select the **MobileWorkList** check box in the **Channels** section.



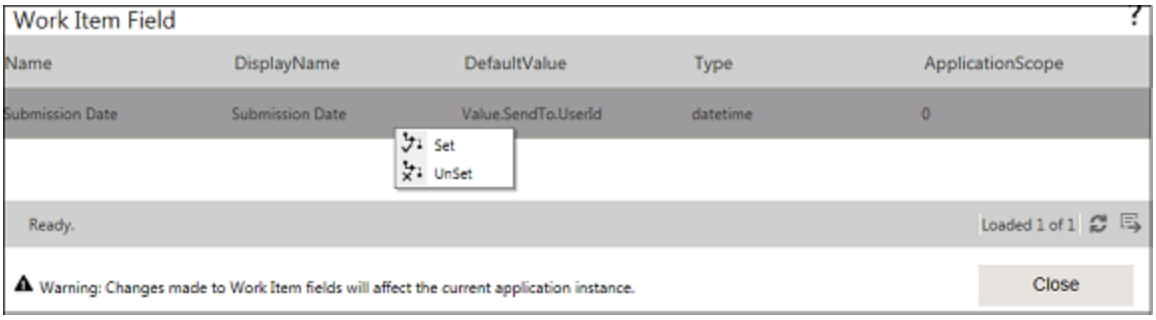
4. Click **Save Channels** and **Close**.
5. Set all the other properties and click **Publish and Test Run**.

Approval - Work Item Fields

Work Item Fields Property Window

You can set the Work Item Fields for this activity in this property window. To open this window, click the button for the 'Work Item Field' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a grid listing all the Work Item Fields that have been set for the process.

- Each row in the grid represents a Work Item Field. You can click a Work Item Field to select it.
- The context menu has the **Set** option, which you can use to set the selected Work Item Field for this activity. It also has the **UnSet** option to remove the Work Item Field setting.
- **Close** - You can use this button to close the window.

Note: While entering a value in the Work Item field, it is recommended to keep it less than the specified database column size. If there is a need to increase the size of the Work Item field, the user should manually increase the column size in the database.

Approval - Assign Actor(s)

Assign Actor(s) Property Window

You can set the actors to whom the approval work item is to be sent in this property window. To open this window, click the button for the 'Assign Actor(s)' property.

See Assigning a Task to a Manager for more details.

The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Different Methods of Assigning an Actor

- **Assigning an Actor by building an expression:**

Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

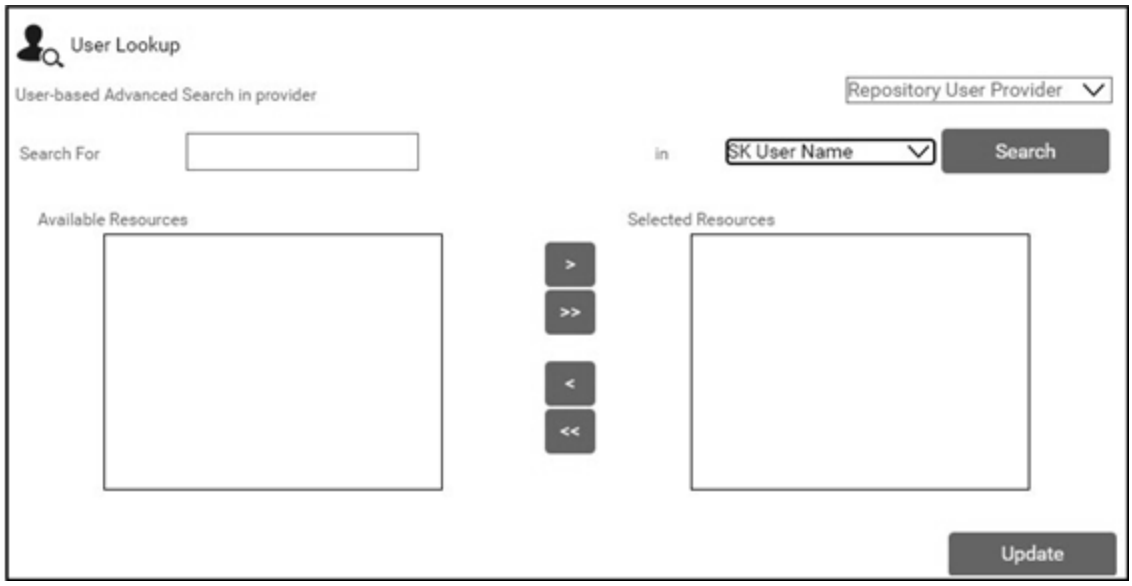
- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

- **Assigning an Actor through User Lookup**

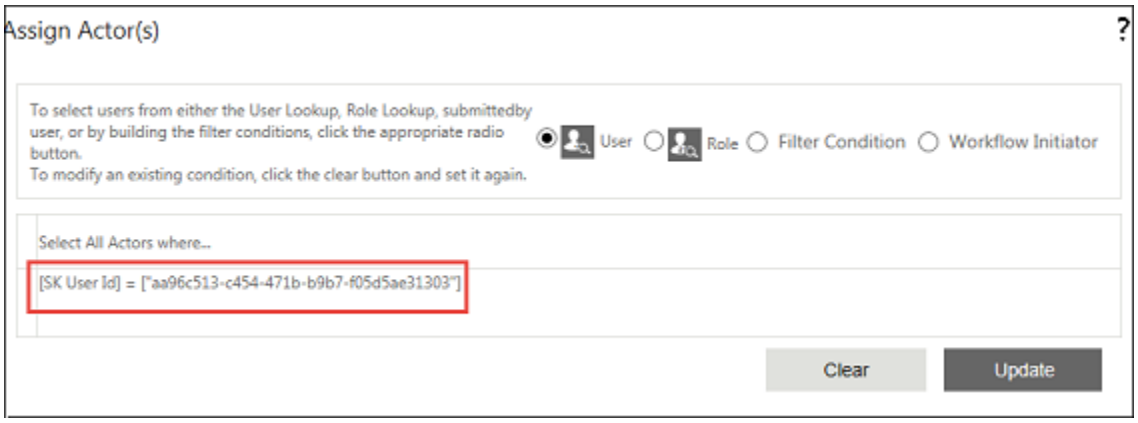
Through the **User Lookup** you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.
2. The **User Lookup** dialog is displayed, search and select the required actor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:



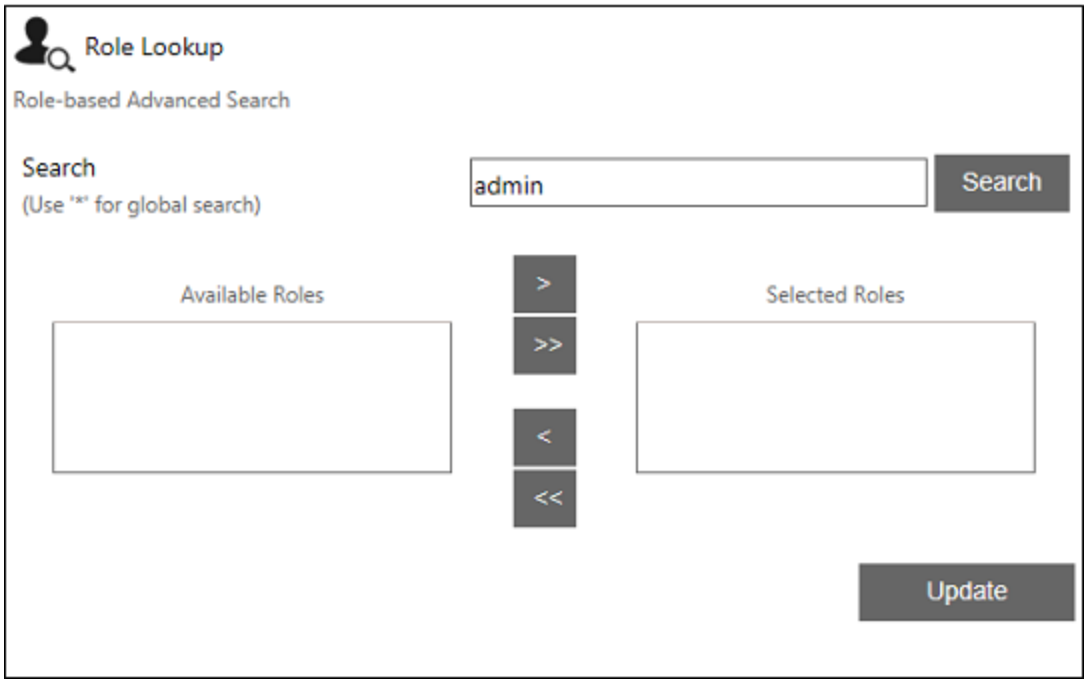
4. Click **Update** in the Assign Actor(s) window to update the selected actor(s) in the **Assign Actor** property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

- **Assigning an Actor through Role Lookup**

Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

1. Select the **Role Lookup** option.
2. The **Role Lookup** dialog is displayed, search and select for the required role.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



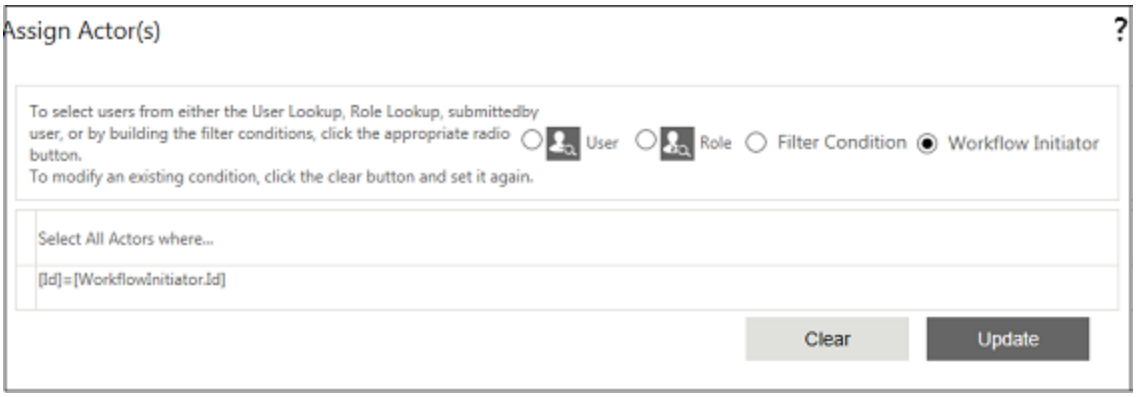
3. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

4. Click **Update** in the Assign Actor(s) window to update the Assign Actor property.

- **Assigning an Actor through Workflow Initiator**

Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

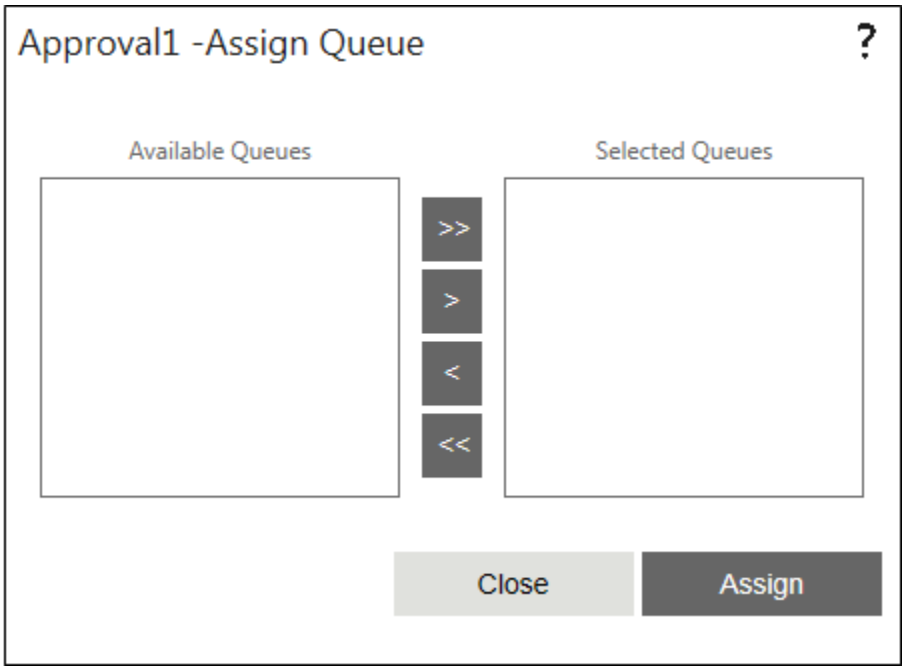
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Approval - Assign Queue(s)

Assign Queue(s) Property Window

You can select the queues to which the approval work item is to be sent in this property window. To open this window, click the button for the **'Assign Queue(s)' property**.



The window has a pair of list boxes to specify the queue(s). The **Available Queues** list shows all the available queues. You can select the required queue(s) from this list box. The **Selected Queues** list shows all the selected queues to which the activity will be assigned.

You can use the following buttons located between the list boxes to move the queues to and from the list boxes.

- [>>] - You can click this button to add all the queues in the Available Queues list to the Selected Queues list.
- [>] - You can click this button to add the selected queue(s) to the Selected Queues list.
- [<] - You can click this button to add the selected queue(s) from the Selected Queues list to the Available Queues list.
- [<<] - You can click this button to add all the queues in the Selected Queues list to the Available Queues list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

At the bottom of the window you will find the following buttons:

- **Assign** - You can click this button to assign the activity to the selected queues.
- **Close** - You can click this button to close this window.

Assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

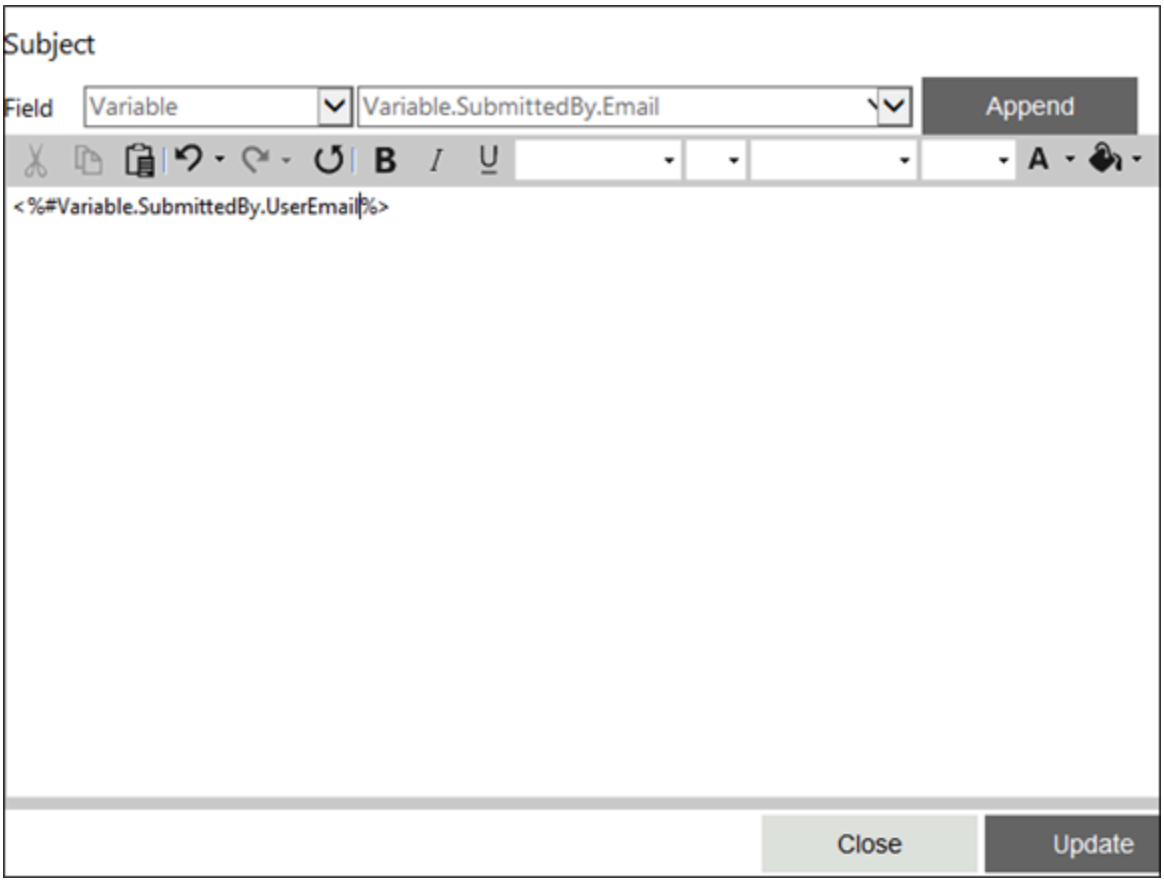
For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the **Start** activity with the Type as **String**. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

To assign multiple queues, specify the queue-ids with semicolon (;) separator. **Skelta.HWS.Queue.QueueCollection** class can be used to retrieve queue information.

Approval - Subject

Subject Property Window

You can specify a subject for the Approval work item assignment in this property window. To open this window, click the button for the 'Subject' property.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The window has a text editor to enter the subject. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the subject. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the subject in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the subject.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the subject using a combination of discrete value and variable or content. For example, in the image above, the subject entered is "Submitted By <%#Variable.SubmittedBy.Email%>". Here, the string "Submitted By" has been typed directly in the text editor. The second half of the subject is a placeholder, which will be replaced by the variable value during run-time. This has been added by selecting the appropriate variable option from the **Field** drop down and clicking **Append**.

The text tool bar above the text editor allows you to manage and format the subject. It has the following text tools:

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools:

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

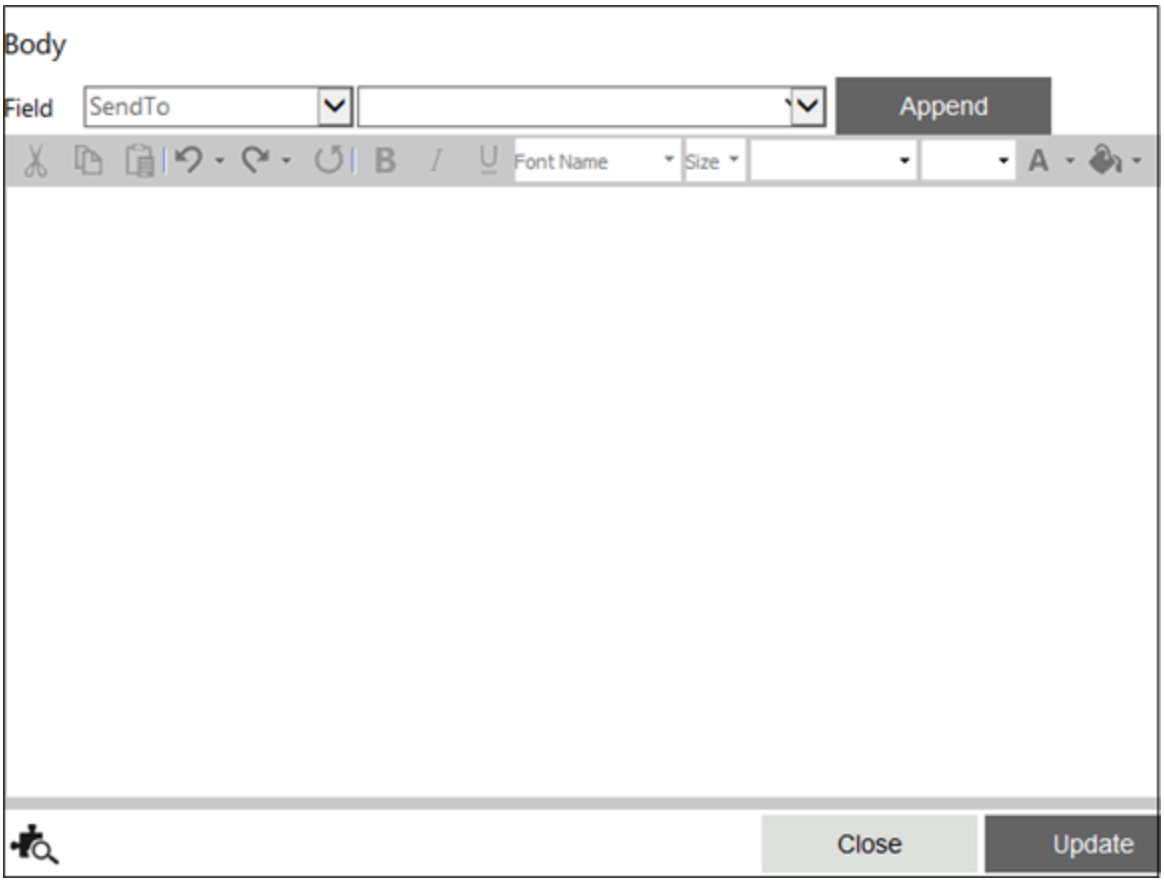
- **Update** - You can click this button to save the subject.
- **Close** - You can click this button to close this window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Approval - Body

Body Property Window

You can specify a message Body for the Approval work item assignment in this property window. To open this window, click the button for the 'Body' property.



The window has a text editor to enter the body text. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the body using a combination of discrete value and variable or content. For example, in the image above, the body message entered is "Hi, Please approve the leave application submitted by <%#Variable.SubmittedBy.name%>". Here, the string "Hi, Please approve the leave application submitted by" has been typed directly in the text editor. The second half of the body is a placeholder, which will be replaced by the variable value during run-time. This has been added by selecting the appropriate variable option from

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the **Field** drop down and clicking **Append**.

The text tool bar above the text editor allows you to manage and format the body text. It has the following text tools:

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools:

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the body message.
- **Close** - You can click this button to close this window.

Approval - Redirect URL

Redirect URL Property Window

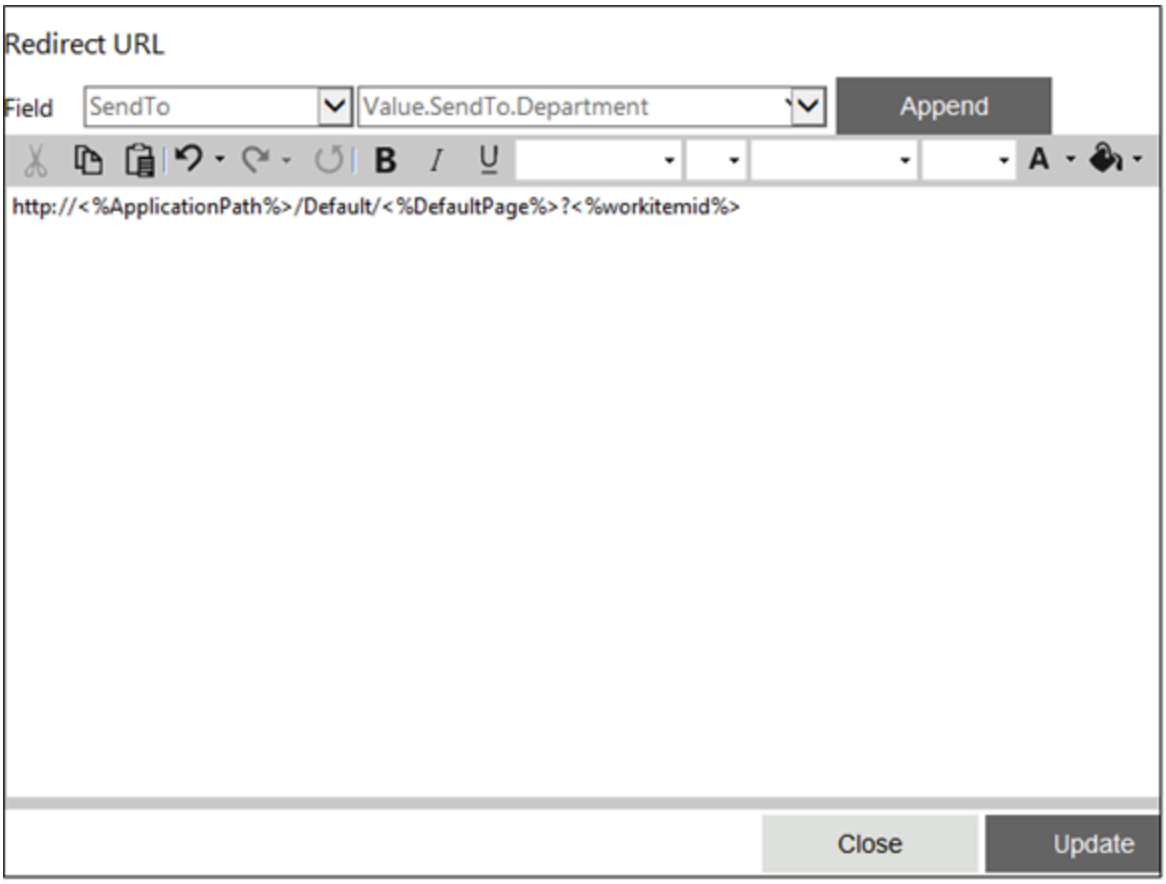
You can specify a message Redirect URL for the Approval activity in this property window. To open this window, click the button for the 'Redirect URL' property.

Set Redirection URL property as

http://<%ApplicationPath%>/Default/<%DefaultPage%>?<%workitemid%>

Note: While pasting the above URL, remove the extra spaces.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the Redirect URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/ drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Redirect URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Redirect URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Redirect URL. It has the following text tools:

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

It also has the following formatting and other tools:

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

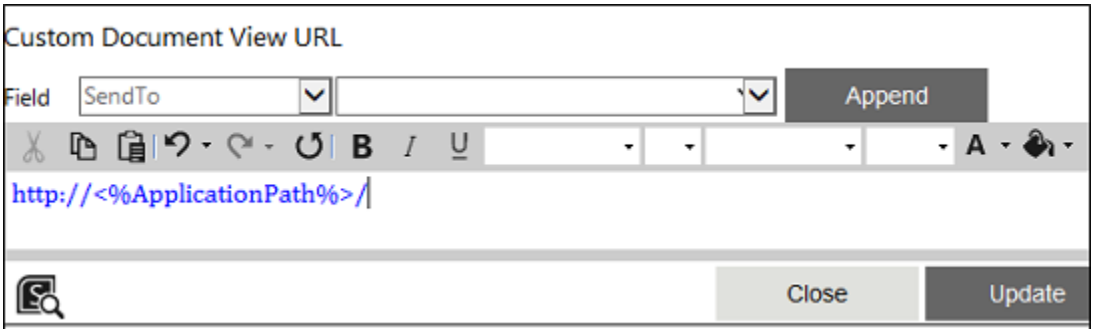
- **Update** - You can click this button to save the Redirect URL.
- **Close** - You can click this button to close this window.

Approval - Custom Document View URL

Custom Document View URL Property Window

You can specify a message Custom Document View URL for the Approval activity in this property window. To open this window, click the button for the 'Custom Document View URL' property.

For more details, see Custom Document View URL



The window has a text editor to enter the Custom Document View URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Custom Document View URL in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

to assign the Custom Document View URL.

- Append - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Custom Document View URL. It has the following text tools:

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools:

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window, you will find the following buttons:

Update - You can click this button to save the Custom Document View URL.

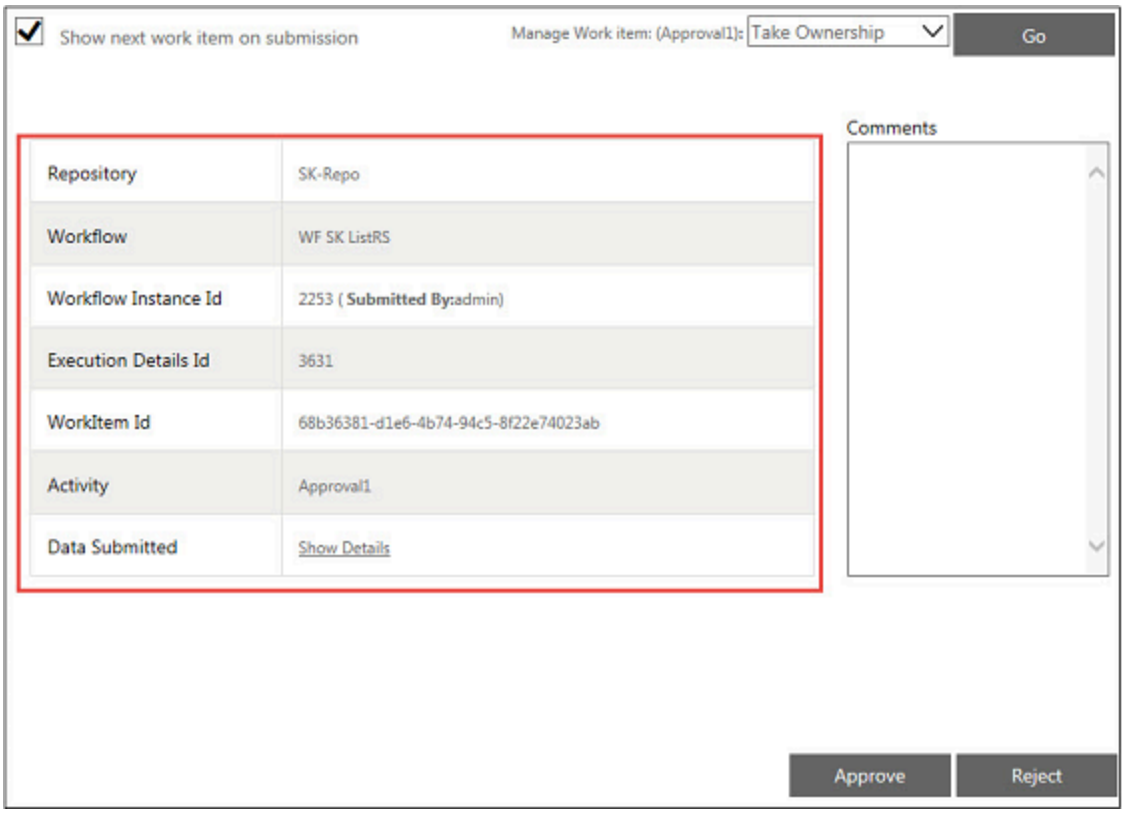
Close - You can click this button to close this window.

Custom Document View URL

Custom Document View

Custom Document View can be viewed on the left side of the Work Item Detail View page inside the frame as shown in the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



In AVEVA Work Tasks, you can provide a custom document view while using any human activity. Enter the proper URL for the Custom Document View URL property to view the custom document view. By default, the custom document view for any human resource activity normally shows the item metadata that alerts the workflow.

1. In AVEVA Work Tasks, if the workflow is alerted due to a Enterprise Console event association, document view shows the list item metadata that alerted the workflow. For example, if the workflow is alerted due to a workflow event association to a list, document view shows the list item that alerted the workflow.
2. If the workflow is alerted due to a workflow association in Forms, document view shows the form that alerted the workflow.

You can change the document view as required. If a URL is provided to the custom document view URL property, this can be overridden with the default document view.

1. If you want to show a different form in the Custom Document View, see Custom Document View URL for Forms.
2. If you want to show a different form or list item in the Custom Document View, see *Custom Document View URL for List Item*.

You can provide any URL to this property. This URL can be a URL to your custom page, too.

Custom Document View URL for Forms

If the workflow is alerted due to a workflow association in forms, document view shows the form that alerted the workflow.

Use the following URL with respective parameters to show any forms.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
http://<%ApplicationPath%>/BPMUITemplates/Default/SkeltaForms/
FormDocumentView.aspx?FormName=<Form Name>&FormVersion=<Form
Version>&FormInstanceXmlVariable=<Xml Variable Name>
```

FormName: Name of the form to be loaded in document view.

FormVersion: Version of the form to be loaded. By default the latest published version will be loaded. This parameter need to be used only if a specific version has to be loaded.

FormInstanceXMLVariable: XML variable name which is to be used for populating the form. If this parameter is not passed, SFFormData XML Variable will be considered as default.

Example:

For showing any form and populate data from XML variable, pass the following parameters in query string provided both having same XML schemas.

FormName=<Form Name>& FormInstanceXmlVariable=<Xml Variable Name>

Sample URL:

```
http://<%ApplicationPath%>/BPMUITemplates/Default/SkeltaForms/
FormDocumentView.aspx?FormName=Leave Report&FormInstanceXmlVariable=LeaveReportxml
```

When the workflow is executed, the **Custom Document View** of the specified form populated using XML variable is displayed in the Work Item detail View page.

Custom Document View URL for List Item or Enterprise Console

If the workflow is alerted due to a workflow association in list, document view shows the form that alerted the workflow.

Use the following URL with respective parameters to show any List Item metadata:

```
http://<%ApplicationPath%>/BPMUITemplates/Default/SkeltaForms/
FormDocumentView.aspx?FormName=<Form Name>& ListName=<List Name>&ListItemId=<List Item
Id>&FormInstanceXmlVariable=<Xml Variable Name>
```

FormName: Name of the form to be loaded in document view.

ListName: Name of the list from where the list item need to be loaded. This parameter needs to be passed only if ListItemId parameter is used.

ListItemId: Id of the list item which need to be loaded to populate the form. If the value of this is passed as 0, then list item id which alerted the workflow is considered as default. This parameter need not be passed, if form is populated using a FormInstanceXMLVariable.

FormInstanceXMLVariable: XML variable name which is to be used for populating the form.

Example:

- To show any form and populate data from XML variable, pass the following parameters in query string: FormName=<Form Name>& FormInstanceXmlVariable=<Xml Variable Name>

  Sample URL:

  ```
  http://<%ApplicationPath%>/BPMUITemplates/Default/SkeltaForms/
  FormDocumentView.aspx?FormName=Leave Report&FormInstanceXmlVariable=LeaveReportxml
  ```
- To show any form and populate data from a list item, use the following format:

  FormName=<Form Name>& ListName=<List Name>&ListItemId=<List Item Id>

  Sample URL:

  ```
  http://<%ApplicationPath%>/BPMUITemplates/Default/SkeltaForms/
  ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
FormDocumentView.aspx?FormName=Leave Report&ListName=Leave
Application&ListItemId=4b3a003-1719-4015-9128-2fe29b9dc535
```

If <List Item Id> is passed as 0, then list item ID which alerted the workflow is taken.

Sample URL:

```
http://<%ApplicationPath%>/BPMUITemplates/Default/SkeltaForms/
FormDocumentView.aspx?FormName=Leave Report&ListName=Leave Approval&ListItemId=0
```

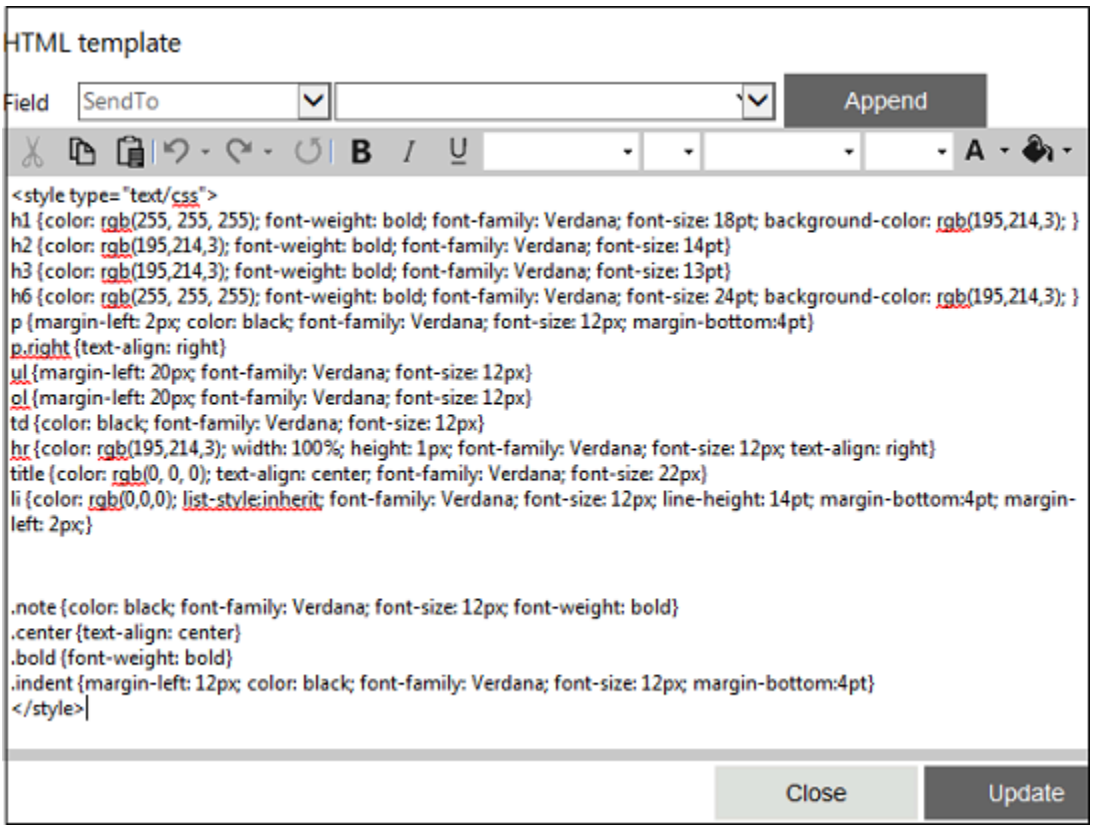
Approval - HTML Template

HTML Template Property Window

You can specify a html template for the Approval activity in this property window. To open this window, click the edit button for the 'HTML template' property.

In this property window, user can write their own html template for approval activity or make use of the default approval activity template for customizing the email.

See Using 'HTML Template' property of Approval Activity for more information.



The window has a text editor to enter the body text. You can do this either by entering a html template directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the message body. The control displayed

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.

- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the html template. It has the following text tools:

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools:

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

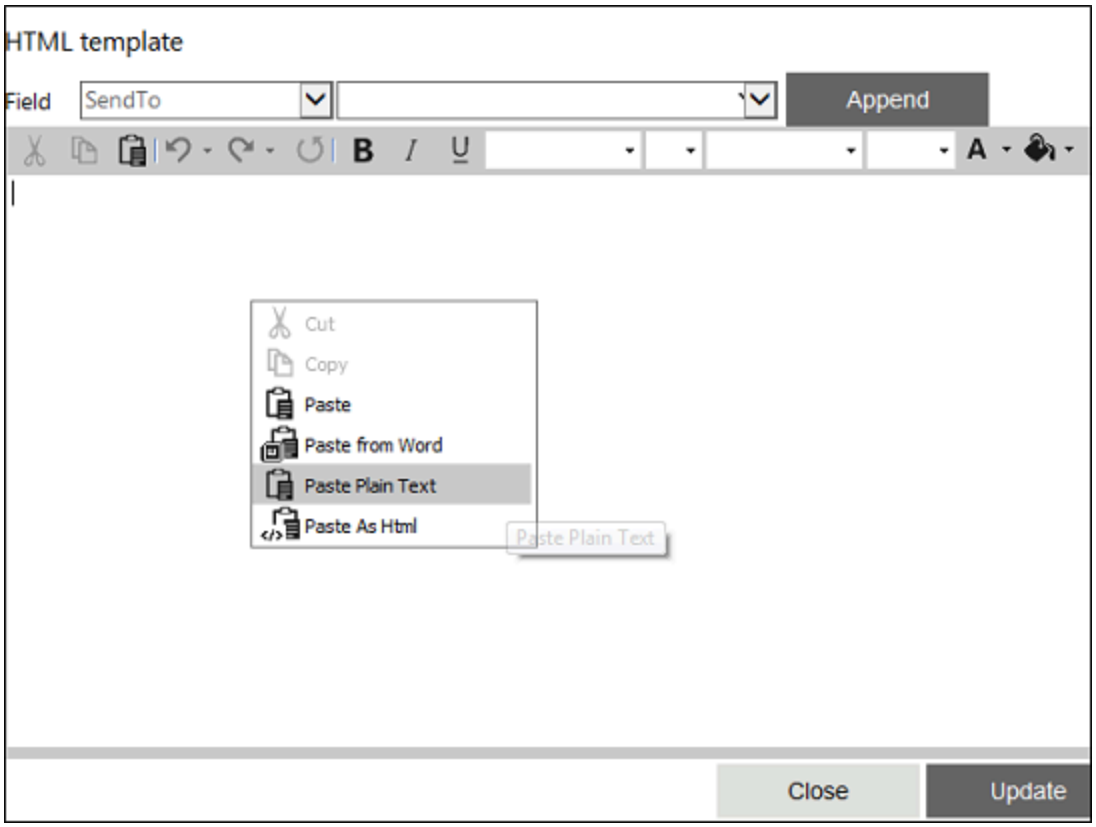
At the bottom of the window you will find the following buttons:

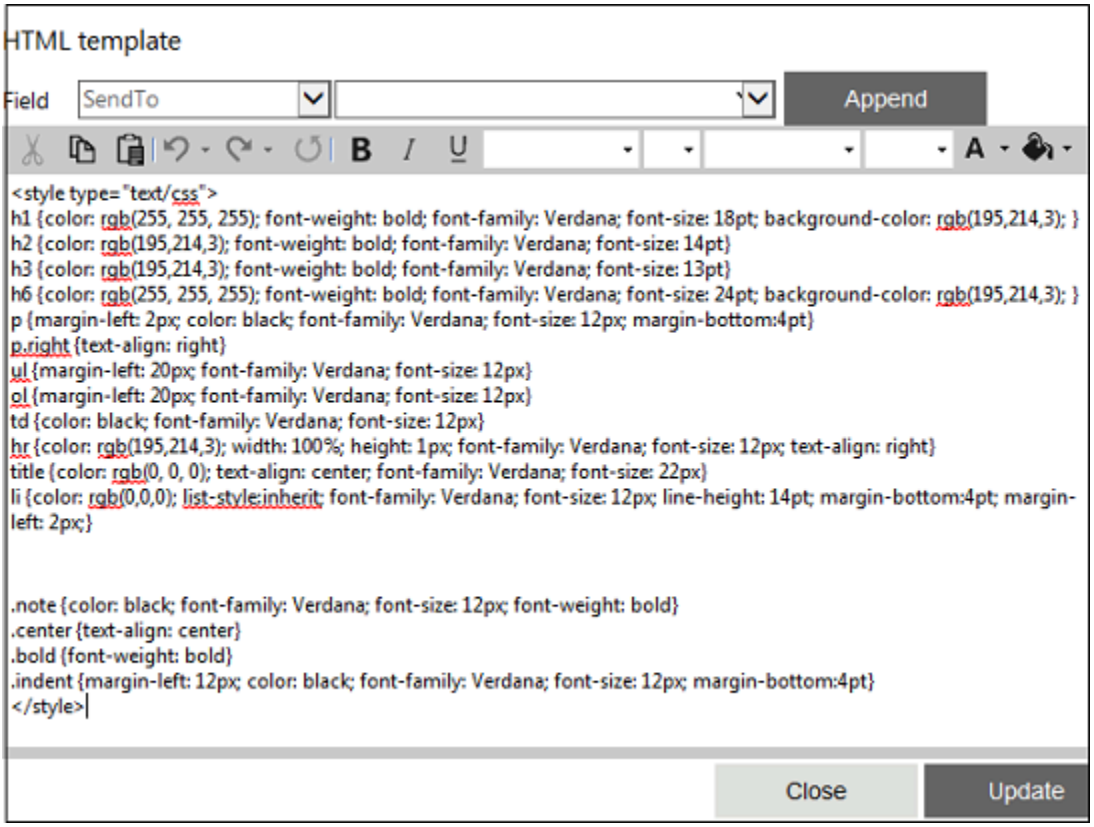
- **Update** - You can click this button to save the HTML template.
- **Close** - You can click this button to close this window.

To set the HTML template, follow the steps given below:

1. To specify the HTML template, click the edit icon to open the property window.
2. In the **HTML template** window, provide the template which should be in html format. If you are pasting a copied html content, right-click the **HTML template** window and paste it as **"Paste Plain Text".**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.





3. Click **Update**.

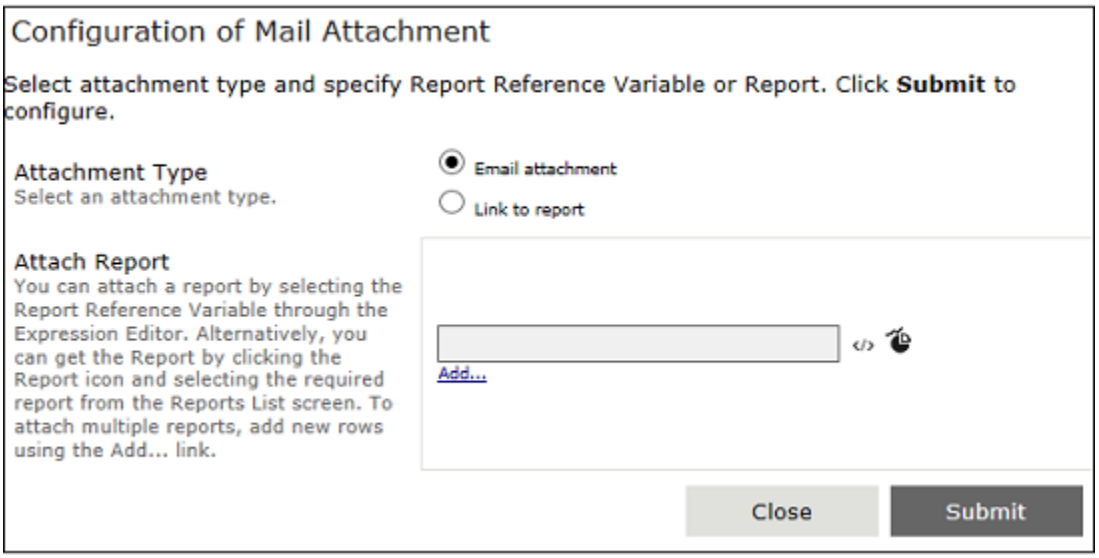
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Approval - Attach Reports

Configuration of Mail Attachment property window

You can configure the **Attachment** property for the **Approval** activity in this property window.

In this property window, you can configure the report attachments to be emailed.



- **Attachment Type:** Use this property to select the type of attachment for sending the report as an email.
  - **Email attachment:** Select this type to send the report as an email attachment.
  - **Link to report:** Select this type to embed a link in the email to download the configured report.
- **Attach Report:** Use this property to attach a report generated by the **Report Generator** activity within the workflow or to select a report from the list of generated reports.

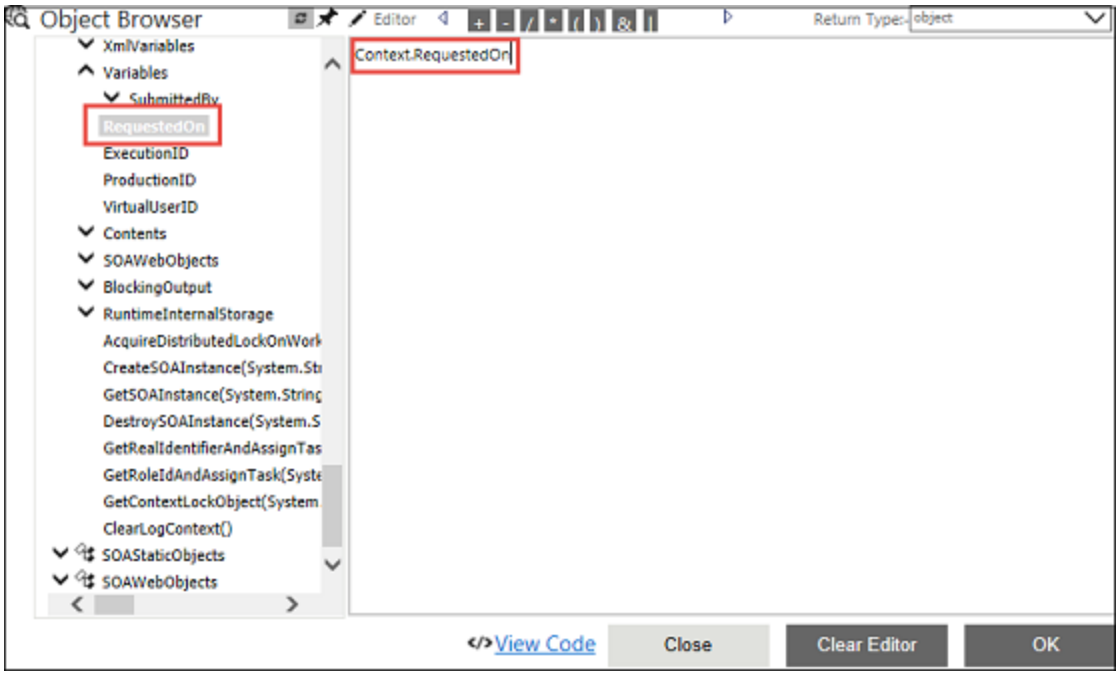
  You can attach a report by selecting the required **Report Reference Variable** using the **Expression Editor**. Alternatively, you can select the report from the Reports List window which consists of the already generated reports.

There must be at least one **Report Generator** activity before an **Approval** activity in the workflow to select the **Report Reference Variable** (using Expression Editor) as an **Attachment**. The **Report Reference Variable** is generated while configuring the **Report Generator** activity. By default, the report reference variable is **[Template Name]_[activity Name]**. For more information about **Report Reference Variable**, see Configure Report Template.

To configure mail attachment

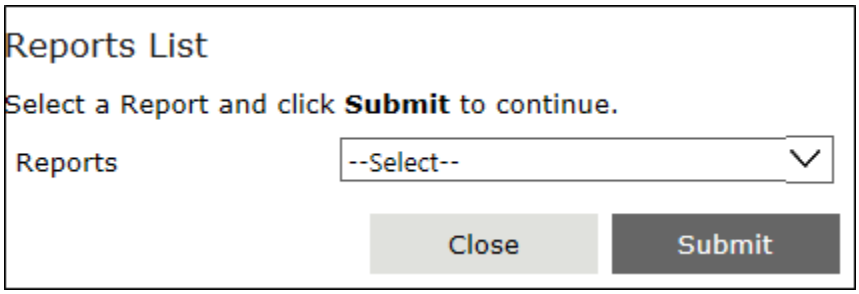
1. Click the edit button for the **Attachment** property.
2. Select the required **Attachment Type**.
3. In the **Attach Report** text box, you can either use the expression editor to get the variable of the report or use the **Reports List** to select the report.
4. To get the variable, click the **Expression Editor** icon given beside the text box to select the required **Report Reference Variable**.
5. In the **Expression Editor** window, select the required variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



6. Click **OK.**
7. Click **Close**.

   The variable appears in the **Attach Report** text box.
8. To select the report from the Reports List dialog box, click the **Reports** icon.



9. In the Reports List dialog box, select the required report, and then click **Submit**.

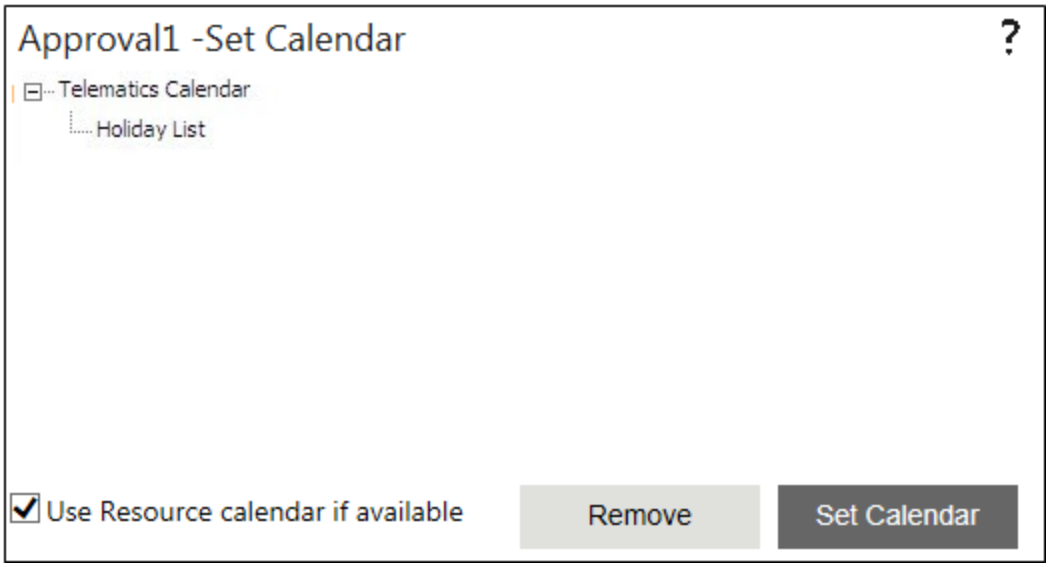
   This window closes automatically.
10. Click **Close** to close the configuration window.

Approval - Set Calendar for Timeout

Set Calendar for Timeout Property Window

You can set a Calendar to calculate the timeout for the Approval activity in this property window. To access this window, click the button for the 'Set Calendar for Timeout' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



This window lists the available calendars in a tree structure, with lower-level calendars nested within the respective top-level calendar. You can click a calendar to select it for setting.

There are two buttons in this window:

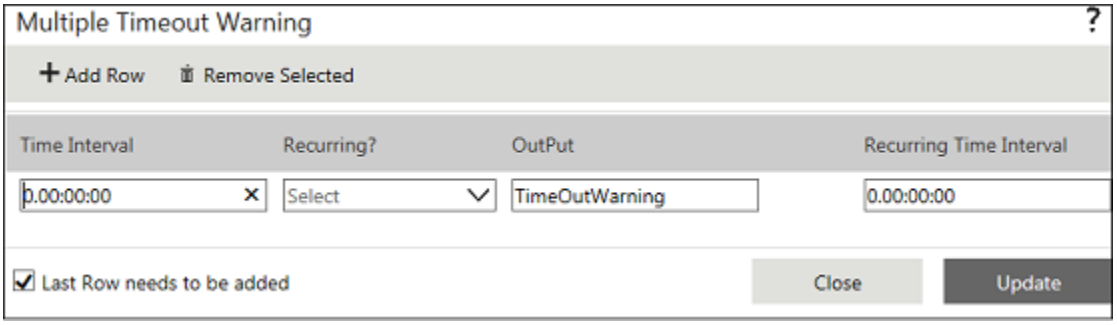
- **No Calendar** - You can click this button to indicate that there is no calendar set for the activity.
- **Set Calendar** - You can click this button to set the selected calendar.

The **User Resource calendar if available** check box can be selected to enable the use of resource calendars if none are displayed in this window for selection.

Approval - Multiple Timeout Warning

Multiple TimeOut Warning Property Window

You can set Multiple Timeout Warning in this property window. To open this window, click the button for the 'Multiple TimeOut Warning' property.



The window has two tabs at the top to manage the rows of timeout settings in the grid.

- **Add Row** - You can click this tab to add the newly entered timeout setting and to display a new row in the grid where you can add another timeout.
- **Remove Selected** - You can click this tab to remove the selected timeout setting row in the grid.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The grid has the following parameters that you need to specify for the new timeout setting.

- **Time Interval** - You can use this field to enter the time interval in which you to generate a timeout warning. You must enter the time in d.hh:mm:ss format.
- **Recurring?** - You can use the drop down in this field to indicate whether the timeout is recurring or not. You can select "Yes" to set the timeout warning to recur.
- **Output** - You can enter the output message corresponding to the timeout warning here.
- **Recurring Time Interval** - In this field, you can set the time interval in which the timeout warning has to recur. Enter the time in the d.hh:mm:ss format.

There are two buttons and a check box at the bottom of the screen.

- **Update** - You can click this button to save the newly added timeout settings.
- **Close** - You can click this to close the property window without saving the newly added timeout settings.
- **Last Row needs to be added** - You need to select this check box to save the last added row.

Checklist Activity

Activity Description

The Checklist Activity is incorporated in the AVEVA Work Tasks Workflow process design to send questions that need to be answered by actors in a workflow. These questions can be sent with a Text box, Memo, Checklist, Drop-down, or Radio Button to capture the answers provided by actors.

Application Scenarios

A typical application scenario could be a Technical Support Request Workflow Process, which will have a set of radio buttons on the Request Form which the customer can use to select the Support option applicable to him/her. A Checklist activity can be used in the workflow to display these radio buttons and retrieve the customer selection information.

Workflow Variables for Checklist

Steps to define variables for use in the Checklist activity

This section describes how to define variables for use in Checklist activities. It gives the syntax for creating the variables and typical examples for different cases.

Variables for capturing WorkItemIds

The Work Item Id's for the Checklist work item can be retrieved using a variable of type Array, with the following syntax:

'^'+CurrentActivity.Name+'WorkItemIds'.

For example, **^ReviewDocumentWorkItemIds**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Variables for capturing All Actors' Details

The details of the actor(s) to whom the Checklist work item gets assigned, can be accessed by declaring a variable of type array with the following syntax:

'^'+CurrentActivity.Name+'ActingUsers'

For example, **^ReviewDocumentActingUsers**.

Variables for capturing Responding Actors' Details

The details of the actors' submitting the Checklist response, can be accessed by declaring a variable of type resource with the following syntax:

'**^'+HWSActivity.ActivityName+'RespondingUser'**

For example, **^ReviewDocumentRespondingUser**.

This variable stores the responded user information. If it is more than one user, it will retrieve the last user details.

Note: For single user, you can declare this variable as Resource type. If task is assigned to multiple users and to get all responded users information, then the variable has to be declared as Array.

Variables for capturing Comments

Comments entered in a Checklist can be retrieved using a variable with the following syntax:

'^'+_HWSActivity.CurrentActivity.Name+'CommentCheckList'.

For example, **^ReviewDocumentCommentCheckList**.

(To add a comment field to the Checklist HTML page create a Text box control in Checklist HTML pages with the name "txtcomment". The HTML template pages for the Checklist activity can be found under the ..[AVEVA Work Tasks Install Path] \WorkflowElements\Default\en-US\Activities\Activity folder.)

Variables for capturing Checklist Questions and Answers in HTML format

Question and answers entered in the Checklist can be accessed in HTML format by using a variable with the following syntax:

'**^'+CurrentActivity.Name+'AnswersHTMLCheckList'**

For example, ^ReviewDocumentAnswersHTMLCheckList.

Variables for capturing Checklist Questions and Answers in XML format

Question and answers entered in the Checklist can be accessed in XML format by using a variable with the following syntax:

'**^'+ CurrentActivity.Name+'AnswersXMLCheckList'**

For example, **^ReviewDocumentAnswersXMLCheckList**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Variables for assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the Start activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the Workflow.

To assign multiple queues, specify the queue-ids with semicolon (;) as the separator. Skelta.HWS.Queue.QueueCollection class can be used to retrieve queue information.

Variable for Queue Acting Users

You can declare an Array type variable to get updated information about the name of the actor to whom an activity gets assigned from a queue.

In the Workflow Variables, define the variable in the following format:

^+CurrentActivityName+'QueueActingUsers'

Ex: ^HRFormQueueActingUsers

HRForm is the activity name.

For example the value of the user who acts on the HRForm activity will be stored in the variable. The Approval1 task will be assigned to the same user.

Variable for storing timeout warning value

Suppose if you want to store the timeout warning value for Checklist activity in your workflow. The naming convention for the system variable to be declared is

^<ActivityName>ResponseByWarningTimeout

The value returned by the variable will be in UTC format. For multiple user scenarios the highest timeout value calculated for user will be available in the variable.

Note: Multiple timeout warnings cannot be tracked using this feature.

Retrieving Stored Checklist Data:

When an actor has acted upon the checklist, it is marked as completed and the details are stored in XML format in the database. It can be retrieved as SELECT DATA FROM SKWorkItem SKW, SKWorkItemData SKWD WHERE

SKW.DataId = SKWD.Id AND SKW.Id = <WorkItemId>.

See the XML file to know how the details are stored in the database.

The following image shows the format in which the Checklist Details are stored in the XML file:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
- <checklist>
  - <question identifier="q1" decisionmaker="" questiontype="radio" questiontypecount="radio1" optioncount="1" includeinxml=""
      mandatory="yes" defaultvalue="" submittedby="John Smith" submittedbyidentifier="" name="RADIO">
      <option value="FirstQuestion">FirstQuestion</option>
      <answer>FirstAnswer</answer>
    </question>
  - <question identifier="q2" decisionmaker="" questiontype="radio" questiontypecount="radio2" optioncount="2" includeinxml=""
      mandatory="yes" defaultvalue="" submittedby="John Smith" submittedbyidentifier="2" name="RADIO1">
      <option value="SecondQuestion">SecondQuestion</option>
      <option value="SecondQuestion1">SecondQuestion1</option>
      <answer>SecondAnswer</answer>
    </question>
  - <question identifier="q3" decisionmaker="yes" questiontype="dropdown" questiontypecount="dropdown1" optioncount="3"
      includeinxml="" mandatory="yes" defaultvalue="" submittedby="John Smith" submittedbyidentifier="3" name="DROPDOWN">
      <option value="ThirdQuestion">ThirdQuestion</option>
      <option value="ThirdQuestion2">ThirdQuestion2</option>
      <answer>ThirdAnswer</answer>
    </question>
  - <question identifier="q4" decisionmaker="" questiontype="dropdown" questiontypecount="dropdown2" optioncount="4" includeinxml=""
      mandatory="yes" defaultvalue="" submittedby="John Smith" submittedbyidentifier="4" name="DROPDOWN1">
      <option value="FourthQuestion">FourthQuestion</option>
      <answer>FourthAnswer</answer>
    </question>
  - <question identifier="q5" decisionmaker="" questiontype="textbox" questiontypecount="textbox1" optioncount="5" includeinxml=""
      mandatory="yes" defaultvalue="" submittedby="John Smith" submittedbyidentifier="5" name="TEXTBOX">
      <answer>FifthAnswer</answer>
    </question>
  - <question identifier="q6" decisionmaker="" questiontype="memo" questiontypecount="memo1" optioncount="6" includeinxml=""
      mandatory="yes" defaultvalue="" submittedby="John Smith" submittedbyidentifier="6" name="MEMO">
      <answer>SixthAnswer</answer>
    </question>
  - <question identifier="q7" decisionmaker="" questiontype="checkbox" questiontypecount="checkbox1" optioncount="7" includeinxml=""
      mandatory="yes" defaultvalue="" submittedby="John Smith" submittedbyidentifier="7" name="CHECKBOX">
      <option value="SeventhQuestion">SeventhQuestion</option>
      <answer>SeventhAnswer</answer>
    </question>
</checklist>
```

Activity Properties:

The Checklist activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Checklist activity. The Activity Properties are organized under the following groups of related properties.

Name & Description

You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Display Name - This property specifies the name to be displayed for the work item in the Work items list.

Property Type: Optional (This property need not be set if the display name is already configured in the Actions.XML. The name entered here overrides any display name that is configured for this activity in the Actions.xml file.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the functionality of the activity.

Delivery Channels - This property is used to specify the delivery channels to deliver the activity to the assigned actor(s). Multiple channels can be specified giving the actor(s) different options to act on the work item.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Delivery Channels property

1. To select the Delivery Channels, click on the button for the 'Delivery Channels' property the property window will be displayed.
2. In this window, the Delivery Channels that have been configured and will be listed on the left along with a check box against each channel. Click and select the check box corresponding to the channel you wish to select. You can select multiple channels by selecting the respective check-box.
3. Click on Save Channels button.

See CheckList - Delivery Channels for a detailed description of the Delivery Channels property window.

Questions - This property is used to define the Question and Answer Types in a Checklist Activity; the Answer Types provided are Textbox, Memo, Checklist, Drop-down, and Radio button.

Property Type: Mandatory (This property must be set for the activity to be used in the workflow.)

Attributes of the Questions property

- The user interface displays the Question and Answers
- Decision Maker functionality
  - There can be a Decision Maker associated with a Checklist. The outcome of the Checklist Activity is determined by the Answers tagged with the Decision Maker. A Question marked with a red exclamation mark ! is said to be the 'decision maker' for the Checklist activity.
  - Only Drop-down can have the Decision Making functionality tagged. There can be only one 'decision maker' for a Checklist activity. For example, if the decision maker is tagged with a Question which has 'Approved' & 'Rejected' as values, and the Checklist is submitted with 'Approved' selected in the drop down, then the engine is alerted about the 'Approved' status and the engine executes the next process in the workflow.
  - Whenever an actor acts upon (submits) the checklist the engine is notified with the answer to the question, provided the question is marked as a decision maker.
- Mandatory Functionality
  - If an answer type is marked as mandatory the actor will be prompted to enter the value before submission. A Question marked with a red asterisk * is set as 'mandatory' when the actor fills up the Checklist. The actor cannot submit the checklist answers without filling the 'mandatory' questions.
- Default Functionality
  - Inputs given as default will be shown as the default answer for the answer type in the user interface.
- Edit and Delete options have been provided
- To add Questions and Answers:
  - First the actor clicks on the Add New button (the option to add more than one question has also been provided.)
  - Then the actor is asked to enter the question and the answer type,
  - Based on the answer type the actor is asked to enter the values (for example, if the answer type is drop-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

down, then you have to enter "display text" and the value for the display text.)

Steps to set the Questions property

1. To specify the questions and answers, click the button for the 'Questions' property to open the property window.
2. In this window, specify the number of questions to add in the **No. of additional questions** text box and click **Add New**. This will open a new window.
3. In the second window, enter the question in the text box under the **Questions** column.
4. Next specify the answer type for the question by selecting an option from the drop down under the Answer Type column. Depending on your selection, the question will take answers in a text box or memo, or as options in a drop-down, check box, or radio.
5. Specify the number of options for the selected answer in the **No. of Options** text box.
6. If you selected "drop-down" as the answer type in step 4, a radio is displayed under the ! column. Click this radio, if you want to make this question the decision maker. This implies that the answer to this question will determine the outcome of this activity.
7. Select the check box under the **Mandatory*** column to make this question mandatory.
8. Repeat steps 3 through 7, if you are adding more than one question in the checklist.
9. After setting the parameters for all the questions click on the **Next** button to go to the next window.
10. In the third window, specify the options that will be presented to the actor to answer the question (if the question type is drop-down, check box or radio) in the text boxes under the **Options** column.
11. Next, specify the option that will be the default value by clicking the corresponding radio button under **Default Value** column.
12. You must also specify the values that the answer will take internally in the **Values** text box.
13. In addition you can select a variable from the drop down for the **Map question answer to variable** field. The answers to this question will be stored in the selected variable and will be available for use in the workflow.
14. After completing all the details you can click on the **Finish** button to save the question.

    The first window is displayed again when you click on the Finish button. It shows the newly added question(s). You can modify or delete the new question or add more new questions. If you do not wish to perform any more operations, click Close to exit this window.

See CheckList - Questions for a detailed description of the Questions property windows.

Work Item Fields - This property is used to add the Work Item Fields to be used to display application data as well as value in variables and content in the Work items list when this activity is executed.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Work Item Fields property

1. To select the Work Item Fields, click the button for the 'Work Item Fields' property to open the property window.
2. In this window, the Work Item Fields that have been set are listed in a grid. Click on a row to select the corresponding Work Item Field.
3. Right click the row, and select the "SetWorkItemField" option from the context menu to set it.

See CheckList - Work Item Fields for a detailed description of the Work Item Fields property windows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Actors

The properties in the Actors group have been described below. You can use these properties to specify the actors who will work on the activity and related properties.

Assign Actor(s) - This property is used to specify the actor(s) who has to take up the Checklist activity. It is tagged with the Ignore Unavailable Actors property which in turn is tagged with the Alternate Actor Allowed property. This will be explained in detail later.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any actor(s).)

Steps to set the Assign Actor(s) property

1. To specify the actors, click the button for the 'Assign Actor(s)' property to open the property window.
2. In this window build an expression to select the actors using the actors properties. Select an attribute from the first drop-down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop-downs. Select "End" from the last drop-down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See CheckList - Assign Actor(s) for a detailed description of the Assign Actor(s) property window.

See the Using XML Variables section in the About XML Variables topic for details on how to use the XML Variables to build the expression.

Assign Queue(s) - This property specifies the queue(s) to which the Checklist activity is to be assigned.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any Queue(s).)

Steps to set the Assign Queue(s) property

1. To specify the queue(s), click the button for the 'Assign Queue(s)' property to open the property window.
2. In this window select the queue(s) from the **Available Queues** list box. You can select multiple queues by selecting with the <Ctrl> key pressed.
3. After selecting the required queue(s) click the [>] button to move it to the **Selected Queues** list box. You can click the [>>] button to select all the queue(s) and move them to the **Selected Queues** list box.
4. You can remove any queue from the selected list by selecting it and clicking the [<] button. You can remove all the queues by clicking on the [<<] button.
5. After adding the required queue(s), click **Assign** to assign the activity to the selection.
6. Click **Close** to close the window without saving the selection.

See CheckList - Assign Queue(s) for a detailed description of the Assign Queue(s) property window.

Min. Ownerships [% or Number] - This property, can be specified as a number or percentage, is the minimum number of actors who should take ownership of the Checklist request. This property is related to the 'Timeout warning for Ownerships' and 'Timeout for Ownership' properties.

To illustrate, when 'Timeout for Ownerships' is reached, the activity first checks if the ownership count is less

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

than 'Min.Ownerships' (if it is set).

- If it is, the work item will be removed from all Work Item lists.
- If the 'Min.Ownerships' has already been satisfied, then the work item will be removed from the Work Item lists of those who have not taken ownership.
- If the 'Min.Ownerships' value is 0, then the activity checks if the ownership count is less than 'Max.Ownerships' (if it is set). If it is, the work item will be removed from all Work Item lists. Else the work item will be removed from the Work Item lists of those who have not taken ownership.
- If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work Item lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Max. Ownerships [% or Number] - This property, which can be specified as a number or percentage, is the maximum number of actors who should take ownership of the Checklist request.

- If 'Timeout for Ownerships' is specified and achieved, and the 'Min.Ownerships' is also satisfied, then the activity waits for action from those who took ownership.
- If the Min.Ownerships value is set to 0, then it checks if the ownership count is less than Max.Ownerships.
- If it is, the work item will be removed from all Work Item lists. Else the work item will be removed from the list of those who have not taken ownership.
- If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work Item lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Enable Take Ownership? - If this property is set to **Yes**, then the actor is provided with a **Take Ownership** option in the Work items list to take ownership of the task. If you are not required **Take Ownership** option in the Work items list, then set this property as **No**.

Note: By default, this property is set as Yes.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Enable Take Ownership? property

- Select **Yes** or **No** option from the Enable Take Ownership? drop-down list.

Re-Assign Allowed? - If this property is set, then the actor is provided with a Re-Assign option. Using this the actor can re-assign the work item to another actor.

Property Type: Optional (This property needs to be set only if necessary.)

Ignore Unavailable Actors? - This property provides the following options - **Yes**, **Yes** - **holidays excluded** and **No**.

- If this property is set to 'No', when the Approval activity is assigned to the actor's Work Item list, a check is first done to see whether the actor is available. If the actor is not available then a check is made for an alternate actor assigned to the unavailable actor. Simultaneously the value of the property 'Alternate Actor Allowed' is checked. If that is also set, then the Approval work item is assigned to this actor.
- If this property is set to **Yes - holidays excluded**, the Workflow engine will ignore the fact that actors are not available if the day is an organization holiday and the output Not Enough Resources will not be generated.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if necessary.)

Alternate Actor Allowed? - This property is set to assign a particular task to an alternate actor in the absence or unavailability of any actor.

Property Type: Optional (This property needs to be set only if necessary.)

Notification Contents

The properties in the Notification Contents group have been described below. You can use these properties to specify the content used to notify actors about the activity.

Subject - This is the subject of the Checklist work item. The subject can be customized for each actor.

Property Type: Optional (This property need not be set. If it is not set, the default subject line from the Actions.XML file is displayed.)

Steps to set the Subject property

1. To specify the subject, click the button in the 'Subject' property to open the property window.
2. In this window, enter the subject text directly in the text editor.
3. If you want to customize the subject you can do so using the Field controls to add resource property values. For example, to add a subject that includes the email id of the actor submitting the checklist, you can do as follows: First type in the text "Submitted By". Next select "Variable" and "Variable.SubmittedBy.Email" from the two drop downs for Field and click Append. This will add the placeholder for the variable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the email id.
4. You can edit the text, using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes and also zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the subject, click **Update** to save it.
7. Click **Close** to close the window without saving the subject.

See CheckList - Subject for a detailed description of the Subject property window.

Body - This is the actual message of the Checklist sent to the actor(s). The body can be customized for each actor. There is an Enterprise Console Explorer(plug-in) edit icon in the Body property window to get the Work item details URL. See Enterprise Console Control (EC Plug-in) for more information.

Property Type: Optional (This property need not be set.)

Steps to set the Body property

1. To specify the message body, click the button in the 'Body' property to open the property window.
2. In this window, enter the body text directly in the text editor.
3. If you want to customize the body you can do so using the Field controls to add resource property values. For example, to add a body that includes the name of the actor submitting the checklist, you can do as follows: Type in the text "Hi, Please cast your vote in the poll conducted by". Select "Variable" and "Variable.SubmittedBy.name" from the two drop downs for Field and click Append. This will add the placeholder for the variable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the name.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. You can edit the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command and also zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the body, click **Update** to save it.
7. Click **Close** to close the window without saving the body.

See CheckList - Body for a detailed description of the Body property window.

From Email Address? - This property is used to specify the email address from which notification messages will be sent for the activity.

Property Type: Optional (This property needs to be set only if necessary. If it is not set, the default value in Actions.XML will be used.)

Hide Responses? - If this property is set to 'No', then actors can see the responses of other actors in a group response scenario. To view these responses, such as ownerships and action, select the 'Take Ownership' option. This will open a new pop up with the responses of other actors who have taken ownership.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Priority - This property is set to indicate the priority of the checklist request. Enter the values 0-33 to set as Low, 34-66 for Medium, or 67-99 for High. The default value is set to 34.

Property Type: Optional (This property needs to be set only if necessary.)

Show Custom Document View Window - If this property is set to 'Yes', then the Custom Document View Window will be shown to the users in their Work Item Lists.

Property Type: Optional (This property needs to be set only if necessary.)

Send Notification Email - If this property is set to 'Yes', then notification emails will be sent to the actors. If it is set to 'Yes-HTML', then the emails will be sent in HTML format.

Property Type: Optional (This property needs to be set only if necessary.)

Notify if alternate actor is considered- If this property is set to 'Yes', then notification emails will be sent to the actors when an alternate actor is considered.

Property Type: Optional (This property needs to be set only if necessary.)

Redirect URL - This property is used to get the URL of the web page to redirect the client to, in the case of synchronous execution of the action.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Redirect URL property

1. To specify the Redirect URL, click the button in the 'Redirect URL' property to open the property window.
2. In this window, enter the Redirect URL directly in the text editor.
3. If you want to customize the Redirect URL you can do so using the Field controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for Field, and clicking Append. This will add the placeholder for the variable in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the URL.
4. You can edit the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

and redo changes, repeat the last command and also zoom in and zoom out the view as well as print the text.

5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Redirect URL, click **Update** to save it.
7. Click **Close** to close the window without saving the Redirect URL.

See CheckList - Redirect URL for a detailed description of the Redirect URL property window.

Custom Document View URL - This property is used to get the URL of the page to which the custom document dlls does a post. The data posted to the page are application, workflow, executionid, executiondetailsid, and workitemid. The posted data can be retrieved as Request.Form["workitemid"].

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Custom Document View URL property

To specify the Custom Document View URL, click the button in the 'Custom Document View URL' property to open the property window.

1. In this window, enter the Custom Document View URL directly in the text editor.
2. If you want to customize the Custom Document View URL you can do so using the Field controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for Field, and clicking Append. This will add the placeholder for the variable in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the URL.
3. You can edit the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command and also zoom in and zoom out the view as well as print the text.
4. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
5. After adding the Custom Document View URL, click **Update** to save it.
6. Click **Close** to close the window without saving the Custom Document View URL.

See CheckList - Custom Document View URL for a detailed description of the Custom Document View URL property window.

Mail Channel

The properties in the Mail Channel group have been described below. You can use these properties to customize the approval activity (activities) with different mail channel templates.

Use html template from property - This property is set to 'No', then it will take the mail channel template from the physical path

'[AVEVA Work Tasks Installed Path]\BPM.NET\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\CheckListEmail.html.

If this property is set to **'Yes'** and the "**HTML Template**" property is empty, then it will take the mail channel template from the physical path

'[AVEVA Work Tasks Installed Path]\BPM.NET\WorkflowElements\Default\en-US\Templates\

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

OutlookEmailTemplate\CheckListEmail.html.

Property Type: Optional (This property needs to be set only if necessary.)

HTML template - This property is used to define the html template for mail channel.

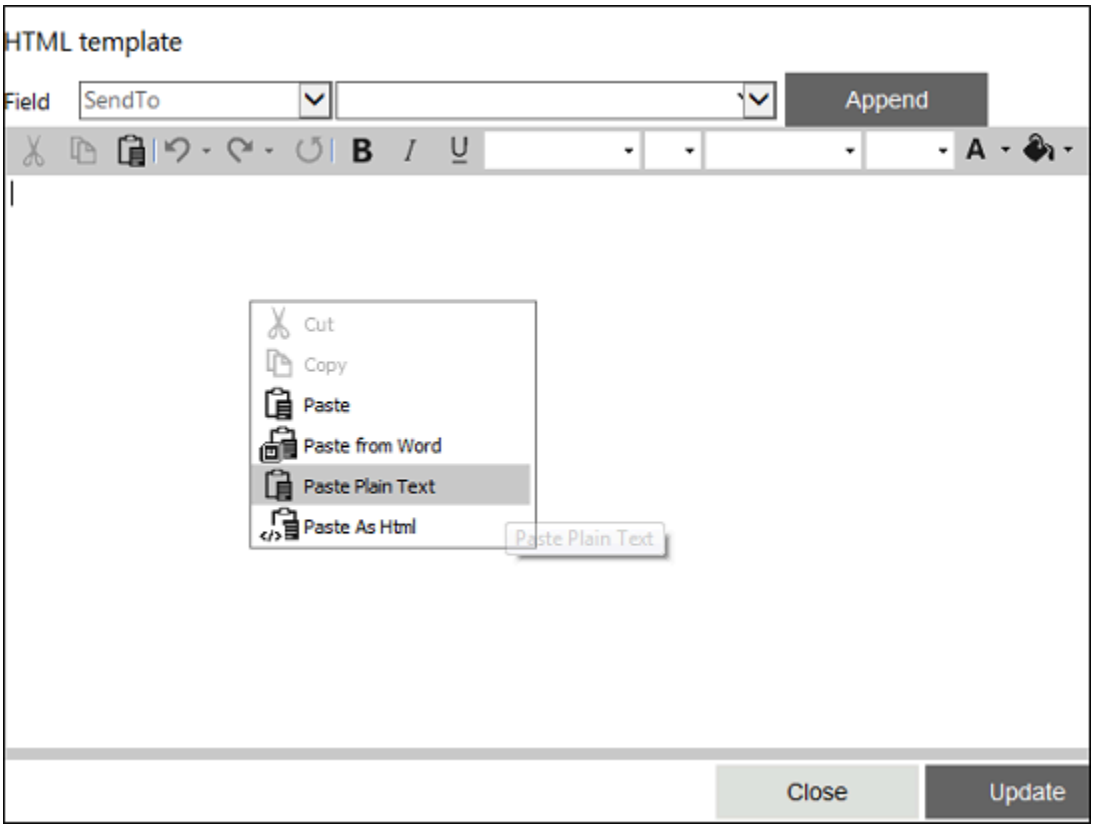
Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the HTML template

1. To specify the HTML template, click the edit icon to open the property window.
2. Click **Update**.

See CheckList - HTML Template for a detailed description of the property HTML template window.

Attachment



Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Attachment property

1. To specify the Attachment property, click the edit icon to open the property window.
2. In the Configuration of Mail Attachment window, select the required attachment type by clicking the radio button.
3. In the Attach Report property, you can attach a report by selecting the Report Reference Variable through the Expression Editor. Alternatively, you can get the Report by clicking the Report icon and selecting the required report from the Reports list screen. To attach multiple reports, add new rows using the Add... link.
4. Click **Submit**.

See CheckList - Attach Reports for a detailed description of this property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Decision Influencers

The properties in the Decision Influencers group have been described below. You can use these properties to specify some parameters that influence the outcome of the activity.

Max Actors Required For Checklist Completion [% or Number] - This property is used to check what percentage or number of actors should take action on the Checklist. Decision on the Checklist is taken only after the specified number or percentage of actor(s) have taken the action. The activity waits for the actors' action until the 'Timeout for activity' is reached.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Min Actors Required For Checklist Completion [% or Number] - This property specifies the minimum number of actors who need to act on the Checklist before the 'Timeout Warning for Activity'. If the action limit is reached before the 'Timeout Warning for Action' then the warning message is not shown.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Escalation & Timeout

The properties in the Escalation & Timeout group have been described below. You can use these properties to specify escalation and timeout settings for the activity.

Set Calendar for Timeout - This property is used to specify the calendar to be used to determine activity timeouts.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to Set Calendar for Timeout

1. To specify the calendar for timeout, click the button in the 'Set Calendar for Timeout' property to open the property window.
2. In this window, select the required calendar by clicking on the calendar name in the tree structure.
3. Click **Set Calendar** to set the selected calendar.
4. You can specify that a resource calendar should be used, if available, by clicking to select the corresponding check box.

See CheckList - Set Calendar for Timeout for a detailed description of the Set Calendar for Timeout property window.

Minimum available time for Activity - This property is used to check the availability of an actor to whom a task must be assigned while executing the activity. The availability check for the actor is done based on the Calendar settings (Global or Resource Calendar). If the time taken to complete the task is within the value set for the **Minimum available time for Activity** property, then the task will be assigned to that actor. However, if the time taken to complete the task exceeds the value set for the **Minimum available time for Activity** property, then the task will be assigned based on the values set for the **Ignore Unavailable Actors?** and **Alternate Actor Allowed?** properties.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout warning for Ownership - This property specifies the time at which a warning is provided to an actor to take ownership of the work item assigned to him/her.

- If the actor does not take ownership within this time the engine is notified with a warning. Thus if 'Min.Ownerships' is set, and it is not satisfied before the given time then a warning is sent.
- If Min.Ownerships is 0, it will check for Max.Ownerships. If the ownership count does not reach the Max.Ownerships value within the set time, a warning is sent.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- If both Min.Ownerships and Max.Ownerships are not set, this property is ignored.

Property Type: Optional (This property needs to be set only if necessary.)

Understand how Ownership Properties relate to each other:

- Checklist activity is assigned to the actors in the Checklist assignment list by the Workflow Engine. Checklist activity has a default property set that at least one actor has to make a Take Ownership of the Checklist activity.
- A drop-down interface has been provided for the actor to Take Ownership. However, the actor can act on the Checklist without taking ownership first, in which case ownership will be considered to be done. i.e. if the actor acts upon the Checklist activity without taking ownership, then it is considered as an ownership with action taken.
- After the actor has taken ownership, a link is provided to the actor, which opens a pop up with the details of the other actors. This is again tagged with the 'Hide Responses?' property. The link will be provided only If 'Hide Responses?' is set as 'No'. The pop up will have details like Actor name, Ownership Status, Action taken, Comments.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] then the 'Timeout Warning For Ownership' property is not considered.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] and the 'Time Out For Ownership' is also reached then the work item is removed from other actors' Work Item list and the work item waits for action from those who have taken ownership.
- If the 'Min.Ownerships' limit is not reached for the Checklist activity, and the 'Timeout Warning For Ownership' has been exceeded then the actor is notified with a warning by including an email or an information activity, with the Timeout Warning for Ownership link to it.
- If 'Timeout For Ownership' is set, if the Checklist activity has not reached the minimum limit (Min.Ownerships) and the 'Time Out For Ownership' has been reached, then the work item is removed from the Work Item lists of all the actors.
- If 'Min.Ownerships' = 0 and if 'Max.Ownerships' is set, the activity waits for actors to take ownership or action until the 'Time Out For Ownership' is reached (if this is set).
- If none of the Ownership properties are set then the activity waits until the resource takes ownership or acts upon it.

Timeout for Ownership - This property specifies the time until which taking ownership is allowed i.e. the time until which the Checklist activity will wait for ownership.

- If Min.Ownerships is set and is not reached, the work item is removed from all Work Item lists.
- If Min.Ownerships is 0, it will check if ownership count is less than Max.Ownerships.
- If it is, the work item is removed from all Work Item lists. Else the work item will be removed from the Work Item lists of those who have not taken ownership. The work item will remain in the Work Item list of those who have taken ownership and wait for action from them.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout warning for Activity - This property specifies when a reminder has to be provided for the actors to act on the Checklist request, if the action count has not been reached. If properties such as Max.Ownerships, Affirmative Action for Checklist (AAA), and Action Limit (AL) are set, once the property values are reached then a warning is sent to the participant. When the timeout is reached, the following checks are made: If AL is set,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

when the action count i.e. the no. of Checklists (AC) < AL, then a warning is sent, if AL = 0 and AAA is set, and when AC < AAA, then a warning is sent to the participant.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Timeout for Activity - This property's value is used to decide until what time the activity should wait for the actors' action. If properties such as Max.Ownerships, AAA, AL are set, once the property values are reached the work item is removed from the Work Item lists of all the actors.

When the timeout is reached, the following checks are made:

- If AL is set, when the action count i.e. the no. of Checklists (AC) < AL, then the timeout for action is performed and the work item is removed from all Work Item lists.
- If AC > AL, it checks with the no. of Checklists or 'decision count' (DC) and AAA. If DC >= AAA (if it is set), the action is considered completed.
- If AAA = 0 and if DC >= AL, the action is considered completed.
- If AL = 0 and AC < AAA, then the timeout for action is performed and the work item is removed from all Work Item lists.
- If AL = 0 and AC > AAA, it will check with DC. If DC > AAA it is completed.
- If AL = 0 and AAA = 0, then if DC is at least 1, then it is considered completed.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Multiple Timeout Warning - This is used to set recurring timeout warnings that will be sent to the actors who have to take action. To set multiple timeout warnings, click the 'Multiple TimeOut Warning' button in the Properties pane for the Checklist activity. A new window pops up. In this window, set the interval for the recurring warning as well as the output message. You can also set the Recurring Time Interval. This can be used to set a different recurring time out after the first timeout happens. For example, if 'Time Interval' for the property is set as 7 days and 'Recurring Time Interval' is set as 2 days, then the first time out will happen after 7 days and there will be a recurring time out every second day after the seventh day (first) time out.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Steps to set the Multiple TimeOut Warning property

1. To set the multiple timeout warning, click the button for the 'Multiple TimeOut Warning' property to open the property window.
2. Enter the time interval for the timeout in the **Time Interval** field.
3. Select "Yes" from the drop-down for the **Recurring?** field, if you want the timeout warning to recur. Select "No" otherwise.
4. Enter the output message corresponding to the timeout warning in the **Output** field.
5. Enter the time interval for the recurring timeout in the **Recurring Time Interval** field.
6. After entering the content details, click **Add Row** to add this timeout setting and display a new row for entering another timeout setting.
7. Remove rows, if necessary, by selecting them and clicking **Remove Selected**.
8. After specifying the timeout settings, click **Update**.

See CheckList - Multiple Timeout Warning for a detailed description of the Multiple TimeOut Warning property window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Miscellaneous

Redirection Enabled - This property enables redirection. When this property is set to Yes, the workflow redirects you to the URL in the Redirect URL property in the next activity.

Activity outputs:

The Checklist Activity returns the following values -

Action Completed: Displayed when all the actors have taken part.

Action Completed - Affirmative Percentage Not Reached: Displayed when all the actors have not taken part.

Not Enough Resources to Take Acknowledge: Displayed when the filter condition did not yield enough actors to whom the request can be sent.

Timeout - Acknowledge: Displayed when the specified minimum number of actors did not take ownership of the request in the given time.

Timeout - Action: Displayed when some or all of the specified actors did not perform the required action in the given time.

Timeout Warning - Acknowledge: Displayed when a warning is generated that the specified minimum actors have not taken ownership of the request in the given time.

Timeout Warning - Action: Displayed when a warning for timeout is generated because some or all of the actors did not perform the required action in the given time.

Note: The following outputs will return only if you assign tasks to a queue user.

Resource Unavailable for Queue - If a queue is assigned to a task and if there are no resources available in the queue then this output will get triggered.

TimeOut for Queue Task - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting the completion of that task.

Timeout Warning -Queue - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting in sending a warning out put. However, the task will still be sleeping.

Timeout Warning -Queue Participant - This output will get triggered when the task in the queue is picked by user but still not acted on that task within the queue escalation time out settings resulting in triggering a warning output. However, the task will still be sleeping.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

CheckList - Delivery Channels

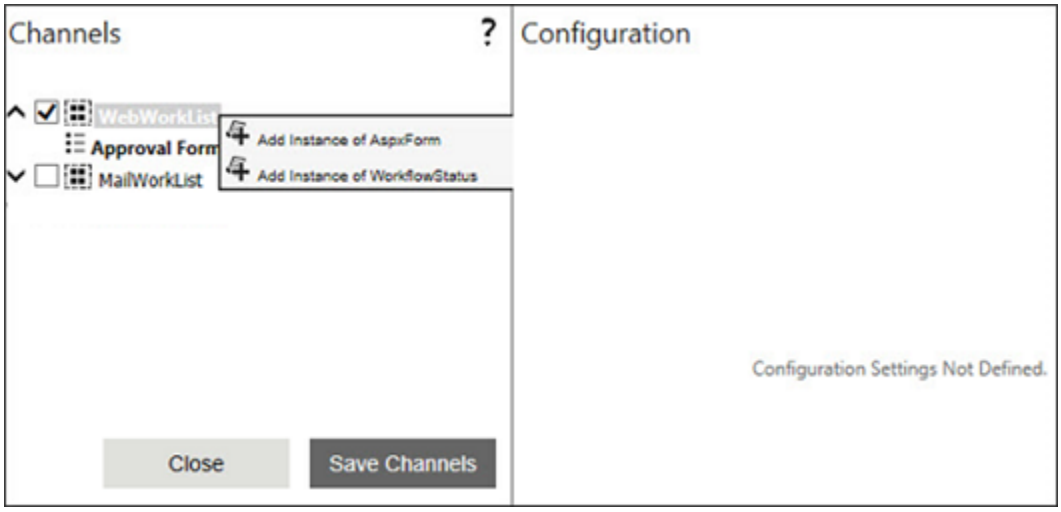
Purpose

The purpose of this chapter is to describe the feature of WorkflowStatus Form of Delivery Channels property under the category Configuration of any human activity available in AVEVA Work Tasks.

Description: This feature enables us to view the status of the workflow in the Work items list in the graphical form. [Optional]

1. Select the **Human Activity**.
2. Click the **Delivery Channels** Property.

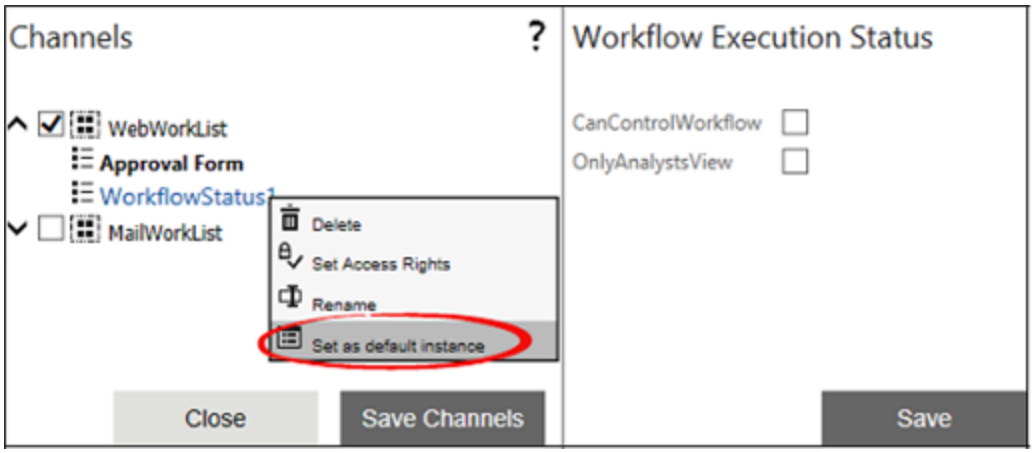
   The Property channel will be displayed as shown in the image below:



3. Right-click **WebWorkList** to display the context menu with the **Add Instance of Workflowstatus** option as shown below.

4. Set the Properties for the Execution View Status Graphical view.
5. Click **Save**.
6. If you want to set the WorkflowStatus form as the default instance, right-click **WorkflowStatus1**, and select **Set as default instance** from the context menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



7. Click the **Save and Save Channels** buttons.

Output

Now, when you click on any row in the Work items list, you will find this Execution view page being displayed as the detailed view, if you set it as a default instance.

Otherwise, Workflow Status form will appear in the context menu of each row in Work items list and you can open the execution view by selecting it from the context menu.

CheckList - Questions

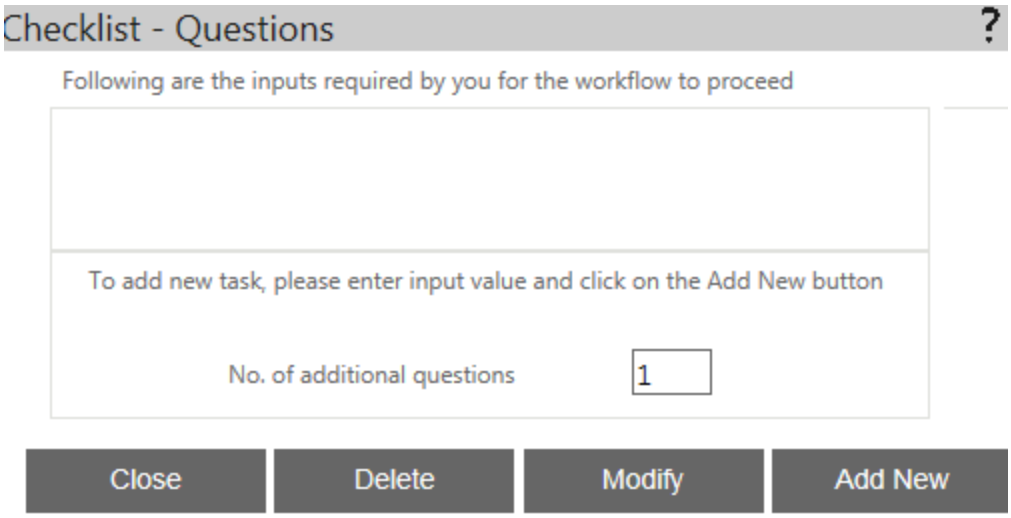
Questions Property Window

You can set the questions and answers that will form the checklist in this property window. To access this window, click the button in the 'Questions' property.

The window shown in the image below is used to Add, Edit and View the Questions and Answers added in the Checklist activity. This is the first window in the Questions property user-interface. It lists all the questions that have been added to the checklist. It has the No. of additional questions field to specify the number of questions to add. This window has the following buttons:

- **Add New** - You can click this button to add new questions. This will open the second window in the Questions user-interface where you can add the questions. The number of questions you can add will depend on the value set in the No. of additional questions field.
- **Modify** - You can click this button to edit the selected question(s). You can select a question for editing using the check box next to it.
- **Delete** - You can delete the selected question(s) by selecting them as described above and clicking this button.
- **Close** - You can click this button to close the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Checklist - Listing of Questions, Answers, Decision Maker, Mandatory Questions

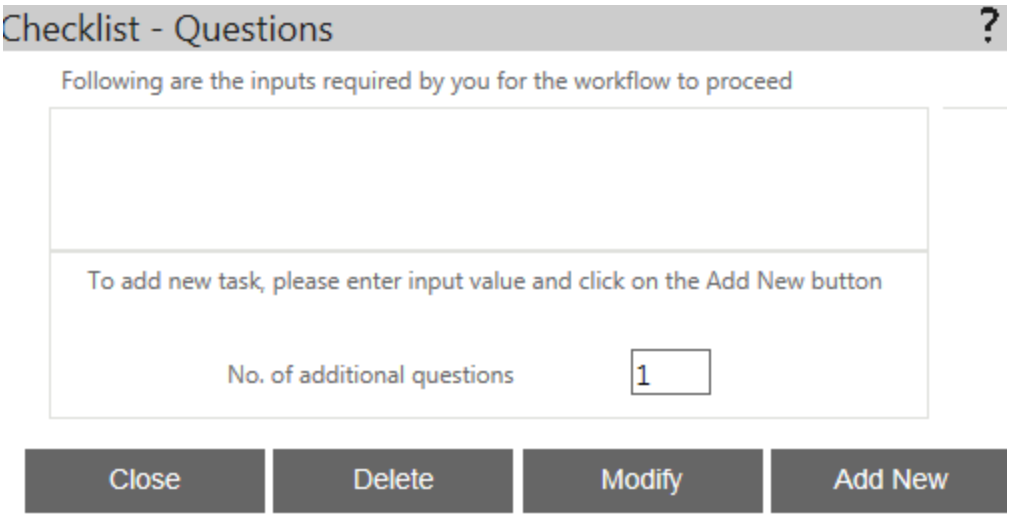
The window shown below is the second in the user-interface for adding questions to a checklist. This interface is displayed after the actor selects **Add New** or **Modify** from the first window shown above. You can add as many questions as were specified the first window and a separate row of fields is displayed for every new question. For each question, the following parameters can be set here:

- **Question** - You can enter the question in this text box.
- **Answer Type** - You can specify the answer type for this question by selecting an option from the drop-down. The options available are text box, drop down, memo, check box, and radio buttons.
- **No. of Options** - You can specify the number of options of the selected Answer Type in this text box.
- **!** - You can click the radio button in this column to specify whether this question is to be the Decision Maker or not. This option is displayed only when you select "drop-down" as the Answer Type. If set, the answer to this question will decide the outcome of the activity.
- **Mandatory*** - You can click the radio button in this column to make the corresponding question mandatory. If set, the checklist cannot be completed without giving an answer to this question.

This window has the following buttons:

- **Back** - Click this to go back to the first window
- **Next** - Click this to go to the next window
- **Close** - Click this to close the window

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

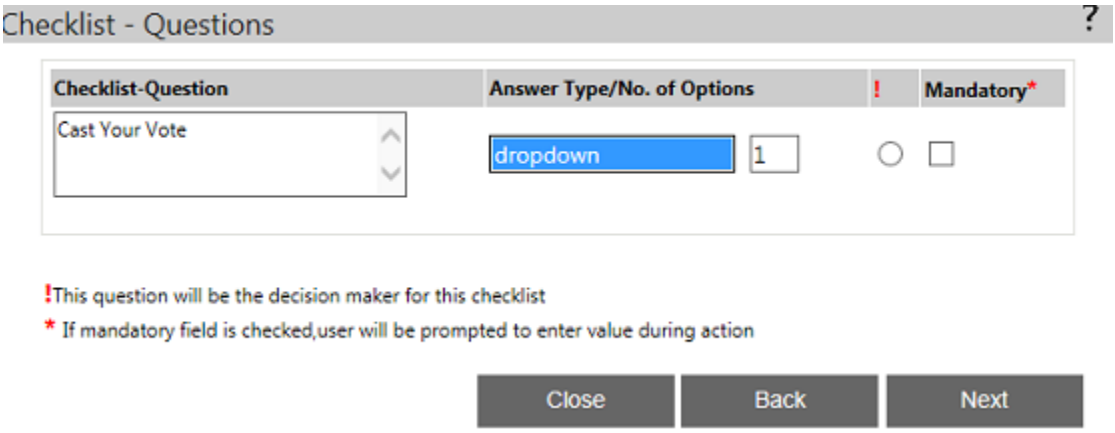


The window shown below is the third in the user-interface of the Questions property. In this window you can set the following parameters:

- **Options** - You can specify the options that will be presented to the actor to answer the question (if the question type is drop down, check box or radio button) in the fields in this column.
- **Default Value** - You can specify the option that will be the default value by clicking the corresponding radio button in this column.
- **Values** - You can specify the values that the answer will take internally in the fields in this column.
- **Map question answer to variable** - In this drop down you can select a variable to which you want to map the question's answer. The answers to this question will be stored in the selected variable and will be available for use in the workflow.

This window has the following buttons:

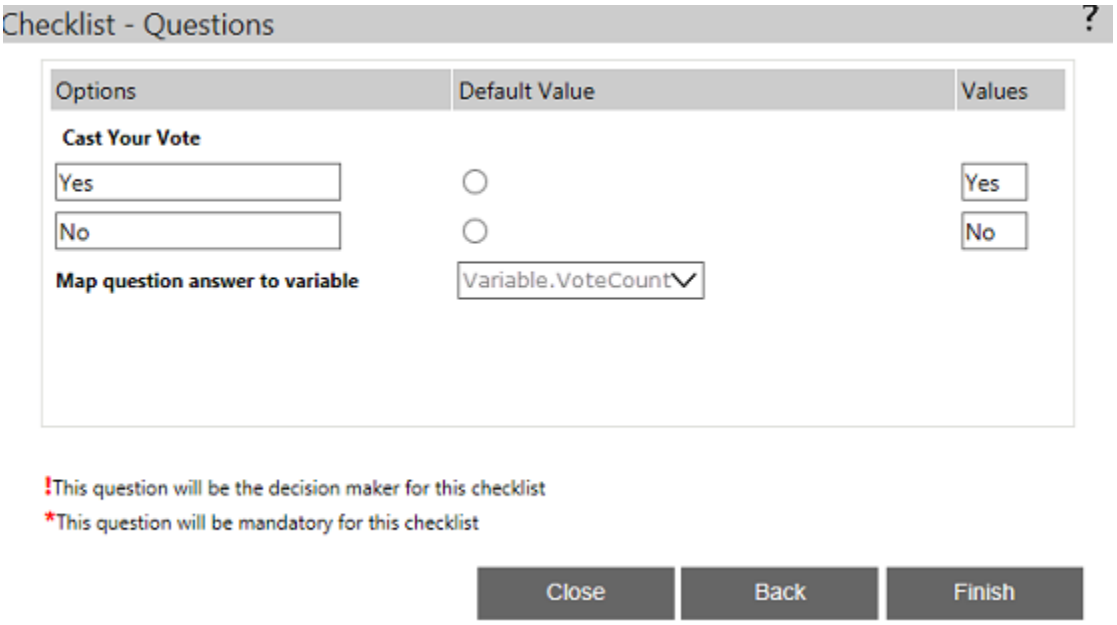
- **Back** - Click this to go back to the first window
- **Finish** - Click this button after completing all the details to save the question
- **Close** - Click this button to close the window.



The window shown below is the last in the sequence and is displayed when you click **Finish** in the previous

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

window. It is identical to the first window with the difference being the newly added question that is displayed here. You can modify or delete the new question or add a new question. If you do not wish to perform any more operations, click **Close** to exit this user interface.

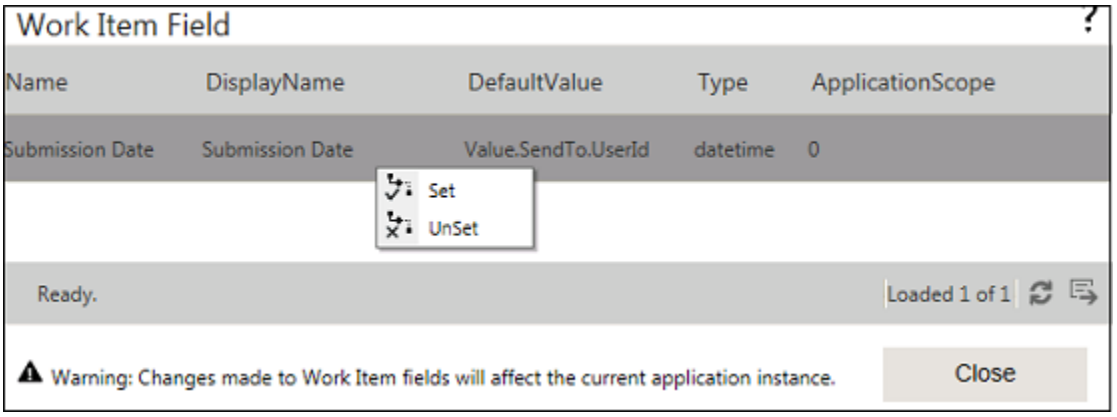


Note: Do not use hash (#), colon (:), and semicolon (;) in the Questions. Also, avoid multi-line questions.

CheckList - Work Item Fields

Work Item Fields Property Window

You can set the Work Item Fields for this activity in this property window. To access this window, click the button in the 'Work Item Fields' property.



The window has a grid listing all the work item fields that have been set for the process.

- Each row in the grid represents a work item field. You can click a work item field to select it.
- The context menu has the Set WorkItemField option, which you can use to set the selected work item field for this activity. It also has the UnSet WorkItemField option to remove the work item field setting.
- **Close** - You can use this button to close the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: While entering a value in the Work Item field, it is recommended to keep it less than the specified database column size. If there is a need to increase the size of the Work Item field, the user should manually increase the column size in the database.

CheckList - Assign Actor(s)

Assign Actor(s) Property Window

You can set the actors to whom the CheckList work item is to be sent in this property window. To access this window, click the button in the 'Assign Actor(s)' property.

See Assigning a Task to a Manager for more details.

The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

Different Methods of Assigning an Actor

- Assigning an Actor by building an expression:

Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

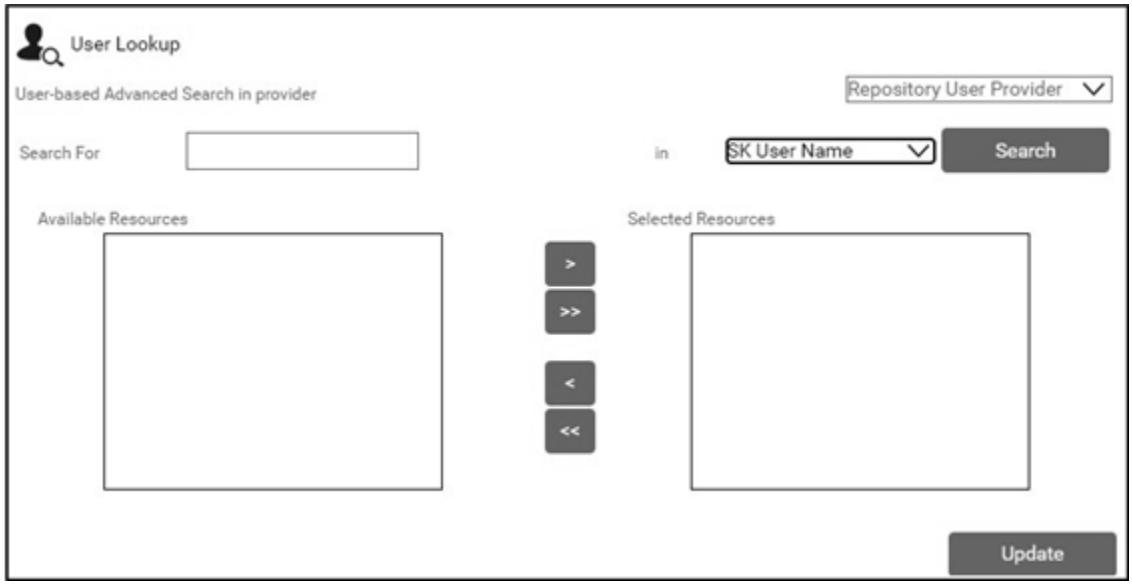
At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

- Assigning an Actor through User Lookup

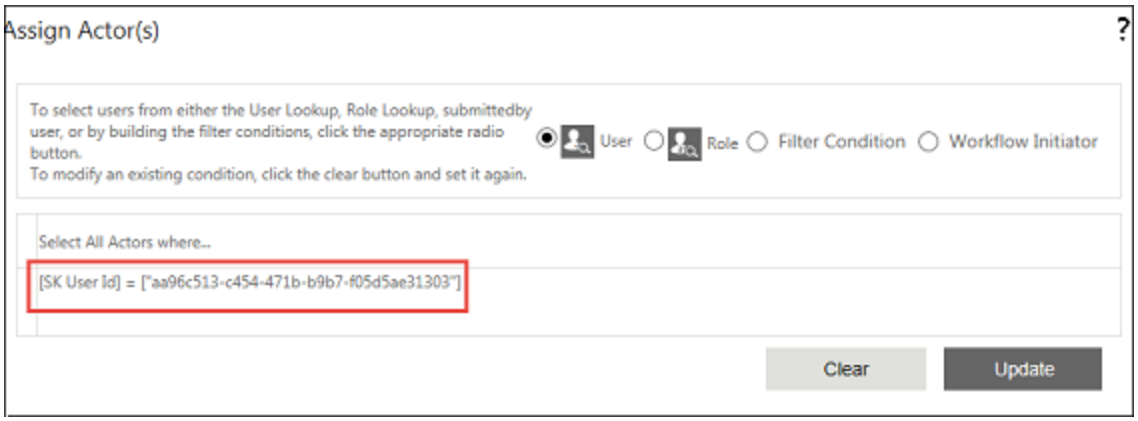
Through the **User Lookup** you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.
2. The **User Lookup** dialog is displayed, search and select the required actor.



3. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
4. In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



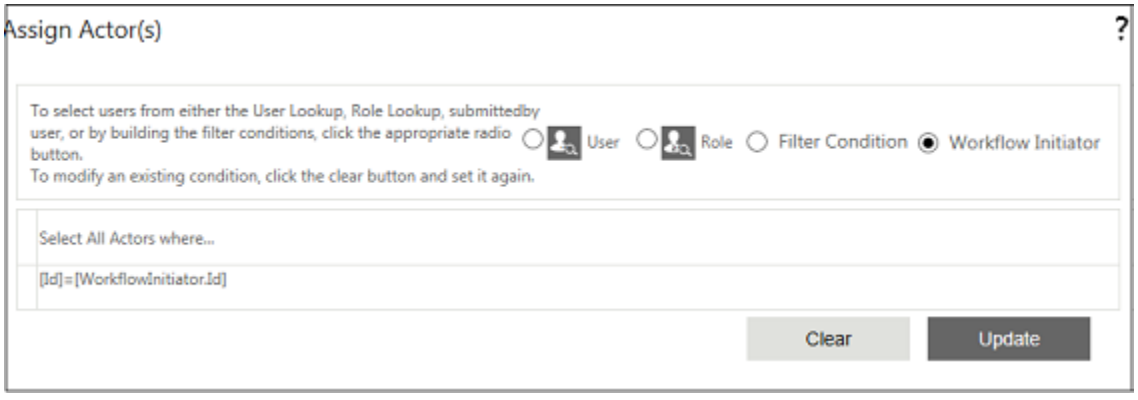
5. Click **Update** in the **Assign Actor(s)** window to update the selected actor(s) in the **Assign Actor** property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

- Assigning an Actor through Role Lookup

Through the Role Lookup you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

1. Select the **Role Lookup** option.
2. The **Role Lookup** dialog is displayed, search and select for the required role.
3. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
4. In the **Assign Actor** window, you will see that an expression is displayed.
5. Click **Update** in the Assign Actor(s) window to update the **Assign Actor** property.

- **Assigning an Actor through Workflow Initiator**
  Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:



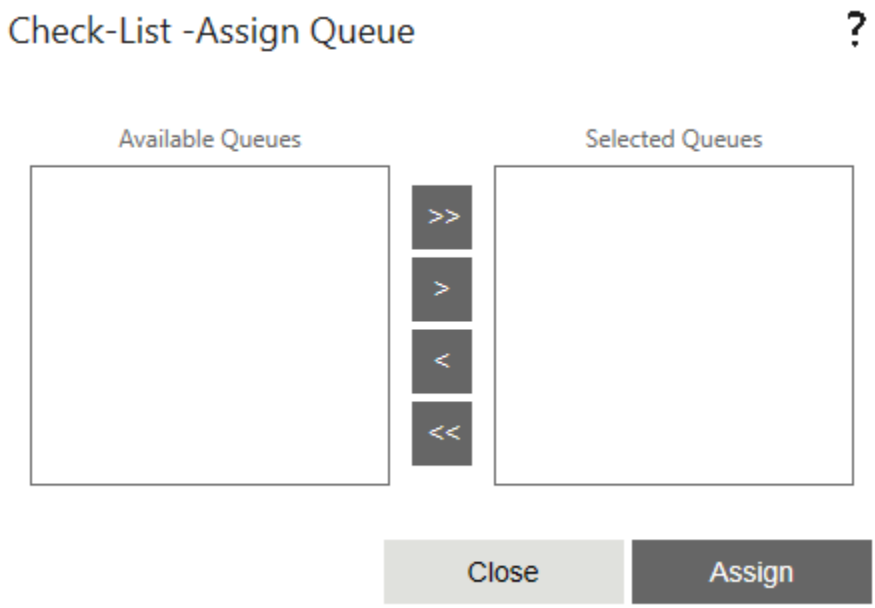
CheckList - Assign Queue(s)

Assign Queue(s) Property Window

You can select the queues to which the CheckList work item is to be sent in this property window. To access this

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

window, click the button in the 'Assign Queue(s)' property.



The window has a pair of list boxes to specify the queue(s). The **Available Queues** list shows all the available queues. You can select the required queue(s) from this list box. The **Selected Queues** list shows all the selected queues to which the activity will be assigned.

You can use the following buttons located between the list boxes to move the queues to and from the list boxes.

- [>>] - Click this button to add all the queues in the **Available Queues** list to the Selected Queues list
- [>] - Click this button to add the selected queue(s) from Available Queues list in to the Selected Queues list
- [<] - Click this button to add the selected queue(s) from the Selected Queues list to the Available Queues list
- [<<] - Click this button to add all the queues in the Selected Queues list to the Available Queues list

At the bottom of the window you will find the following buttons:

- **Assign** - Click this button to assign the activity to the selected queues
- Close - Click this button to close the window

Assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the **Start** activity with the Type as **String**. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

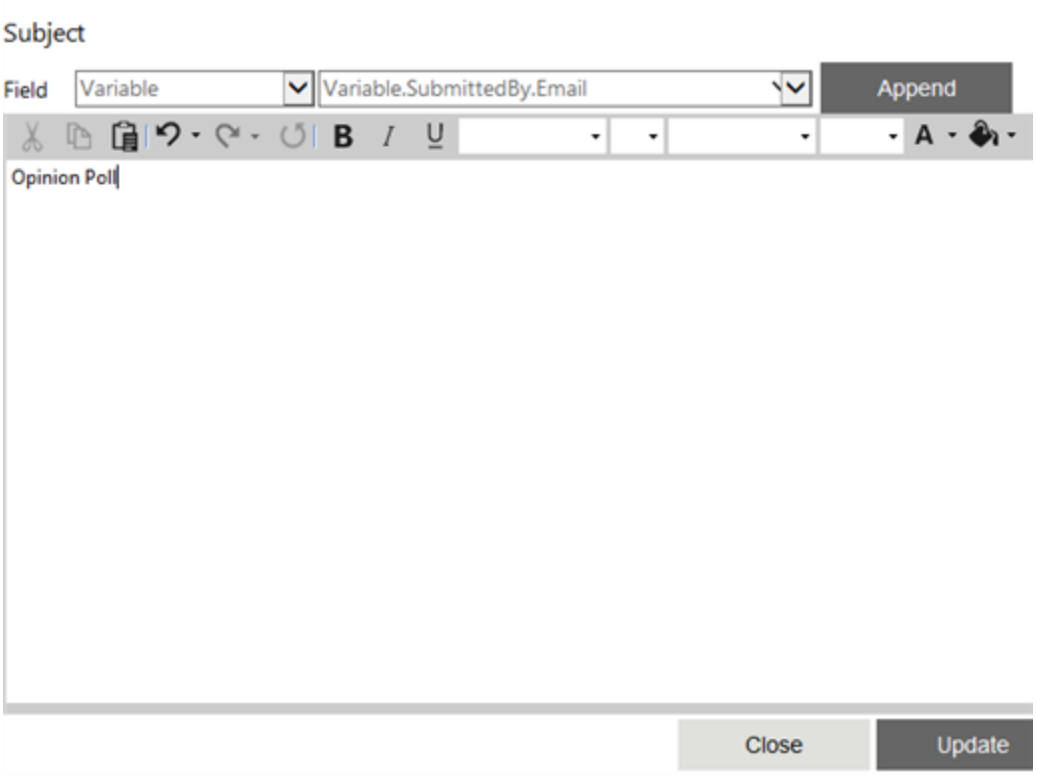
To assign multiple queues, specify the queue-ids with semicolon (;) separator. **Skelta.HWS.Queue.QueueCollection** class can be used to retrieve queue information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

CheckList - Subject

Subject Property Window

You can specify a subject for the CheckList work item assignment in this property window. To access this window, click the button in the 'Subject' property.



The window has a text editor to enter the subject. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the subject. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the subject in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the subject.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the subject using a combination of discrete value and variable or content. For example, you can specify the subject as "Submitted By <%#Variable.SubmittedBy.Email%>". Here, the string "Submitted By" should be typed directly in the text editor. The second half of the subject is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the **Field** drop down and clicking **Append**.

The text tool bar above the text editor allows you to manage and format the subject. It has the following text tools –

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text
- At the bottom of the window you will find the following buttons:
- **Update** - Click this button to save the subject
- **Close** - Click this button to close this window

CheckList - Body

Body Property Window

You can specify a message Body for the CheckList work item assignment in this property window. To access this window, click the button in the 'Body' property.

The window has a text editor to enter the body text. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The Value text box/drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the body using a combination of discrete value and variable or content. For example, you can specify the body message as "Hi, Please cast your vote in the poll conducted by <%#Variable.SubmittedBy.name%>". Here, the string "Hi, Please cast your vote in the poll conducted by" must be typed directly in the text editor. The second half of the body is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the Field drop down and clicking Append.

The text tool bar above the text editor allows you to manage and format the body text. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

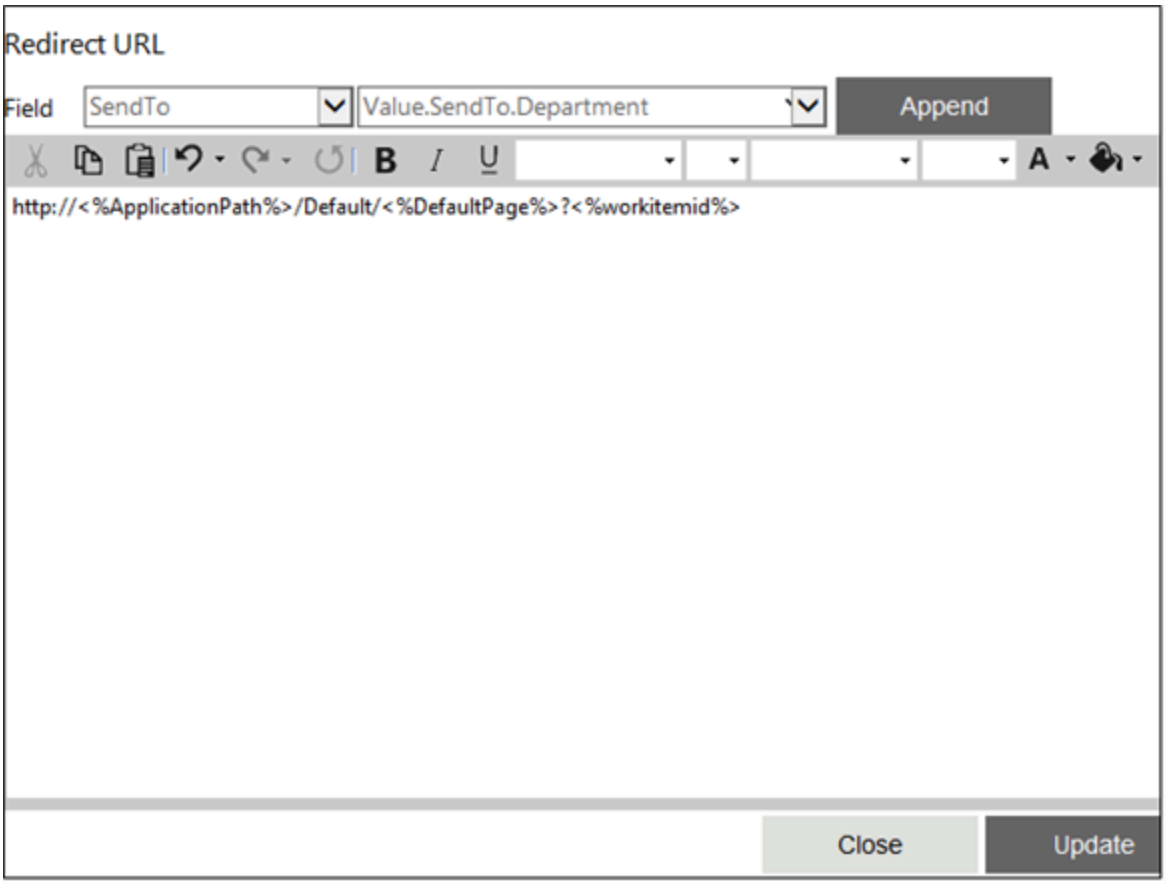
- **Update** - Click this button to save the body message
- **Close** - Click this button to close this window

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

CheckList - Redirect URL

Redirect URL Property Window

You can specify a message Redirect URL for the CheckList activity in this property window. To access this window, click the button in the 'Redirect URL' property.



The window has a text editor to enter the Redirect URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Redirect URL in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Redirect URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

  The text tool bar above the text editor allows you to manage and format the Redirect URL. It has the following text tools
- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

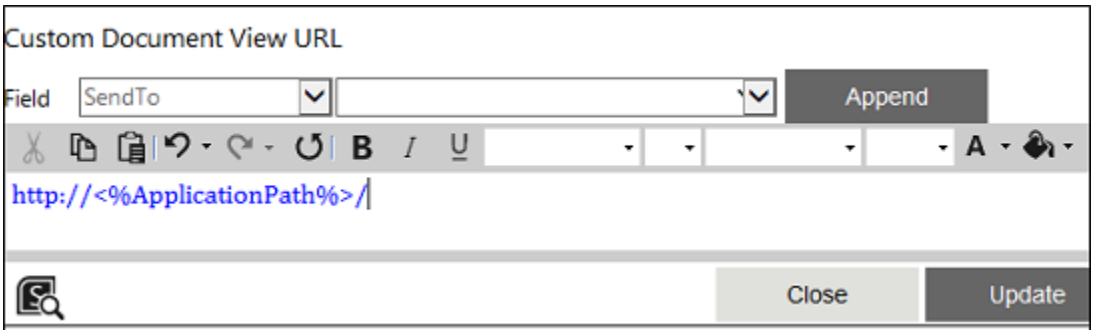
- **Update** - Click this button to save the Redirect URL
- **Close** - Click this button to close this window

CheckList - Custom Document View URL

Custom Document View URL Property Window

You can specify a message Custom Document View URL for the CheckList activity in this property window. To access this window, click the button in the 'Custom Document View URL' property.

For more details, see Custom Document View URL



The window has a text editor to enter the Custom Document View URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

or XMLVariables here.

- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Custom Document View URL in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Custom Document View URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Custom Document View URL. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - Click this button to save the Custom Document View URL
- **Close** - Click this button to close this window

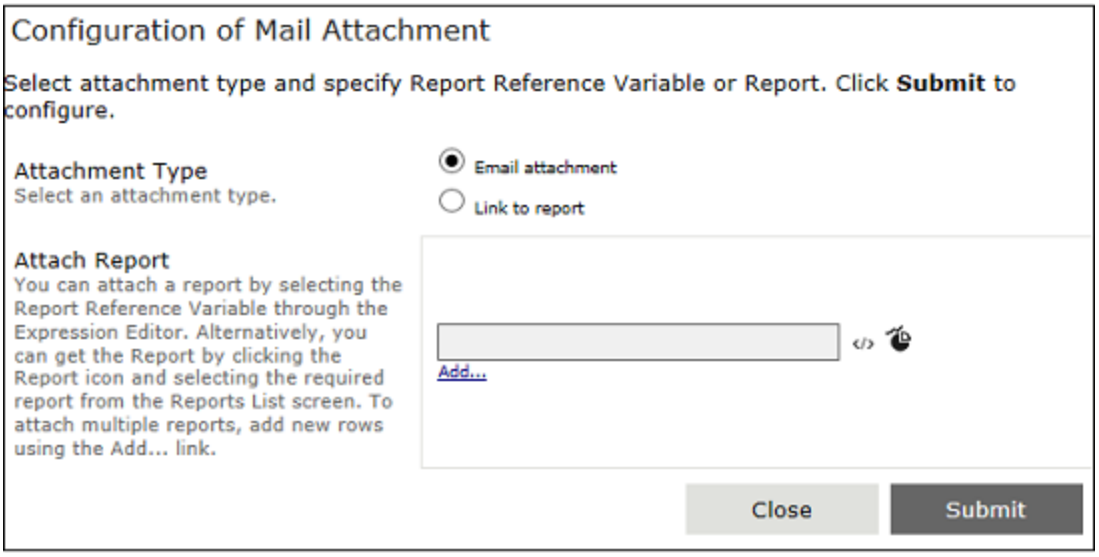
CheckList - Attach Reports

Configuration of Mail Attachment property window

You can configure the **Attachment** property for the **Approval** activity in this property window.

In this property window, you can configure the report attachments to be emailed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



- **Attachment Type:** Use this property to select the type of attachment for sending the report as an email.
  - **Email attachment:** Select this type to send the report as an email attachment.
  - **Link to report:** Select this type to embed a link in the email to download the configured report.
- **Attach Report:** Use this property to attach a report generated by the **Report Generator** activity within the workflow or to select a report from the list of generated reports.

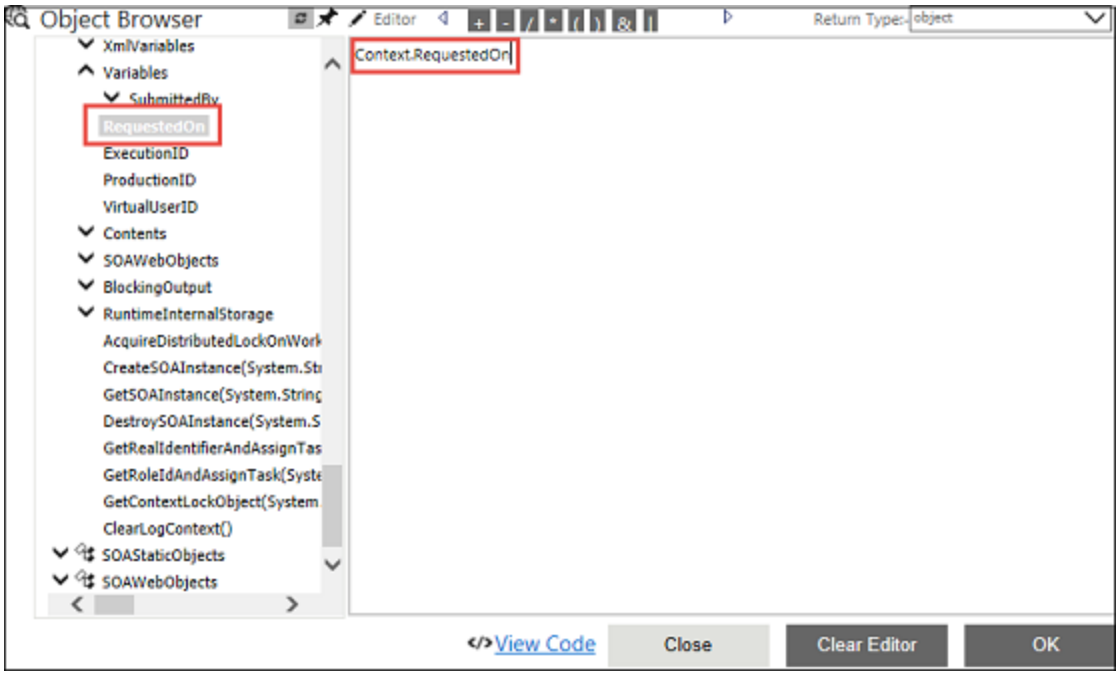
  You can attach a report by selecting the required **Report Reference Variable** using the **Expression Editor**. Alternatively, you can select the report from the Reports List window which consists of the already generated reports.

There must be at least one **Report Generator** activity before an **Approval** activity in the workflow to select the **Report Reference Variable** (using Expression Editor) as an **Attachment**. The **Report Reference Variable** is generated while configuring the **Report Generator** activity. By default, the report reference variable is **[Template Name]_[activity Name]**. For more information about **Report Reference Variable**, see Configure Report Template.

To configure mail attachment

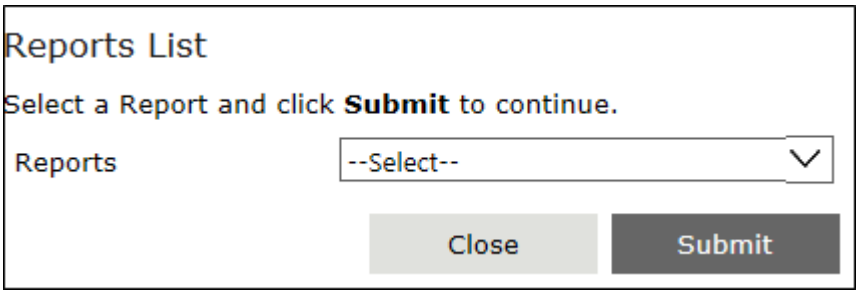
1. Click the edit button for the **Attachment** property.
2. Select the required **Attachment Type**.
3. In the **Attach Report** text box, you can either use the expression editor to get the variable of the report or use the **Reports List** to select the report.
4. To get the variable, click the **Expression Editor** icon given beside the text box to select the required **Report Reference Variable**.
5. In the **Expression Editor** window, select the required variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



6. Click **OK.**
7. Click **Close**.

   The variable appears in the **Attach Report** text box.
8. To select the report from the Reports List dialog box, click the **Reports** icon.



9. In the Reports List dialog box, select the required report, and then click **Submit**.

   This window closes automatically.
10. Click **Close** to close the configuration window.

CheckList - HTML Template

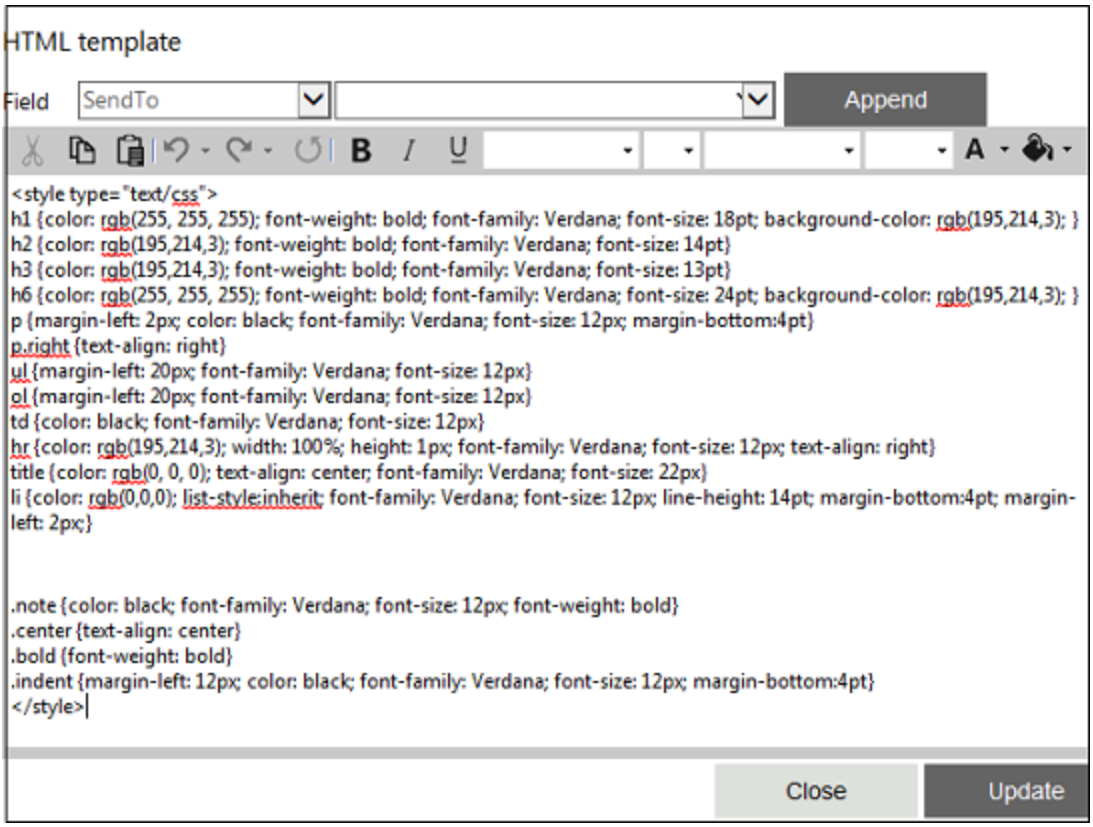
HTML template Property Window

You can specify a html template for the CheckList activity in this property window. To access this window, click the edit button in the 'HTML template' property.

In this property window, you can write your own html template for Check-List activity or you can use the default Check-List activity template for customizing the email.

See Using 'HTML Template' property of Check-List Activity for more information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the body text. You can do this either by entering a html template directly in the text editor or by using the Field controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop-down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop-down instead of the field. This drop-down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the html template. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

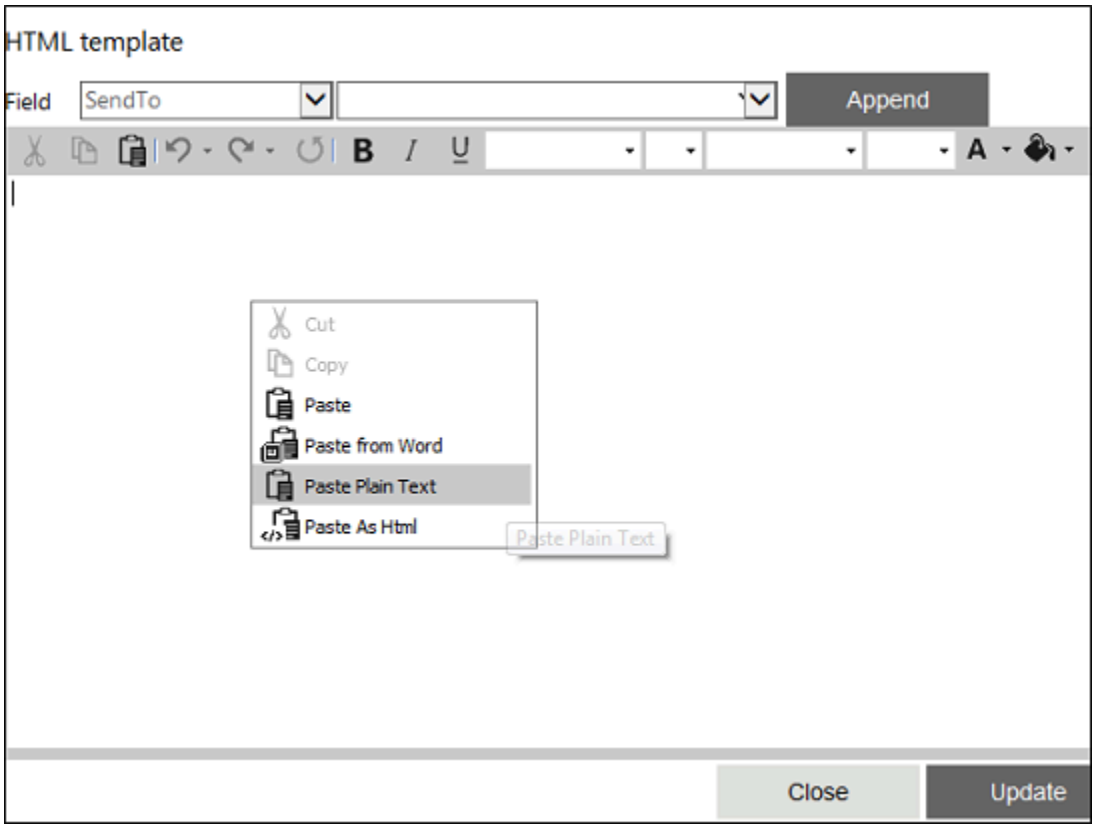
- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

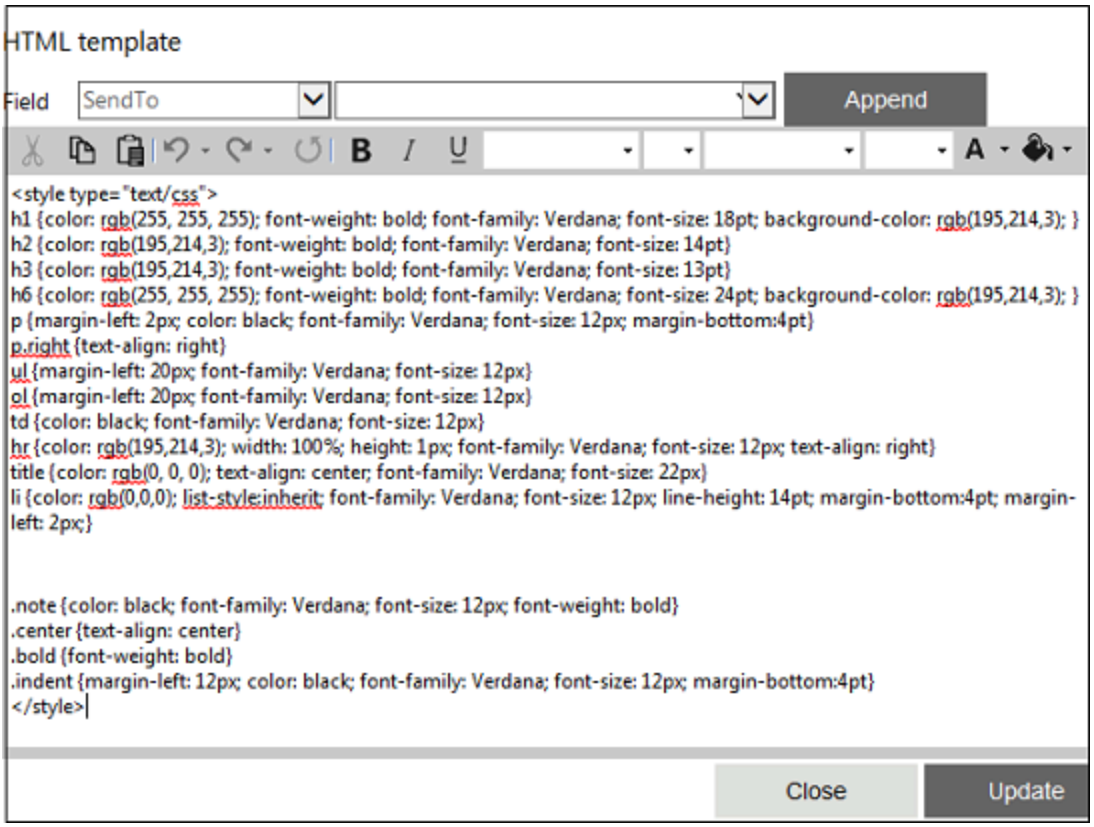
- **Update** - Click this button to save the HTML template
- **Close** - Click this button to close this window

To set this property, do the following steps:

1. To specify the HTML template, click the edit icon to open the property window.
2. In the **HTML template** window, provide template which should be in html format. If you are pasting a copied html content, then right-click the **HTML template** window and paste it as "**Paste Plain Text**".



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



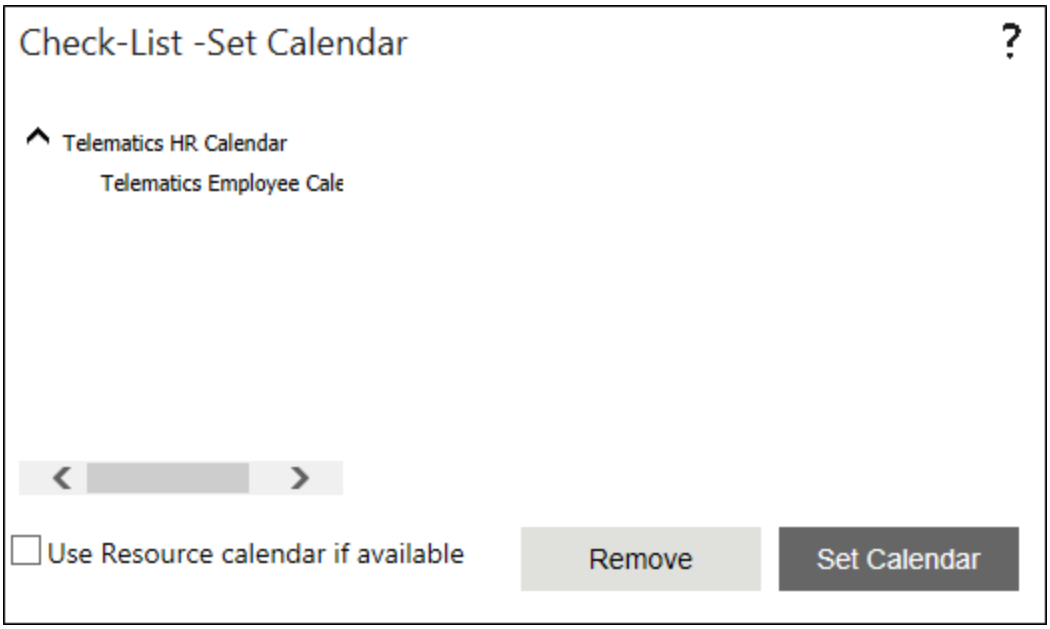
3. Click **Update**.

CheckList - Set Calendar for Timeout

Set Calendar for Timeout Property Window

You can set a Calendar to calculate the timeout for the Checklist activity in this property window. To access this window, click the button in the 'Set Calendar for Timeout' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



This window lists the available calendars in a tree structure, with lower-level calendars nested within the respective top-level calendar. You can click a calendar to select it for setting.

There are two buttons in this window:

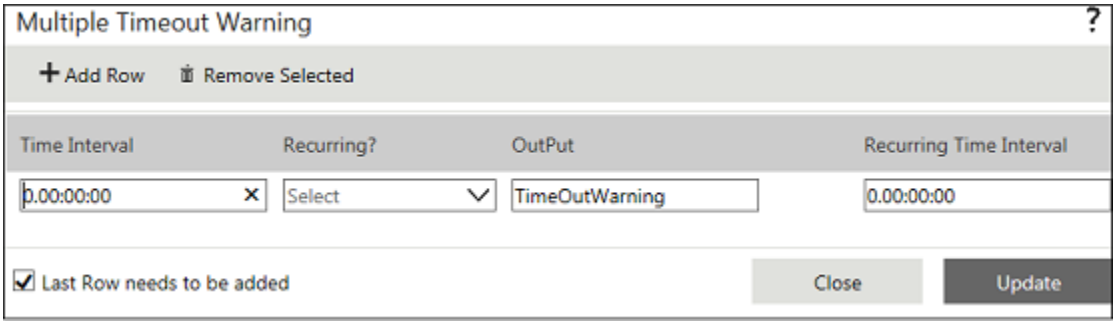
- **No Calendar** - Click this button to indicate that there is no calendar set for the activity
- **Set Calendar** - Click this button to set the selected calendar

The **User Resource calendar if available** check box can be selected to enable the use of resource calendars if none are displayed in this window for selection.

CheckList - Multiple Timeout Warning

Multiple TimeOut Warning Property Window

You can set the Multiple Timeout Warning in this property window. To access this window, click the button in the 'Multiple TimeOut Warning' property.



The window has two tabs at the top to manage the rows of timeout settings in the grid.

- **Add Row** - You can click this tab to add the newly entered timeout setting and display a new row to the grid where you can add another timeout

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Remove Selected** - You can click this tab to remove the timeout setting row selected in the grid.

The grid has the following parameters that you need to specify for the new timeout setting.

- **Time Interval** - You can use this field to enter the time interval in which you can generate a timeout warning. You must enter the time in d.hh:mm:ss format.
- **Recurring?** - You can use the drop down in this field to indicate whether the timeout is recurring or not. You can select "Yes" to set the timeout warning to recur.
- **Output** - You can enter the output message corresponding to the timeout warning here.
- **Recurring Time Interval** - You can set the time interval in which the timeout warning has to recur here. Enter the time in d.hh:mm:ss format.

There are two buttons and a check box at the bottom of the screen:

- **Update** - Click this button to save the newly added timeout settings
- **Close** - Click this to close the property window without saving the newly added timeout settings
- **Last Row needs to be added** - Select this check box to save the last row added

Choice Activity

Activity Description

The Choice Activity is used in the AVEVA Work Tasks workflow process design in cases where a task needs to be assigned with multiple options for an action. This activity can be configured with buttons, with the provision to set the output of each as task completion makers. A maximum of ten such buttons can be configured.

Application Scenarios

To illustrate the functionality of the Choice activity, consider a scenario in which the approval activity in a process requires a provision to send back the approval request in addition to the standard responses of approval and rejection. In this case, the Choice activity can be used to create a customized approval activity with an additional button named 'Send back' in addition to the 'Approve' and 'Reject' buttons. Without this activity, a custom activity would have to be created to achieve this functionality.

Workflow Variables for Choice

Steps to define variables for use in the Choice activity

This section describes how to define variables for use in Choice activities. It gives the syntax for creating the variables and typical examples for different cases.

Variables for capturing WorkItemIds

The Work Item Id's for the Choice work item can be retrieved using a variable of type array, with the following syntax:

'^'+CurrentActivity.Name+'WorkItemIds'.

For example, **^CandidateApprovalWorkItemIds**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Variables for capturing All Actors' Details

The details of the actor(s) to whom the Choice work item gets assigned can be accessed by declaring a variable of type array with the following syntax:

'^'+CurrentActivity.Name+'ActingUsers'

For example, **^CandidateApprovalActingUsers**.

Variables for capturing Responding Actors' Details

The details of the actors' responding to the Choice work item can be accessed by declaring a variable of type resource with the following syntax:

'^'+HWSActivity.ActivityName+'RespondingUser'

For example, **^CandidateApprovalRespondingUser.**

This variable stores the responded user information. If it is more than one user, it will retrieve the last user details.

Note: For single user, you can declare this variable as **Resource** type. If task is assigned to more than one user and to get all responded users information, then declare this variable as array.

Variables for capturing Comments

Comments entered in a Choice work item can be retrieved

using a variable with the following syntax:

'^'+_HWSActivity.ActivityName+'CommentChoiceStatus'.

For example, **^CandidateApprovalCommentChoiceStatus**.

Variables for assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the Start activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

To assign multiple queues, specify the queue-ids with semicolon (;) separator. Skelta.HWS.Queue.QueueCollection class can be used to retrieve queue information.

Variable for Queue Acting Users

You can declare an array type variable to get updated information about the name of the actor to whom an activity gets assigned from a queue.

In the Workflow Variables, define the variable in the following format:

^+CurrentActivityName+'QueueActingUsers'

Ex: ^HRFormQueueActingUsers

HRForm is the activity name.

For example, the value of the user who acts on the HRForm activity will be stored in the variable. The Approval1 task will be assigned to the same user.

Variable for storing timeout warning value

Suppose if you want to store the timeout warning value for Choice activity in your workflow. The naming convention for the system variable to be declared is

^<ActivityName>ResponseByWarningTimeout

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The value returned by the variable will be in UTC format. For multiple user scenarios the highest timeout value calculated for a user will be available in the variable.

Note: Multiple timeout warnings cannot be tracked using this feature.

Activity Properties

The **Choice** activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right-click menu for the Choice activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Display Name - This property specifies name to be displayed for the action in the Work Item List.

Property Type: Optional (This property need not be set if the display name is already configured in the Actions.xml. The name entered here overrides any display name that is configured for this activity in the Actions.xml file.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Delivery Channels - This property is used to specify the delivery channels to deliver this activity to the assigned actor(s). Multiple channels can be specified giving the actor(s) different options to act on the work item.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Delivery Channels Property

1. To select the Delivery Channels, click the button for the 'Delivery Channels' property to open the property window.
2. In this window, the Delivery Channels that have been configured are listed on the left along with a check box against each channel. Click and select the check-box corresponding to the channel you wish to select. You can select multiple channels by selecting the respective check-box.
3. Click on Save Channels button.

See Choice - Delivery Channels for a detailed description of the Delivery Channels property window.

Action Outputs - This property is used to specify the outputs of the Choice work item assignment. These outputs can be specified as buttons, which will be displayed in the Work Item list of the recipient. To assign activity outputs, click the 'Activity Outputs' button in the Properties pane for the Choice activity. A new window pops up. In this window, enter the name of each button and indicate whether it is a completion maker or not.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Mandatory (This property must be set for the activity to be used in the workflow.)

Steps to set the Action Outputs property

1. To specify the Action Outputs, click the button for the 'Action Outputs' property to open the property window.
2. Enter the button name in the **Button Name** field.
3. Select "Yes" from the drop down for the **Completion Maker** field to set the button output as the completion maker of the Action Output. Select "No" if you do not want the action output.
4. After entering the output details, click **Add Row** to add this variable and display a new row for entering another output.
5. Remove rows if necessary by selecting them and clicking **Remove Selected**.
6. After entering all the variables, click **Update**.

See Choice - Action Outputs for a detailed description of the Action Outputs property window.

Work Item Fields - This property is used to add the Work Item Fields to be used to display application data as well as value in variables and content in the Work Item list when this activity is executed.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Work Item Fields property

1. To select the Work Item Fields, click the button for the 'Work Item Fields' property to open the property window.
2. In this window, the Work Item Fields that have been set are listed in a grid. Click a row to select the corresponding Work Item Field.
3. Right-click the row and select the "SetWorkItemField" option from the context menu to set it.

See Choice - Work Item Fields for a detailed description of the Work Item Fields property window.

Actors

The properties in the Actors group have been described below. You can use these properties to specify the actors who will work on the activity and related properties.

Assign Actor(s) - This property specifies the actor(s) who has to take up the Choice activity. It is tagged with the Ignore Unavailable Resources property which in turn is tagged with Alternate Resource Allowed. This will be explained in detail later.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any actor(s).)

Steps to set the Assign Actor(s) property

1. To specify the actors, click the button for the 'Assign Actor(s)' property to open the property window.
2. In this window build an expression to select the actors using the actors properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Choice - Assign Actor(s) for a detailed description of the Assign Actor(s) property window.

See the Using XML Variables section in the About the XML Variable User Interface topic for details on how to use the XMLVariables to build the expression.

Assign Queue(s) - This property specifies the queue(s) to which the Choice activity is to be assigned.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any Queue(s).)

Steps to set the Assign Queue(s) property

1. To specify the queue(s), click the button for the 'Assign Queue(s)' property to open the property window.
2. In this window select the queue(s) from the **Available Queues** list box. You can select multiple queues by selecting with the <Ctrl> key pressed.
3. After selecting the required queue(s) click on the [>] button to move it to the **Selected Queues** list box. You can click the [>>] button to select all the queue(s) and move them to the **Selected Queues** list box.
4. You can remove any queue from the selected list by selecting it and clicking the [<] button. You can remove all the queues by clicking the [<<] button.
5. After adding the required queue(s), click **Assign** to assign the activity to the selection.
6. Click **Close** to close the window without saving the selection.

See Choice - Assign Actor(s) for a detailed description of the Assign Queue(s) property window.

Min. Ownerships [% or Number] - This property, which can be specified as a number or percentage, is the minimum number of actors who should take ownership of the Choice request. This property is related to the 'Timeout warning for Ownership' and 'Timeout for Ownership' properties. To illustrate, when 'Timeout for Ownership' is reached, the activity first checks if the ownership count is less than Min.Ownerships (if it is set). If it is, the work item will be removed from all Work items lists. If the 'Min.Ownerships' has already been satisfied, then the work item will be removed from the Work items lists of those who have not taken ownership. If the Min.Ownerships value is 0, then the activity checks if the ownership count is less than Max.Ownerships (if it is set). If it is, the work item will be removed from all Work items lists. Else the work item will be removed from the Work items lists of those who have not taken ownership. If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work items lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Max. Ownerships [% or Number] - This property, which can be specified as a number or percentage, is the maximum number of actors who should take ownership of the Choice request. If 'Timeout for Ownerships' is specified and achieved, and the 'Min.Ownerships' is also satisfied, then the activity waits for action from those who have taken ownership. If the Min.Ownerships value is set to 0, then it checks if the ownership count is less than Max.Ownerships. If it is, the work item will be removed from all Work items lists. Else the work item will be removed from the Work items lists of those who have not taken ownership. If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work items lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

scenarios.)

Enable Take Ownership? - If this property is set to **Yes**, then the actor is provided with a **Take Ownership** option in the Work items list to take ownership of the task. If you are not required **Take Ownership** option in the Work items list, then set this property as **No**.

Note: By default, this property is set as **Yes**.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Enable Take Ownership? property

Select **Yes** or **No** from the **Enable Take Ownership** drop-down list.

Re-Assign Allowed? - If this property is set then the actor is provided with a Re-Assign option. Using this the approver can re-assign the work item to another actor.

Property Type: Optional (This property needs to be set only if necessary.)

Ignore Unavailable Actors? - This property provides the following options - **Yes**, **Yes** - holidays excluded and No. If this property is set to 'No', when the Approval activity is assigned to the actor's Work item list, a check is first done to see whether the actor is available. If the actor is not available then a check is made for an alternate actor assigned to the unavailable actor. Simultaneously the value of the property 'Alternate Actor Allowed' is checked. If that is also set, then the Approval work item is assigned to this actor. If this property is set to **Yes** - **holidays excluded**, the workflow engine will ignore the fact that actors are not available if the day is an organization holiday and the output Not Enough Resources will not be generated.

Property Type: Optional (This property needs to be set only if necessary.)

Alternate Actor Allowed? - This property is set to assign a particular task to an alternate actor in the absence or unavailability of any actor.

Property Type: Optional (This property needs to be set only if necessary.)

Notification Contents

The properties in the Notification Contents group have been described below. You can use these properties to specify the content used to notify actors about the activity.

Subject - This is the subject of the Choice work item. The subject can be customized for each actor.

Property Type: Optional (This property need not be set. If it is not set, the default subject line from the Actions.xml file is displayed.)

Steps to set the Subject property

1. To specify the subject, click the button for the 'Subject' property to open the property window.
2. In this window enter the subject text directly in the text editor.
3. If you want to customize the subject you can do so using the **Field** controls to add resource property values. For example, to add a subject that includes the email id of the actor submitting the approval request, you can do as follows: Type in the text "Submitted By". Select "Variable" and "Variable.SubmittedBy.Email" from the two drop downs for **Field** and click **Append**. This will add the placeholder for the variable after the text entered in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the email id.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the font or size, set text color or background color.

6. After adding the subject, click **Update** to save it.
7. Click **Close** to close the window without saving the subject.

See Choice - Subject for a detailed description of the Subject property window.

Body - This is the actual message body of the Choice sent to the actor(s). The body can be customized for each actor. There is an **Enterprise Console Explorer** (plug-in) icon in the Body property window to get the work item details URL. See Enterprise Console Control (EC Plug-in) for more information.

Property Type: Optional (This property need not be set.)

Steps to set the Body property

1. To specify the message body, click the button for the 'Body' property to open the property window.
2. In this window, enter the body text directly in the text editor.
3. If you want to customize the body you can do so using the **Field** controls to add resource property values. For example, to add a body that includes the name of the actor submitting the approval request, you can do as follows: Type in the text "Hi, Please cast your vote in the poll conducted by". Select "Variable" and "Variable.SubmittedBy.name" from the two drop downs for Field and click **Append**. This will add the placeholder for the variable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the name.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the body, click **Update** to save it.
7. Click **Close** to close the window without saving the body.

See Choice - Body for a detailed description of the Body property window.

From Email Address? - This property is used to specify the email address from which notification messages will be sent for the activity.

Property Type: Optional (This property needs to be set only if necessary. If it is not set, the default value in Activities.xml will be used.)

Hide Responses? - If this property is set to 'No', then actors can see the responses of other actors in a group Choice scenario. To view these responses i.e., ownerships, choices or rejections, select the 'Take Ownership' option. This will open a new pop up with the responses of other actors who have taken ownership.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Priority - This property is set to indicate the priority of the Choice request. Enter the values 0-33 to set as Low, 34-66 for Medium, or 67-99 for High. The default value is set to 34.

Property Type: Optional (This property needs to be set only if necessary.)

Show Custom Document View Window - If this property is set to 'Yes', then the Custom Document View Window will be shown to actors in their Web Work Item Lists. The data posted to the page are application, workflow, executionid, executiondetailsid, and workitemid. The posted data can be retrieved as Request.Form["workitemid"].

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Send Notification Email - If this property is set to 'Yes', then notification emails will be sent to the actors. If it is set to 'Yes-HTML', then the emails will be sent in HTML format.

Property Type: Optional (This property needs to be set only if necessary.)

Notify if alternate actor is considered - If this property is set to 'Yes', then notification emails will be sent to the actors when an alternate actor is considered.

Property Type: Optional (This property needs to be set only if necessary.)

Redirect URL - This property is used to get the URL of the web page to redirect the client to, in the case of synchronous execution of the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Redirect URL property

1. To specify the Redirect URL, click the button for the 'Redirect URL' property to open the property window.
2. In this window, enter the Redirect URL directly in the text editor.
3. If you want to customize the Redirect URL you can do so using the **Field** controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for Field, and clicking **Append**. This will add the placeholder for the variable in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the URL.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Redirect URL, click **Update** to save it.
7. Click **Close** to close the window without saving the Redirect URL.

See Choice - Redirect URL for a detailed description of the Redirect URL property window.

Custom Document View URL - This property is used to get the URL of the page to which the custom document dlls does a post.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Custom Document View URL property

1. To specify the Custom Document View URL, click the button for the 'Custom Document View URL' property to open the property window.
2. In this window, enter the Custom Document View URL directly in the text editor.
3. If you want to customize the Custom Document View URL you can do so using the **Field** controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for **Field**, and clicking **Append**. This will add the placeholder for the variable in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the URL.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Custom Document View URL, click **Update** to save it.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. Click **Close** to close the window without saving the Custom Document View URL.

See Choice - Custom Document View URLfor a detailed description of the Custom Document View URL property window.

Mail Channel

The properties in the Mail Channel group have been described below. You can use these properties to customize the Choice activity(activities) with different mail channel templates.

Use html template from property - If this property is set to **'No'**, it will take the mail channel template from the physical path

'[AVEVA Work Tasks Installed Path]\BPM.NET\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\Choice.html.

If this property is set to '**Yes**' and the "**HTML Template**" property is empty, it will take the mail channel template from the physical path

'[AVEVA Work Tasks Installed Path]\BPM.NET\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\Choice.html.

Property Type: Optional (This property needs to be set only if necessary.)

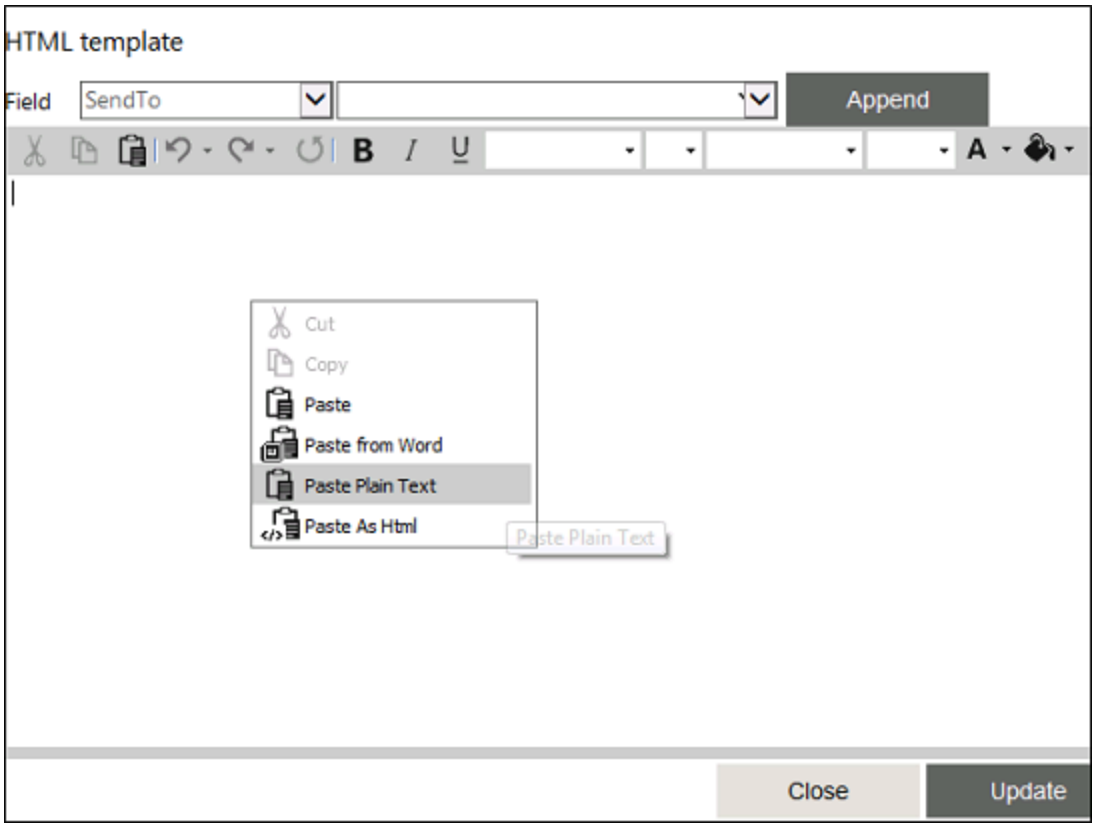
HTML template - This property is used to define the html template for mail channel.

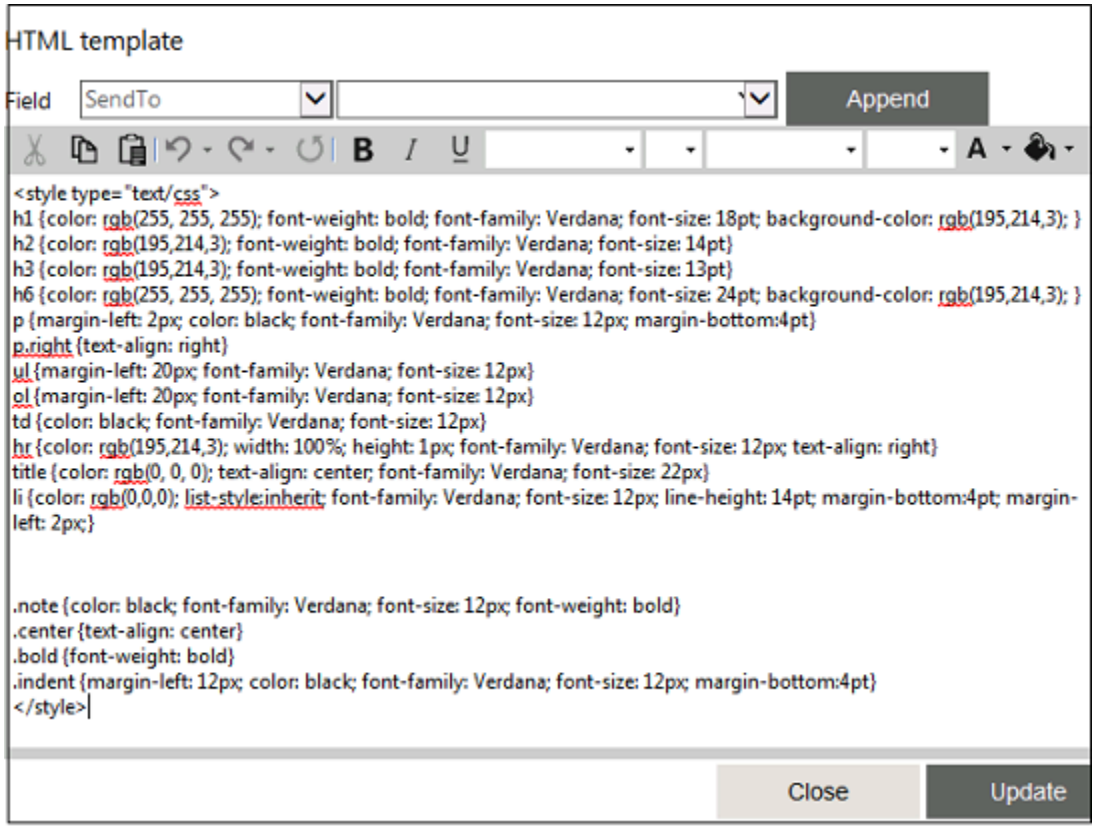
Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the HTML template

1. To specify the HTML template, click the edit icon to open the property window.
2. In the **HTML template** window, provide the template which should be in html format. If you are pasting a copied html content, right-click the **HTML template** window and paste it as "**Paste Plain Text**".

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.





3. Click **Update**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Attachment

This property is used to configure Report to be sent as an Email attachment.

Property Type: Optional (This property needs to be set only if necessary.)

See Choice - Attach Reports for a detailed description of this property.

Decision Influencers

The properties in the Decision Influencers group have been described below. You can use these properties to specify some parameters that influence the outcome of the activity.

Max Actors Required For Choice Completion [% or Number] - This property is used to check what percentage or number of actors should take action for Choice. Decision on the Choice is taken only after the specified number or percentage of actor(s) have taken the action. The activity waits for the actors' action until the 'Timeout for action' is reached.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Min Actors Required For Choice Completion [% or Number] - This property specifies the minimum number of actors to act on the Choice before the 'Timeout Warning for Activity'. If the action limit is reached before the 'Timeout Warning for Activity' then the warning message is not shown.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Escalation & Timeout Top

The properties in the Escalation & Timeout group have been described below. You can use these properties to specify escalation and timeout settings for the activity.

Set Calendar for Timeout - This property is used to specify the calendar to be used to determine activity timeouts.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to Set Calendar for Timeout

1. To specify the calendar for timeout, click the button for the 'Set Calendar for Timeout' property to open the property window.
2. In this window select the required calendar by clicking the calendar name in the tree structure.
3. Click Set Calendar to set the selected calendar.
4. You can specify that a resource calendar should be used, if available, by clicking to check the corresponding check box.

See Choice - Set Calendar for Timeout for a detailed description of the Set Calendar for Timeout property window.

Minimum available time for Activity - This property is used to check the availability of an actor to whom a task must be assigned while executing the activity. The availability check for the actor is done based on the Calendar settings (Global or Resource Calendar). If the time taken to complete the task is within the value set for the **Minimum available time for Activity** property, then the task will be assigned to that actor. However, if the time taken to complete the task exceeds the value set for the **Minimum available time for Activity** property, then the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

task will be assigned based on the values set for the **Ignore Unavailable Actors?** and **Alternate Actor Allowed?** properties.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout warning for Ownership - This property specifies the time at which a warning is provided to an actor to take ownership of the work item assigned to him/her. If the actor does not take ownership within this time the engine is notified with a warning. Thus if 'Min.Ownerships' is set, and it is not satisfied before the given time then a warning is sent. If Min.Ownerships is 0, it will check for Max.Ownerships. If the ownership count does not reach the Max.Ownerships value within the set time, a warning is sent. If both Min.Ownerships and Max.Ownerships are not set, this property is ignored.

Property Type: Optional (This property needs to be set only if necessary.)

- Choice activity is assigned to the actors in the Choice assignment list by the workflow engine. Choice activity has a default property set that at least one actor has to Take Ownership of the Choice activity.
- A drop down interface has been provided for the actor to Take Ownership. However, the actor can act on the Choice without taking ownership first, in which case ownership will be considered to be done. i.e. if the actor acts upon the Choice action without taking ownership, then it is considered as an ownership with action taken.
- After the actor has taken ownership, a link is provided to the actor, which opens a pop up with the details of the other actors. This is again tagged with the 'Hide Responses?' property. The link will be provided only If 'Hide Responses?' is set as 'No'. The pop up will have details like Actor name, Ownership Status, Action taken, Comments.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] then the 'Timeout Warning For Ownership' property is not considered.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] and the 'Time Out For Ownership' is also reached then the work item is removed from other actors' Work items list and the work item waits for action from those who have taken ownership.
- If the 'Min.Ownerships' limit is not reached for the Choice action, and the 'Timeout Warning For Ownership' has been exceeded then the actor is notified with a warning by including an email or an information activity, with the Timeout Warning for Ownership link to it.
- If 'Timeout For Ownership' is set, if the Choice activity has not reached the minimum limit (Min.Ownerships) and the 'Time Out For Ownership' has been reached, then the work item is removed from the Work items lists of all the actors.
- If 'Min.Ownerships' = 0 and if 'Max.Ownerships' is set, the activity waits for actors to take ownership or action until the 'Time Out For Ownership' is reached (if this is set).
- If none of the Ownership properties are set then the work item waits until the actors take ownership or act upon it.

Timeout for Ownership - This property specifies the time until which actors are allowed to take ownership, i.e. the time until which the Choice action will wait for actors to take ownership. If Min.Ownerships is set and is not reached, the work item is removed from all Work Item lists. If Min.Ownerships is 0, it will check if ownership count is less than Max.Ownerships. If it is, the work item is removed from all Work Item lists. Else the work item will be removed from the Work Item list of those who have not taken ownership. The work item will remain in the Work Item list of actors who have taken ownership and wait for action from them.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout warning for Activity - This property specifies when a reminder has to be provided for the actors to act on the Choice request, if the action count has not been reached. If properties such as Max.Ownerships,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Affirmative Action for Choice (AAA), and Action Limit (AL) are set, once the property values are reached then a warning is sent to the actor. When the timeout is reached, the following checks are made: if AL is set, when the action count i.e. the no. of Choices or rejections (AC) < AL, then a warning is sent, if AL = 0 and AAA is set, and when AC < AAA, then a warning is sent to the actor.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Timeout for Activity - This property's value is used to decide until what time the activity should wait for actors' action. If properties such as Max.Ownerships, AAA, AL are set, once the property values are reached the work item is removed from the Work Item lists of all the actors. When the timeout is reached, the following checks are made: If AL is set, when the action count i.e. the no. of Choices or rejections (AC) < AL, then the timeout for action is performed and the activity is removed from all Work Item lists. If AC > AL, it checks with the no. of Choices or 'decision count' (DC) and AAA. If DC >= AAA (if it is set), the activity is considered approved; otherwise, it is rejected. If AAA = 0 and if DC >= AL, the activity is considered approved; otherwise, it is rejected. If AL = 0 and AC < AAA, then the timeout for action is performed and the work item is removed from all Work Item lists. If AL = 0 and AC > AAA, it will check with DC. If DC > AAA it is approved; otherwise, it is rejected. If AL = 0 and AAA = 0, then if DC is at least 1, then it is considered approved; otherwise, it is rejected.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Multiple Timeout Warning - This is used to set recurring timeout warnings that will be sent to the actors who have to take action. To set multiple timeout warnings, click the 'Multiple Timeout Warning' button in the Properties pane for the Choice action. A new window pops up. In this window, set the interval for the recurring warning as well as the output message. You can also set the Recurring Time Interval. This can be used to set a different recurring time out after the first timeout happens. For example, if 'Time Interval' for the property is set as 7 days and 'Recurring Time Interval' is set as 2 days, then the first time out will happen after 7 days and there will be a recurring time out every second day after the seventh day (first) time out.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

1. To set the multiple timeout warning, click the button for the 'Multiple TimeOut Warning' property to open the property window.
2. Enter the time interval for the timeout in the **Time Interval** field.
3. Select "Yes" from the drop down for the **Recurring?** field if you want the timeout warning to recur. Select "No" otherwise.
4. Enter the output message corresponding to the timeout warning in the **Output** field.
5. Enter the time interval for the recurring timeout in the **Recurring Time Interval** field.
6. After entering the content details, click **Add Row** to add this timeout setting and display a new row for entering another timeout setting.
7. Remove rows if necessary by selecting them and clicking **Remove Selected**.
8. After specifying the timeout settings, click **Update**.

See Choice - Multiple Timeout Warning for a detailed description of the Multiple TimeOut Warning property window.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Miscellaneous

Redirection Enabled? - This property enables redirection. When this property is set to Yes, the workflow redirects the user to the URL in the Redirect URL property in the next activity.

Action outputs:

The Choice Action returns the following values -

Action Completed: Displayed when all the actors have taken part.

Action Completed - Affirmative Percentage Not Reached: Displayed when all the actors have not taken part.

Not Enough Resources to Take Acknowledge: Displayed when the filter condition did not yield enough actors to whom the request can be sent.

Timeout - Acknowledge: Displayed when the specified minimum number of actors did not take ownership of the request in the given time.

Timeout - Action: Displayed when some or all of the specified actors did not perform the required action in the given time.

Timeout Warning - Acknowledge: Displayed when a warning is generated that the specified minimum actors have not take ownership of the request in the given time.

Timeout Warning - Action: Displayed when a warning for timeout is generated because some or all of the actors did not perform the required action in the given time.

Note: The following outputs will return only if user assigned tasks to a queue user.

Resource Unavailable for Queue - If a queue is assigned to a task & if there are no resources available in the queue then this output will get triggered.

TimeOut for Queue Task - This output gets triggered when the task in the queue is not picked by any users within the queue escalation time resulting the completion of that task.

Timeout Warning-Queue - This output gets triggered when the task in the queue is not picked by any users within the queue escalation time resulting in sending a warning out put. However, the task will still be sleeping.

Timeout Warning-Queue Participant - This output gets triggered when the task in the queue is picked by user but still not acted on that task within the queue escalation time out settings resulting in triggering a warning output. However, the task will still be sleeping.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Choice - Delivery Channels

Purpose

The purpose of this chapter is to describe the feature of WorkflowStatus Form of Delivery Channels property

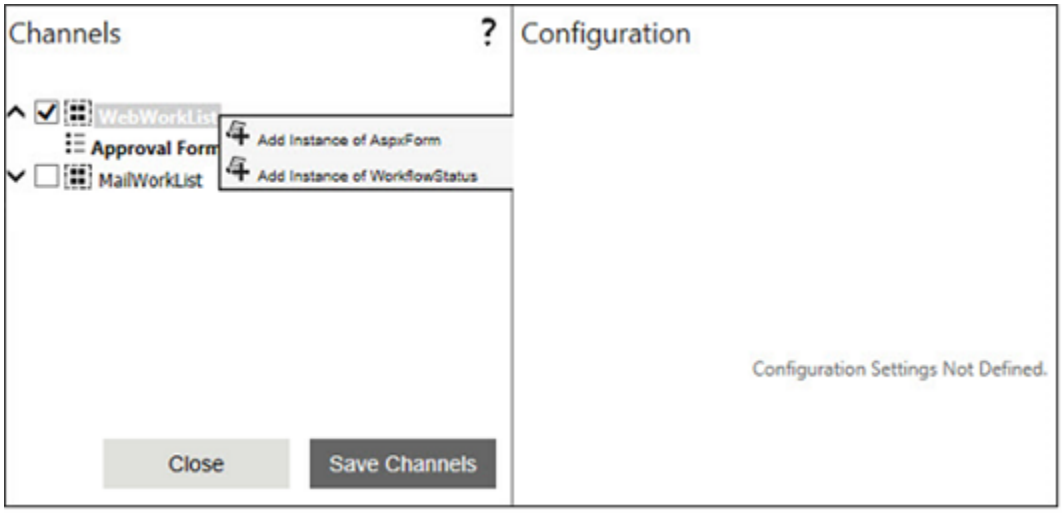
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

under the category Configuration of any human activity available in AVEVA Work Tasks.

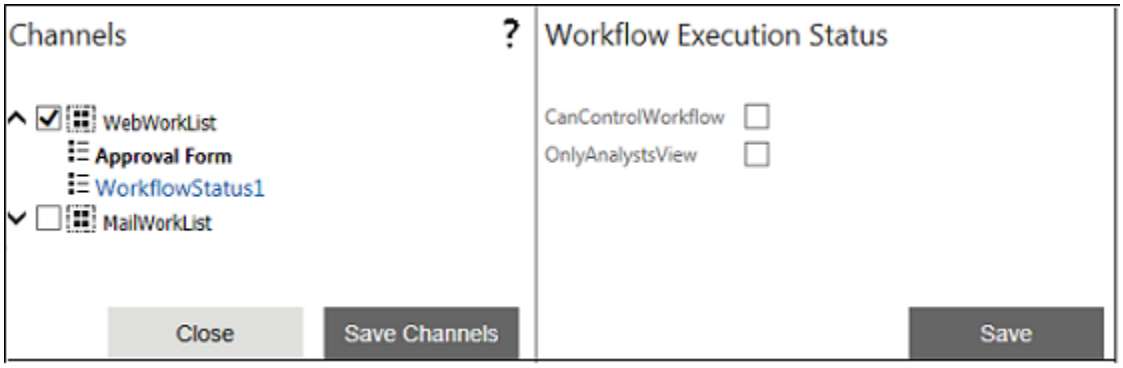
Description: This feature enables us to view the status of the workflow in the work item list in the graphical form. [Optional]

1. Select the Human Activity.
2. Click the **Delivery Channels** Property.

   This will open a window like the one shown below.

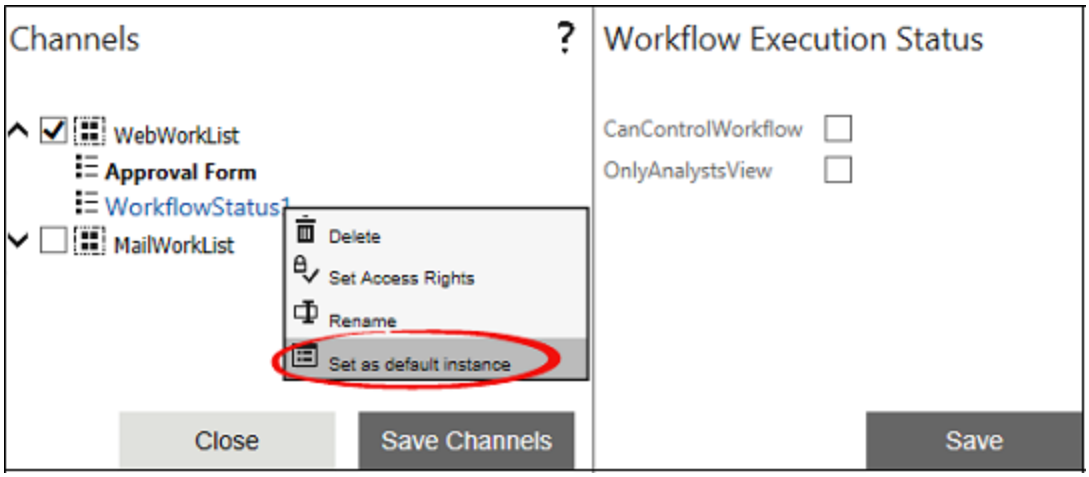


3. Right-click **WebWorkList** to display the context menu with the **Add Instance of Workflowstatus** option as shown below.



4. Set the Properties for the Execution View Status Graphical view.
5. Click **Save**.
6. If you want to set the workflow status form as the default instance, right-click **WorkflowStatus1**, and select **the Set as default instance** option from the context menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



7. Click **Save** and **Save Channels**.

Output

Now, when you click on any row in the Work items list, you will find this Execution view page being displayed as the detailed view if you set it as a default instance.

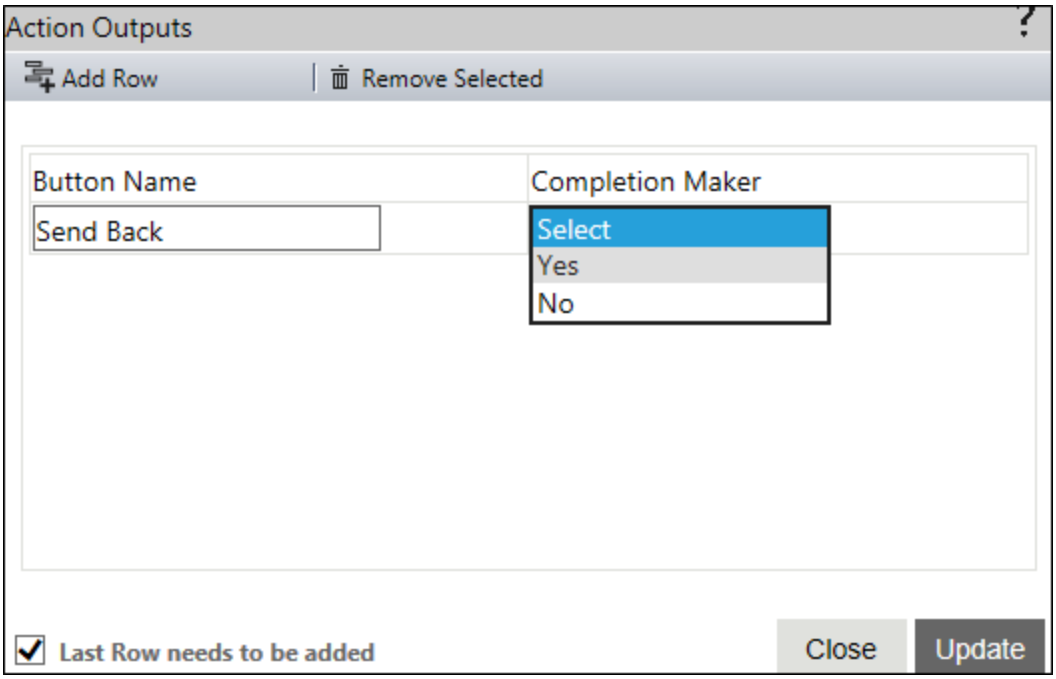
Otherwise, the workflow status form will appear in the context menu of each row in the Work items list and you can open the execution view by selecting it from the context menu.

Choice - Action Outputs

Action Outputs Property Window

You can specify the possible outputs of the Choice activity in this property window. To open this window, click the button for the 'Action Outputs' property. It displays the Outputs in a grid, which also has a blank row where you can enter the parameters for a new action output.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has two tabs at the top to manage the rows of outputs in the grid.

- **Add Row** - You can click this tab to add the newly entered output and display a new row to the grid where you can add another output.
- **Remove Selected** - You can click this tab to remove the output selected in the grid.

The grid has the following parameters that you need to specify for the new output.

- **Button Name** - You can use this field to enter the action output.
- **Completion Maker** - You can use the drop down in this field to specify whether the corresponding output is the completion maker for the activity by selecting "Yes" from the drop down. Select "No" if you do not want the output to be a task completion maker.

There are two buttons and a check box at the bottom of the screen.

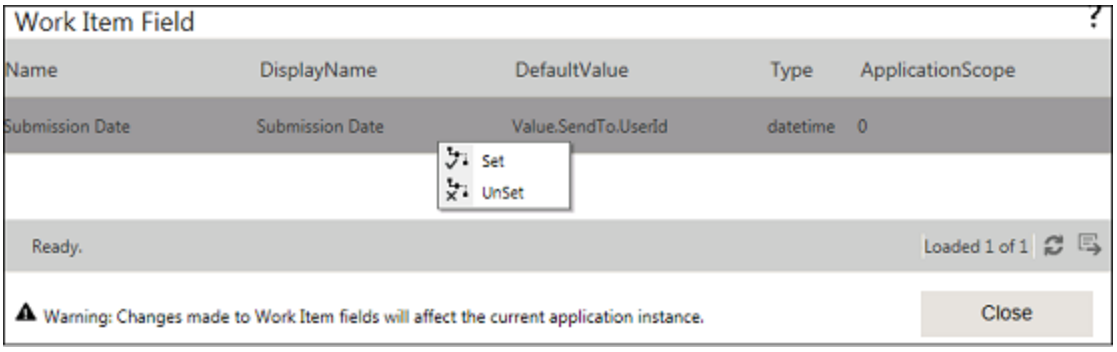
- **Update** - Click this button to save the newly added output(s)
- **Close** - Click this to close the property window without saving the newly added output(s)
- **Last Row needs to be added** - You need to check this check box to add the last row of action outputs that you have entered. If this is not checked, the last row is ignored.

Choice - Work Item Fields

Work Item Fields Property Window

You can set the Work Item Fields for this activity in this property window. To open this window, click the button for the 'Work Item Fields' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a grid listing all the Work Item Fields that have been set for the process.

- Each row in the grid represents a Work Item Field. You can click a Work Item Field to select it.
- The context menu has the Set WorkItemField option, which you can use to set the selected Work Item Field for this activity. It also has the UnSet WorkItemField option to remove the Work Item Field setting.
- **Close** - You can use this button to close the window.

Note: While entering a value in the Work Item field, it is recommended to keep it less than the specified database column size. If there is a need to increase the size of the Work Item field, the user should manually increase the column size in the database.

Choice - Assign Actor(s)

Assign Actor(s) Property Window

You can set the actors to whom the Choice work item is to be sent in this property window. To open this window, click the button for the 'Assign Actor(s)' property.

See Assigning a Task to a Manager for more details.

The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Different Methods of Assigning an Actor

- Assigning an Actor by building an expression:

Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

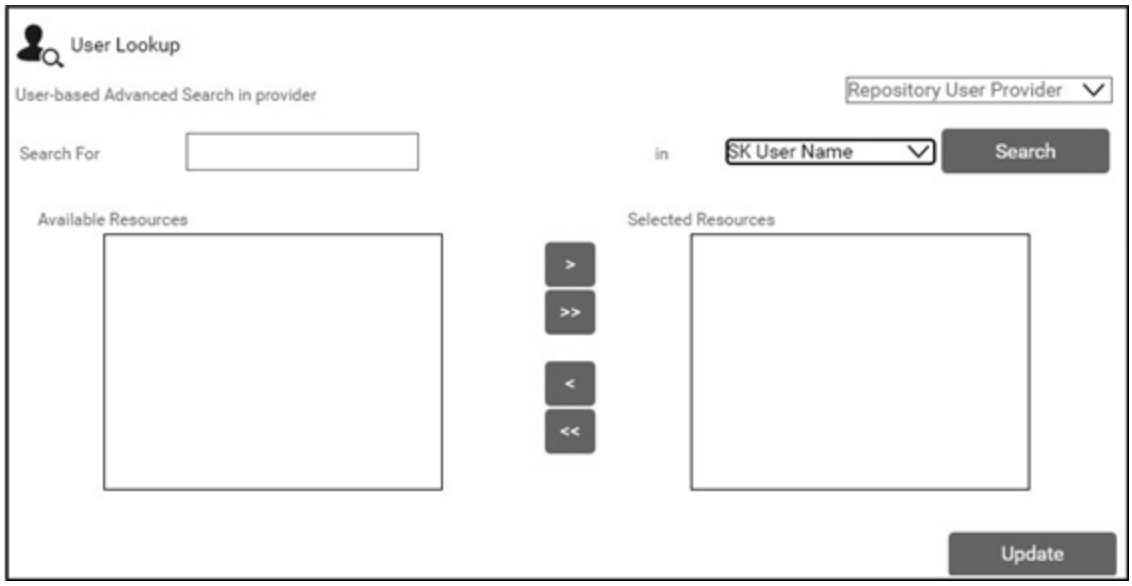
At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.
- **Assigning an Actor through User Lookup**

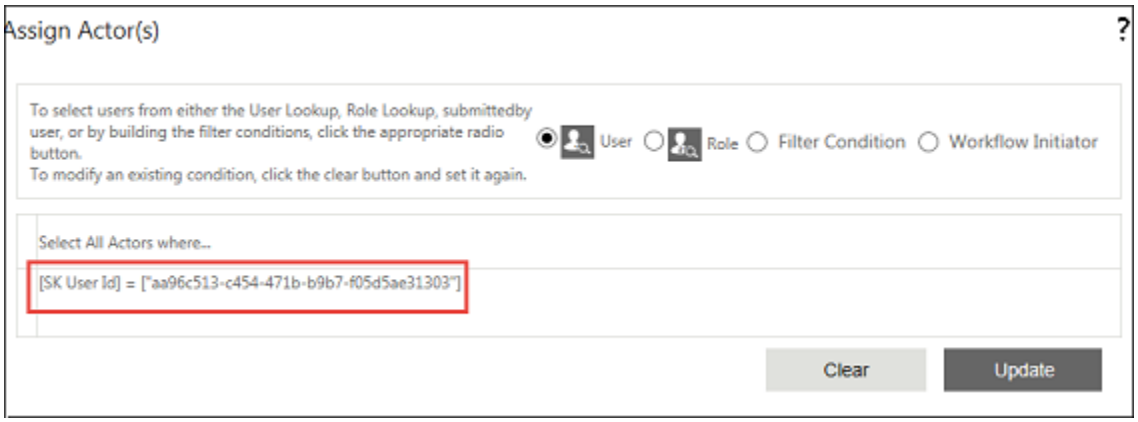
Through the **User Lookup** you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.
2. The **User Lookup** dialog is displayed, search and select the required actor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:



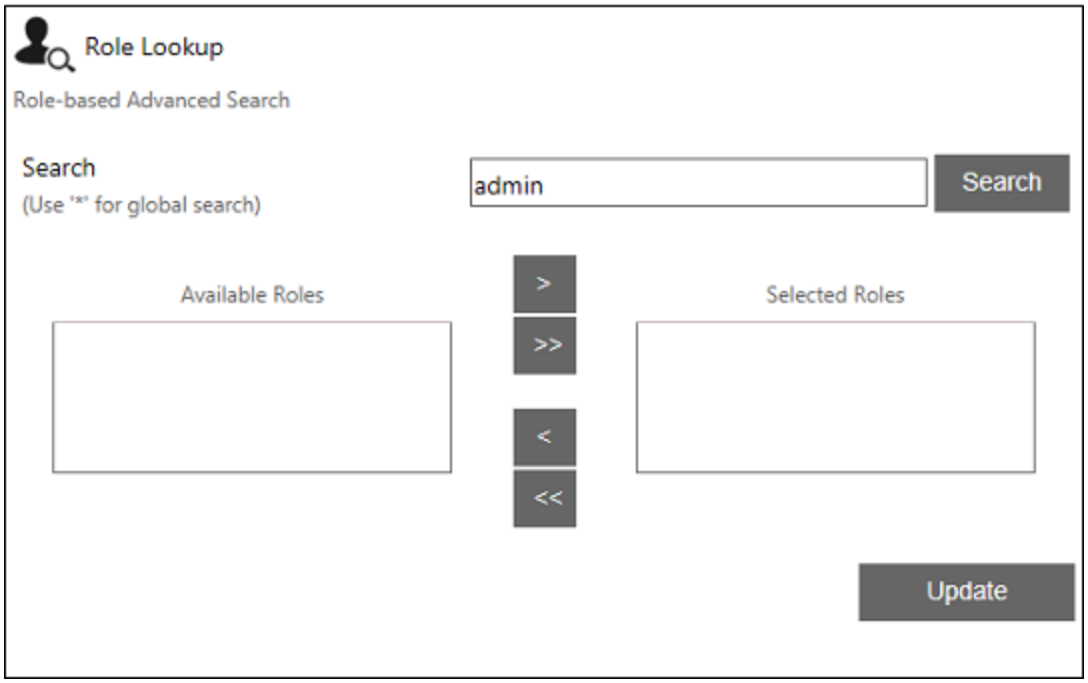
4. **Click Update** in the Assign Actor(s) window to update the selected actor(s) in the **Assign Actor** property.
   Actor(s) selected through User Lookup will be updated in the Assign Actor property.

- Assigning an Actor through Role Lookup

Through the Role Lookup you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

1. Select the **Role Lookup** option.
2. The **Role Lookup** dialog is displayed, search and select for the required role.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



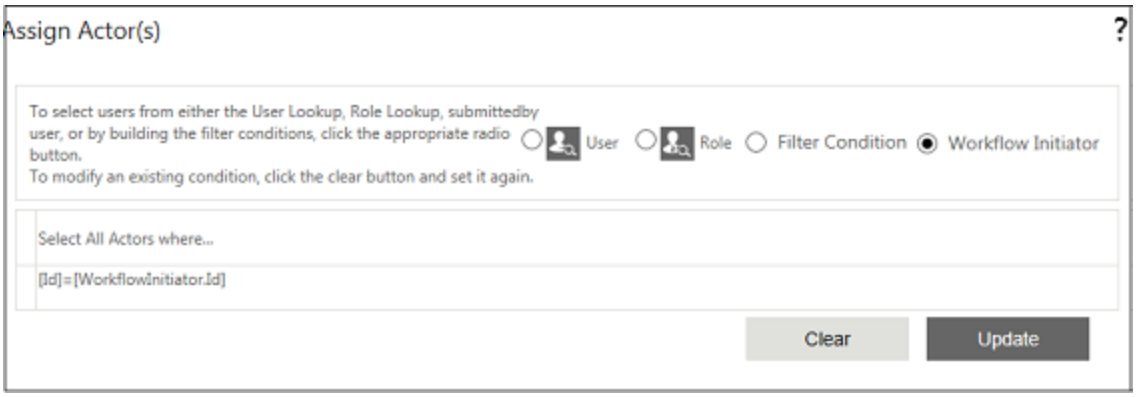
3. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

4. Click **Update** in the Assign Actor(s) window to update the **Assign Actor** property.

- Assigning an Actor through Workflow Initiator

Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

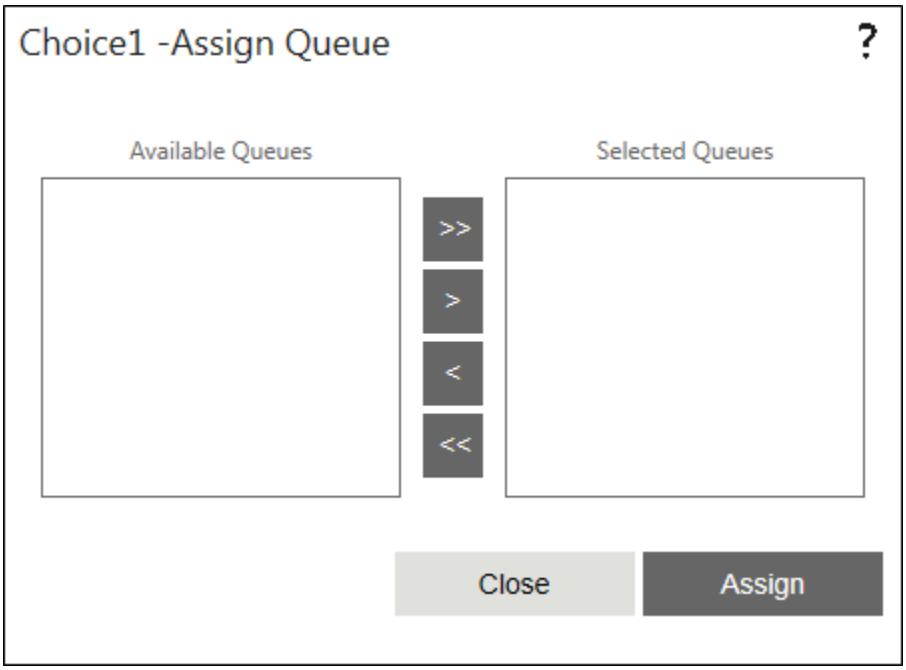
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Choice - Assign Queue(s)

Assign Queue(s) Property Window

You can select the queues to which the Choice work item is to be sent in this property window. To open this window, click the button for the 'Assign Queue(s)' property.



The window has a pair of list boxes to specify the queue(s). The **Available Queues** list shows all the available queues. You can select the required queue(s) from this list box. The **Selected Queues** list shows all the selected queues to which the activity will be assigned.

You can use the following buttons located between the list boxes to move the queues to and from the list boxes.

- [>>] - You can click this button to add all the queues in the Available Queues list to the Selected Queues list.
- [>] - You can click this button to add the selected queue(s) to the Selected Queues list.
- [<] - You can click this button to add the selected queue(s) from the Selected Queues list to the Available Queues list.
- [<<] - You can click this button to add all the queues in the Selected Queues list to the Available Queues list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

At the bottom of the window you will find the following buttons:

- **Assign** - You can click this button to assign the activity to the selected queues.
- **Close** - You can click this button to close this window.

Assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

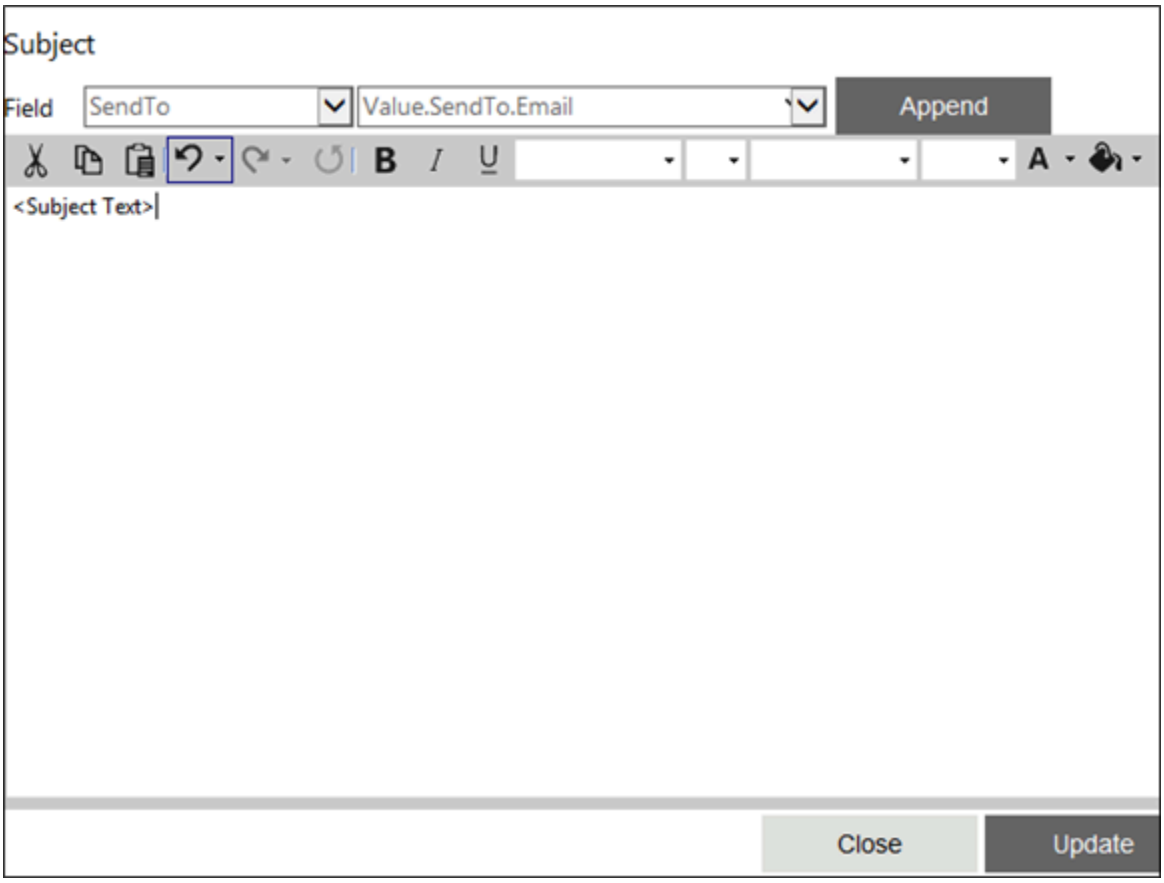
For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the Start activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

To assign multiple queues, specify the queue-ids with semicolon (;) separator. **Skelta.HWS.Queue.QueueCollection** class can be used to retrieve queue information.

Choice - Subject

Subject Property Window

You can specify a subject for the Choice work item assignment in this property window. To open this window, click the button for the Subject' property.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The window has a text editor to enter the subject. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the subject. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the subject in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the subject.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the subject using a combination of discrete value and variable or content. For example, you can specify the subject as "Submitted By <%#Variable.SubmittedBy.Email%>". Here, the string "Submitted By" should be typed directly in the text editor. The second half of the subject is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the Field drop down and clicking Append.

The text tool bar above the text editor allows you to manage and format the subject. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the subject.
- **Close** - You can click this button to close this window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Choice - Body

Body Property Window

You can specify a message Body for the Choice work item assignment in this property window. To access this window, click the button in the 'Body' property.

The window has a text editor to enter the body text. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the body using a combination of discrete value and variable or content. For example, you can specify the body message as "Hi, Please approve the leave application submitted by <%#Variable.SubmittedBy.name%>". Here, the string "Hi, Please approve the leave application submitted by" must be typed directly in the text editor. The second half of the body is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the Field drop down and clicking Append.

The text tool bar above the text editor allows you to manage and format the body text. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

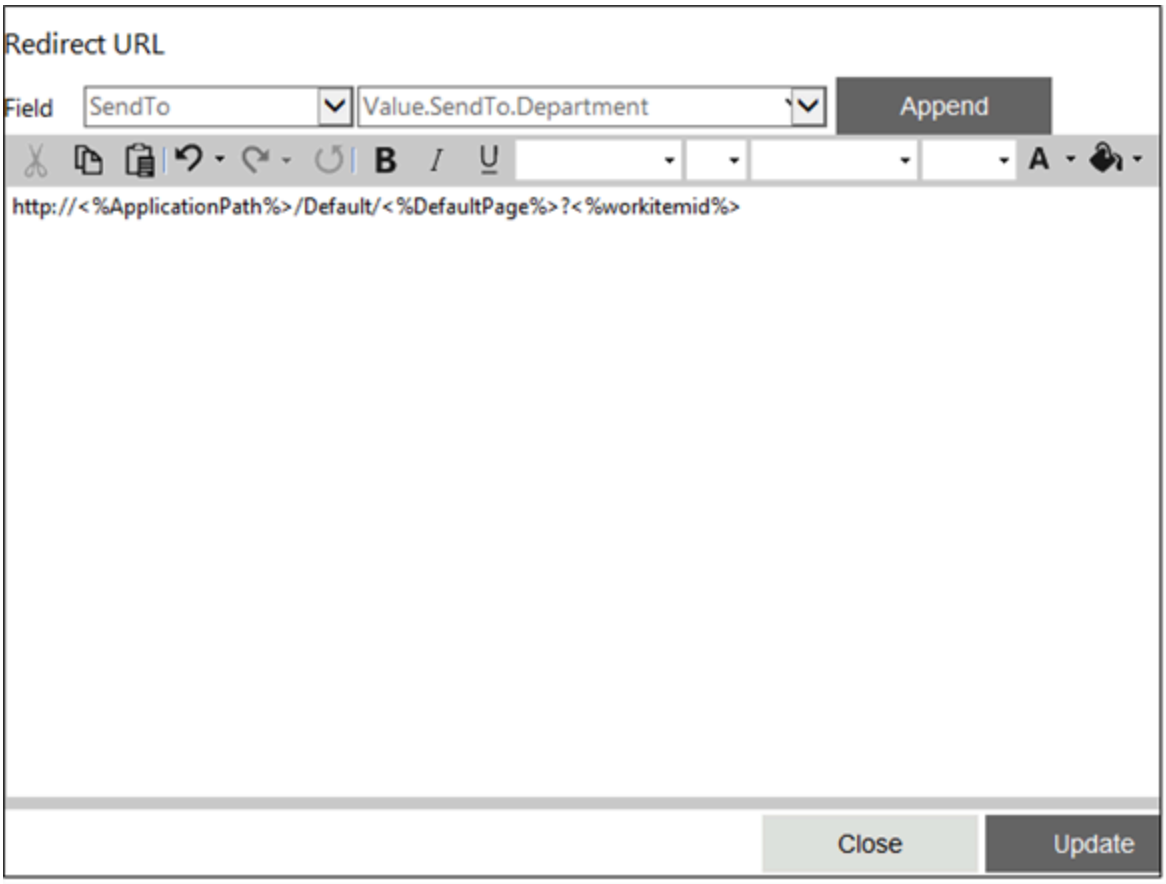
- Update - You can click this button to save the body message.
- Close - You can click this button to close this window.

Choice - Redirect URL

Redirect URL Property Window

You can specify a message Redirect URL for the Choice activity in this property window. To access this window, click the button in the 'Redirect URL' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the Redirect URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Redirect URL in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Redirect URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Redirect URL. It has the following text tools

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to re the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

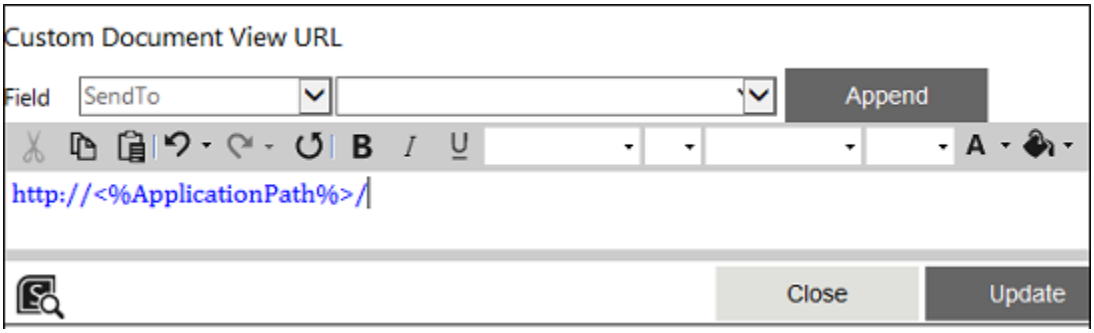
- **Update** - You can click this button to save the Redirect URL.
- **Close** - You can click this button to close this window.

Choice - Custom Document View URL

Custom Document View URL Property Window

You can specify a message Custom Document View URL for the Choice activity in this property window. To access this window, click the button in the 'Custom Document View URL' property.

For more details, see Custom Document View URL.



The window has a text editor to enter the Custom Document View URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop-down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop-down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop-down. If you had selected "Value", it will display a text box here. You can enter the Custom Document View URL in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop-down instead of the field. This drop-down will have the variable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Custom Document View URL.

- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Custom Document View URL. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the Custom Document View URL.
- **Close** - You can click this button to close this window.

Choice - HTML Template

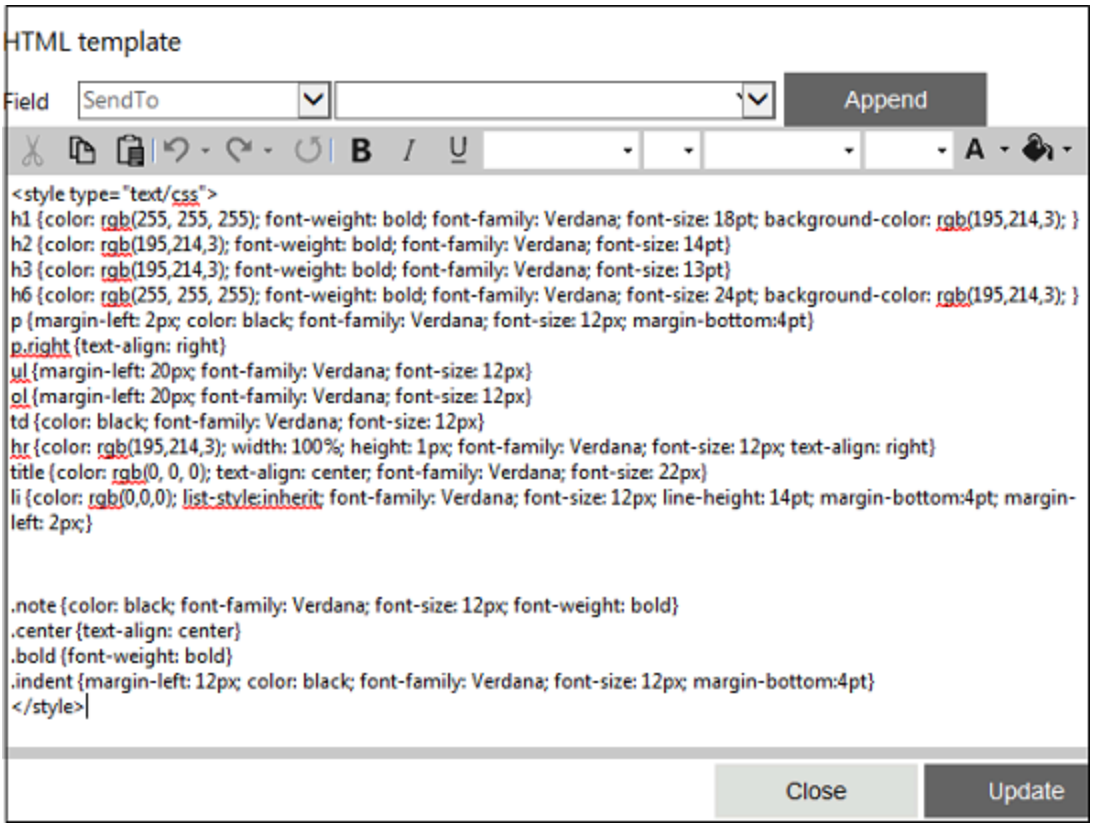
HTML template Property Window

You can specify a html template for the Choice activity in this property window. To access this window, click the edit button in the 'HTML template' property.

In this property window, user can write their own html template for Choice activity or user can make use of default Choice activity template for customizing the email.

See Using 'HTML Template' property of Choice Activity for more information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the body text. You can do this either by entering a html template directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop-down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.
- Append - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the html template. It has the following text tools

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

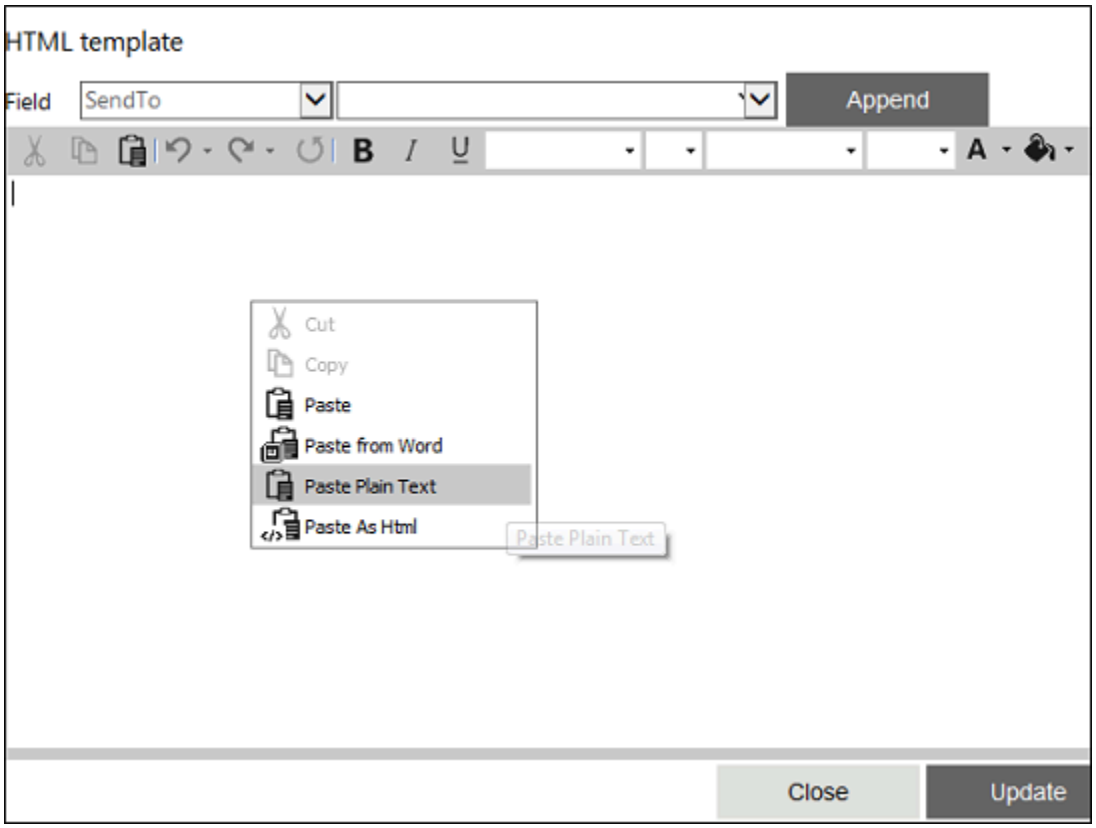
- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

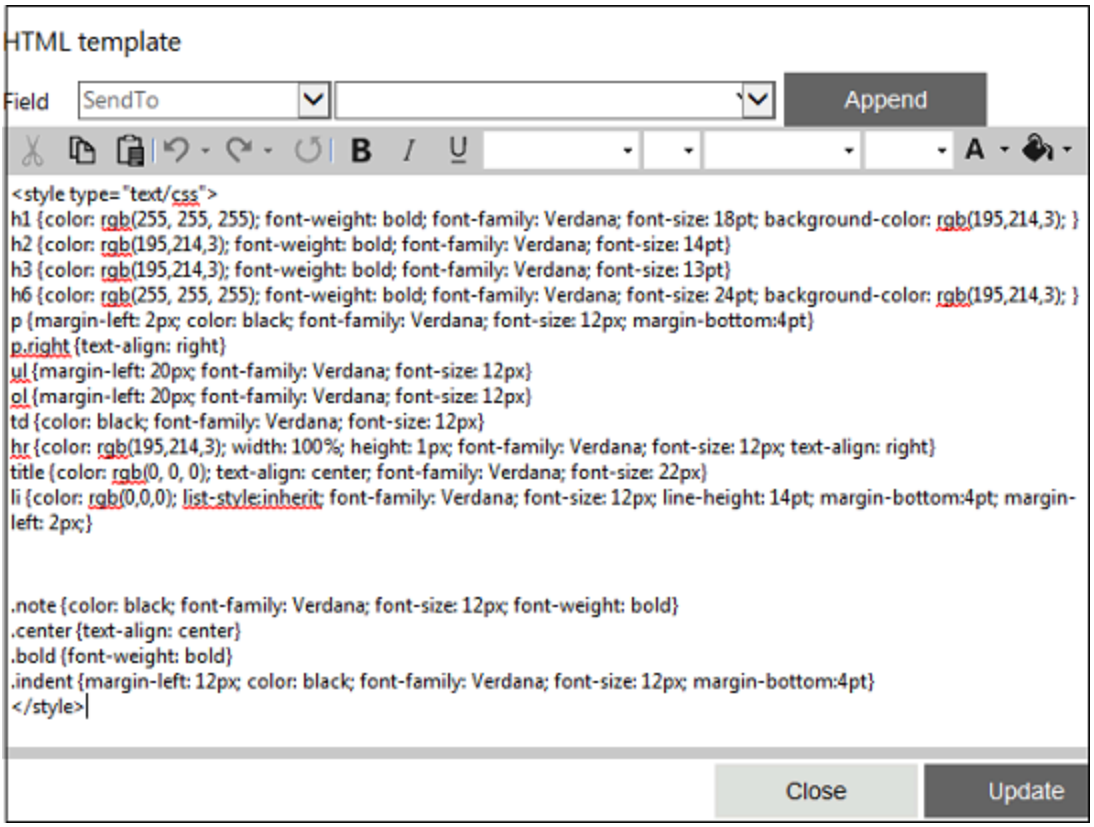
- Update - You can click this button to save the HTML template.
- Close - You can click this button to close this window.

To set HTML temple, follow the steps given below:

1. To specify the HTML template, click the edit icon to open the property window.
2. In the **HTML template** window, provide the template which should be in html format. If you are pasting a copied html content, right-click the **HTML template** window and paste it as **"Paste Plain Text**".



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click **Update**.

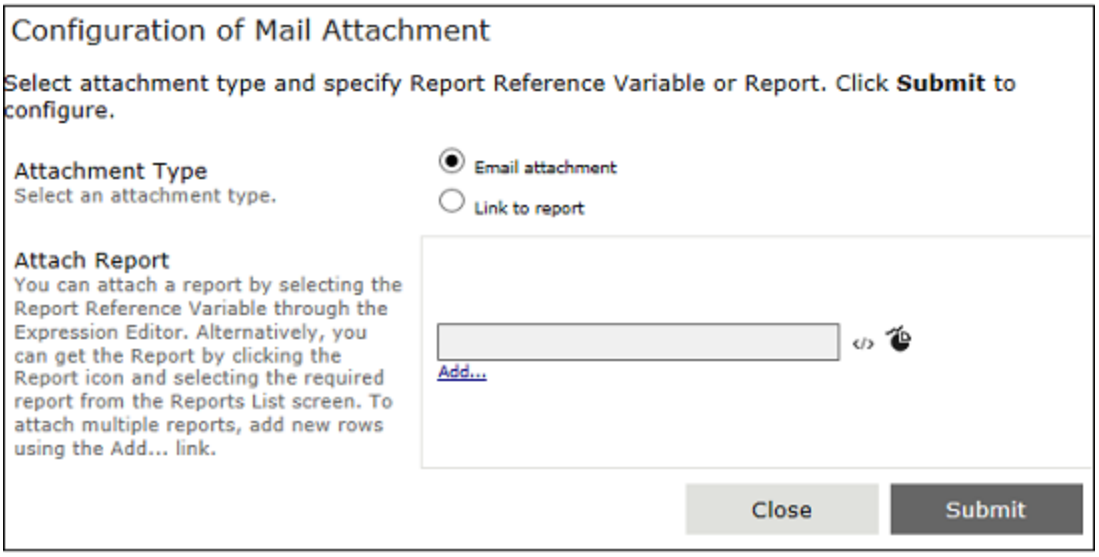
Choice - Attach Reports

Configuration of Mail Attachment property window

You can configure the **Attachment** property for the **Approval** activity in this property window.

In this property window, you can configure the report attachments to be emailed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



- **Attachment Type:** Use this property to select the type of attachment for sending the report as an email.
  - **Email attachment:** Select this type to send the report as an email attachment.
  - **Link to report:** Select this type to embed a link in the email to download the configured report.
- **Attach Report:** Use this property to attach a report generated by the **Report Generator** activity within the workflow or to select a report from the list of generated reports.

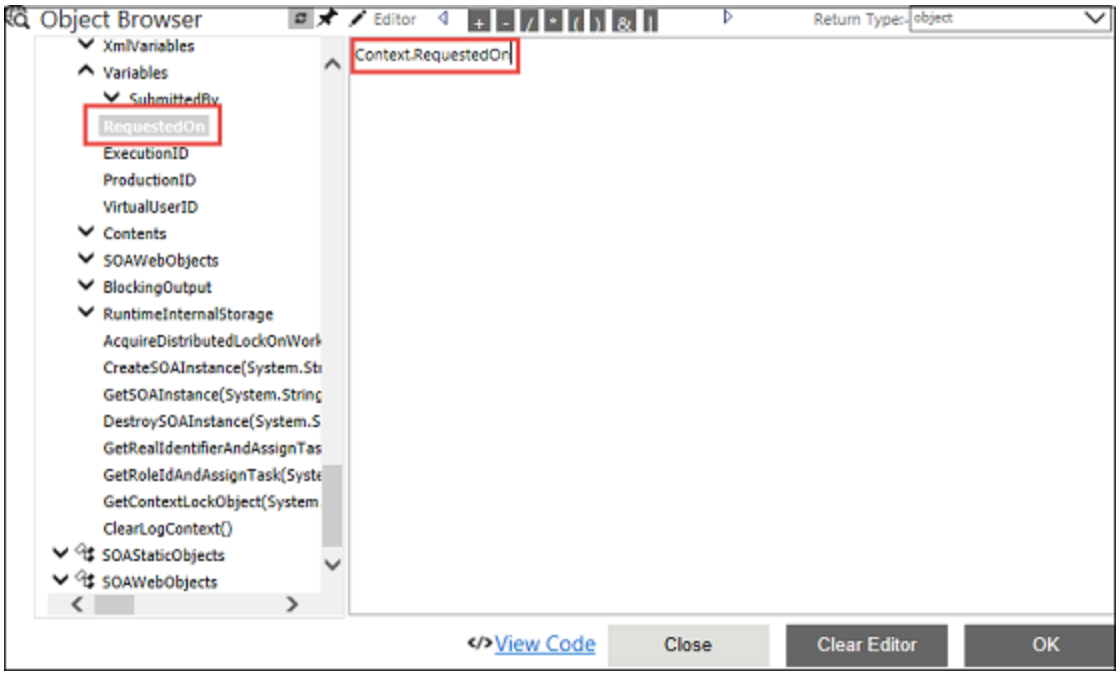
  You can attach a report by selecting the required **Report Reference Variable** using the **Expression Editor**. Alternatively, you can select the report from the Reports List window which consists of the already generated reports.

There must be at least one **Report Generator** activity before an **Approval** activity in the workflow to select the **Report Reference Variable** (using Expression Editor) as an **Attachment**. The **Report Reference Variable** is generated while configuring the **Report Generator** activity. By default, the report reference variable is **[Template Name]_[activity Name**]. For more information about **Report Reference Variable**, see Configure Report Template.

To configure mail attachment

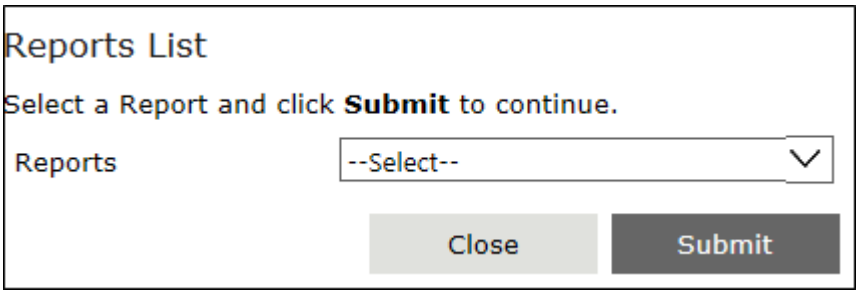
1. Click the edit button for the **Attachment** property.
2. Select the required **Attachment Type**.
3. In the **Attach Report** text box, you can either use the expression editor to get the variable of the report or use the **Reports List** to select the report.
4. To get the variable, click the **Expression Editor** icon given beside the text box to select the required **Report Reference Variable**.
5. In the **Expression Editor** window, select the required variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



6. Click **OK.**
7. Click **Close**.

   The variable appears in the **Attach Report** text box.
8. To select the report from the Reports List dialog box, click the **Reports** icon.



9. In the Reports List dialog box, select the required report, and then click **Submit**.

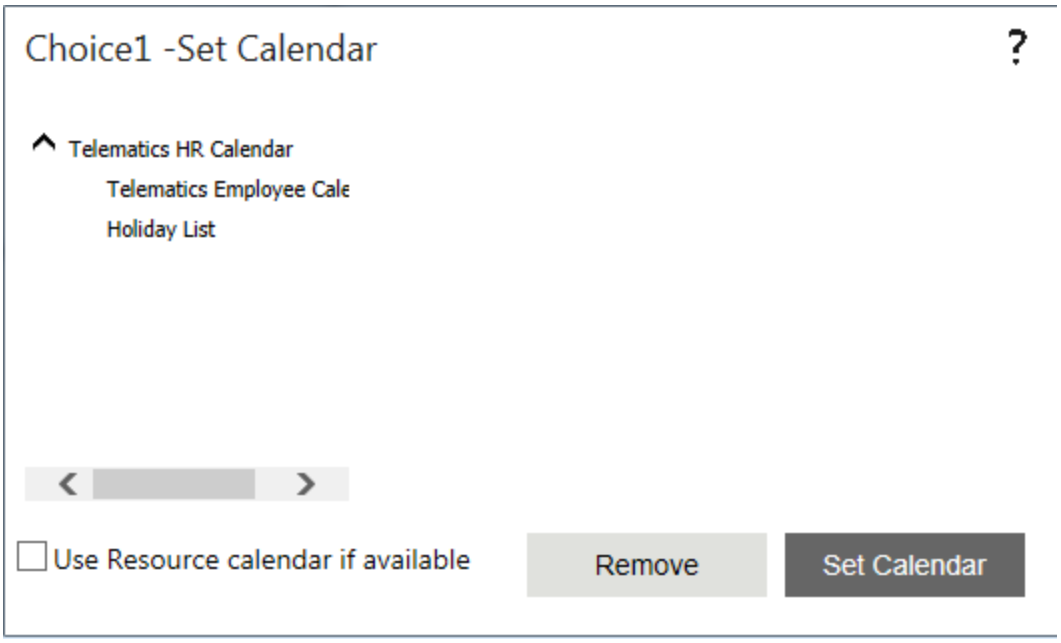
   This window closes automatically.
10. Click **Close** to close the configuration window.

Choice - Set Calendar for Timeout

Set Calendar for Timeout Property Window

You can set a Calendar to calculate the timeout for the Choice activity in this property window. To access this window, click the button in the 'Set Calendar for Timeout' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



This window lists the available calendars in a tree structure, with lower-level calendars nested within the respective top-level calendar. You can click on a calendar to select it for setting.

There are two buttons in this window.

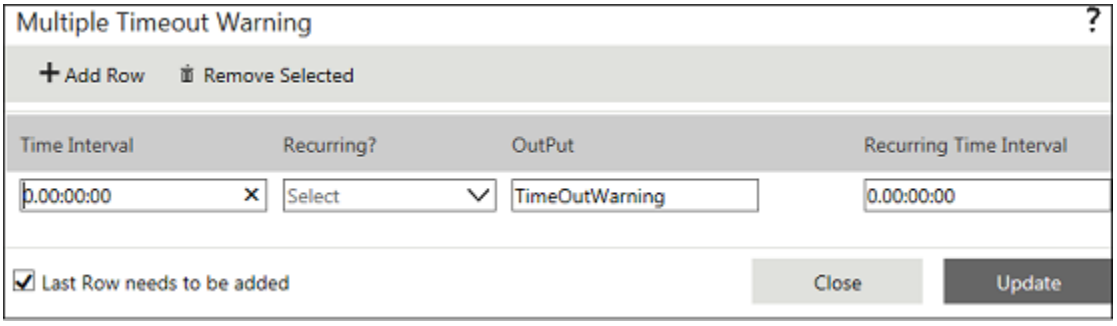
- **No Calendar** - Click this button to indicate that there is no calendar set for the activity.
- **Set Calendar** - Click this button to set the selected calendar.

The **User Resource calendar if available** check box can be selected to enable the use of resource calendars if none are displayed in this window for selection.

Choice - Multiple Timeout Warning

Multiple TimeOut Warning Property Window

You can set the Multiple Timeout Warning in this property window. To access this window, click the button in the 'Multiple TimeOut Warning' property.



The window has two tabs at the top to manage the rows of timeout settings in the grid.

- **Add Row** - You can click this tab to add the newly entered timeout setting and display a new row to the grid where you can add another timeout.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Remove Selected** - You can click this tab to remove the timeout setting row selected in the grid.

The grid has the following parameters that you need to specify for the new timeout setting.

- **Time Interval** - You can use this field to enter the time interval in which you want to generate a timeout warning. You must enter the time in d.hh:mm:ss format.
- **Recurring?** - You can use the drop down in this field to indicate whether the timeout is recurring or not. You can select "Yes" to set the timeout warning to recur.
- **Output** - You can enter the output message corresponding to the timeout warning here
- **Recurring Time Interval** - You can set the time interval in which the timeout warning has to recur here. Enter the time in d.hh:mm:ss format.

There are two buttons and a check box at the bottom of the screen.

- **Update** - You can click this button to save the newly added timeout settings.
- **Close** - You can click this to close the property window without saving the newly added timeout settings.
- **Last Row needs to be added** - You need to select this check box to save the last row added.

Information Activity

Activity Description

The Information Activity is included in the AVEVA Work Tasks workflow process designs when there is a need to provide some information to a set of actors. The actor(s) to whom the information is to be sent can be selected on the basis of various factors like first name, department, designation, email etc. The subject and contents of the information can be customized for each actor.

Application Scenarios

Typical application scenarios could be the Leave Application Workflow Process, where information needs to be sent to the applicant if the leave is approved or rejected, or the Technical Support Request Workflow Process, where information needs to be sent to the appropriate technical support person about the support task being assigned to him/her. The Information activity can be used to send the information in both cases.

Activity Properties

The Information activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Information activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Display Name - This property specifies the name to be displayed for the activity in the Work items list.

Property Type: Optional (This property need not be set if the display name is already configured in the Actions.xml. The name entered here overrides any display name that is configured for this activity in the Actions.xml file.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Delivery Channels - This property is used to specify the delivery channels to deliver this activity to the assigned actor(s). Multiple channels can be specified giving the actor(s) different options to act on the work item.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Delivery Channels property

1. To select the Delivery Channels, click the button for the 'Delivery Channels' property to open the property window.
2. In this window, the Delivery Channels that have been configured are listed on the left along with a check box against each channel. Click and select the check box corresponding to the channel you wish to select. You can select multiple channels by selecting their respective check box.
3. Click **Save Channels**.

See Info - Delivery Channels for a detailed description of the Delivery Channels property window.

Work Item Fields - This property is used to add the Work Item Fields to be used to display application data as well as value in variables and content in the Work items list when this activity is executed.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Work Item Fields property

1. To select the Work Item Fields, click the button for the 'Work Item Fields' property to open the property window.
2. In this window, the Work Item Fields that have been set are listed in a grid. Click a row to select the corresponding Work Item Field.
3. Right-click the row, and select "SetWorkItemField" from the context menu to set it.

See Info - Work Item Fields for a detailed description of the Work Item Fields property window.

Actors

The properties in the Actors group have been described below. You can use these properties to specify the actors who will work on the activity and related properties.

Assign Actor(s) - This property specifies the actor(s) who has to take up the Information activity. It is tagged with

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the Ignore Unavailable Resources property which in turn is tagged with Alternate Resource Allowed. This will be explained in detail later.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any resource(s).)

Steps to set the Assign Actor(s) property

1. To specify the actors, click the button for the 'Assign Actor(s)' property to open the property window.
2. In this window, build an expression to select the actors using the actors properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Info - Assign Actor(s) for a detailed description of the Assign Actor(s) property window.

Assign Queue(s) - This property specifies the queue(s) to which the Information activity is to be assigned.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any Queue(s).)

Steps to set the Assign Queue(s) property

1. To specify the queue(s), click the button for the 'Assign Queue(s)' property to open the property window.
2. In this window, select the queue(s) from the **Available Queues** list box. You can select multiple queues by selecting with the <Ctrl> key pressed.
3. After selecting the required queue(s), click the [>] button to move it to the **Selected Queues** list box. You can click the [>>] button to select all the queue(s) and move them to the **Selected Queues** list box.
4. You can remove any queue from the selected list by selecting it and clicking the [<] button. You can remove all the queues by clicking the [<<] button.
5. After adding the required queue(s), click **Assign** to assign the activity to the selection.
6. Click **Close** to close the window without saving the selection.

See Info - Assign Queue(s) for a detailed description of the Assign Queue(s) property window.

Notification Contents

The properties in the Notification Contents group have been described below. You can use these properties to specify the content used to notify actors about the activity.

Subject - This is the subject of the Information work item. The subject can be customized for each actor.

Property Type: Optional (This property need not be set. If it is not set, the default subject line from the Actions.xml file is displayed.)

Steps to set the Subject property

1. To specify the subject, click the button for the 'Subject' property to open the property window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. In this window, enter the subject text directly in the text editor.
3. If you want to customize the subject you can do so using the **Field** controls to add resource property values. For example, to add a subject that includes the email id of the actor submitting the approval request, you can do as follows: Type in the text "Submitted By". Select "Variable" and "Variable.SubmittedBy.Email" from the two drop downs for **Field** and click **Append**. This will add the placeholder for the variable after the text entered in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the email id.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the subject, click **Update** to save it.
7. Click **Close** to close the window without saving the subject.

See Info - Subject for a detailed description of the Subject property window.

Body - This is the actual message of the Information sent to the actor(s). The body can be customized for each actor. There is an **Enterprise Console Explorer**(plug-in) icon in the Body property window to get the work item details URL. See Enterprise Console Control (EC Plug-in) for more information.

Property Type: Optional (This property need not be set.)

Steps to set the Body property

1. To specify the message body, click the button for the 'Body' property to open the property window.
2. In this window, enter the body text directly in the text editor.
3. If you want to customize the body you can do so using the Field controls to add resource property values. For example, to add a body that includes the name of the actor submitting the approval request, you can do as follows: Type in the text: "Hi, Please cast your vote in the poll conducted by". Select "Variable" and "Variable.SubmittedBy.name" from the two drop downs for **Field** and click **Append**. This will add the placeholder for the variable after the text entered in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the name.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the body, click **Update** to save it.
7. Click **Close** to close the window without saving the body.

See Info - Body for a detailed description of the Body property window.

From Email Address - This property is used to specify the email address from which notification messages will be sent for the activity.

Property Type: Optional (This property needs to be set only if necessary. If it is not set, the default value in Activities.xml will be used.)

Send Notification Email - If this property is set to 'Yes', then notification emails will be sent to the actors. If it is set to 'Yes-HTML', then the emails will be sent in HTML format.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Redirect URL - This property is used to get the URL of the web page to redirect the client to, in the case of synchronous execution of the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Redirect URL property

1. To specify the Redirect URL, click the button for the 'Redirect URL' property to open the property window.
2. In this window, enter the Redirect URL directly in the text editor.
3. If you want to customize the Redirect URL you can do so using the **Field** controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for **Field**, and clicking **Append**. This will add the placeholder for the variable in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the URL.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Redirect URL, click **Update** to save it.
7. Click **Close** to close the window without saving the Redirect URL.

See Info - Redirect URL for a detailed description of the Redirect URL property window.

Custom Document View URL - This property is used to get the URL of the page to which the custom document dlls does a post.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Custom Document View URL property

1. To specify the Custom Document View URL, click the button for the 'Custom Document View URL' property to open the property window.
2. In this window, enter the Custom Document View URL directly in the text editor.
3. If you want to customize the Custom Document View URL you can do so using the Field controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for **Field**, and clicking **Append**. This will add the placeholder for the variable in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the URL.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Custom Document View URL, click **Update** to save it.
7. Click **Close** to close the window without saving the Custom Document View URL.

See Info - Custom Document View URL for a detailed description of the Custom Document View URL property window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Mail Channel

The properties in the Mail Channel group have been described below. You can use these properties to customize the Information activity (activities) with different mail channel templates.

Use html template from property - If this property is set to 'No', it will take the mail channel template from the physical path

'[AVEVA Work Tasks Installed Path]\BPM.NET\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\Information.html.

If this property is set to **'Yes'** and the **"HTML Template"** property is empty, it will take the mail channel template from the physical path

'[AVEVA Work Tasks Installed Path]\BPM.NET\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\Information.html.

Property Type: Optional (This property needs to be set only if necessary.)

HTML template - This property is used to define the html template for mail channel.

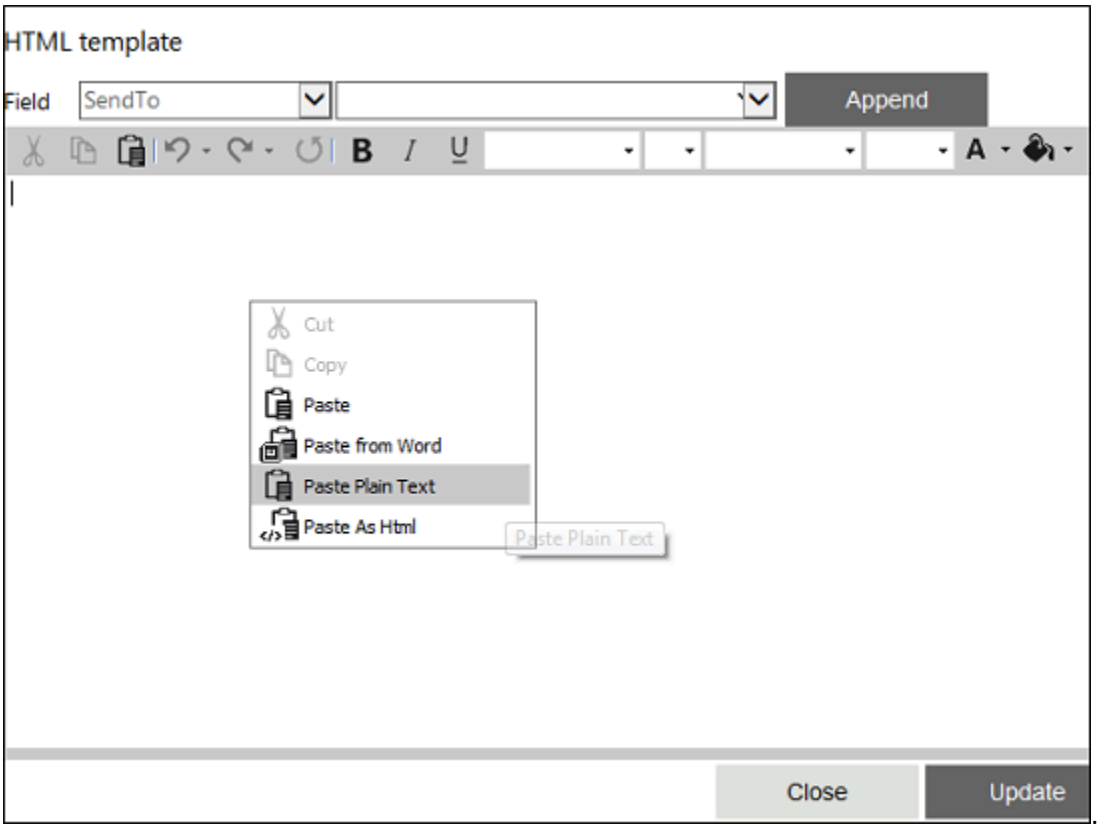
Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the HTML template

1. To specify the HTML template, click the edit icon to open the property window.
2. Click Update.

See Info - HTML Template for a detailed description of the property HTML template window.

Attachment



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Attachment property

1. To specify the **Attachment** property, click the edit icon to open the property window.
2. In the **Configuration of Mail Attachment** window, select the required attachment type by clicking the radio button.
3. In the **Attach Report** property, you can attach a report by selecting the **Report Reference Variable** through the Expression Editor. Alternatively, you can get the **Report** by clicking the **Report** icon and selecting the required report from the Reports list screen. To attach multiple reports, add new rows using the Add link.
4. Click **Submit**.

See Info - Attach Reports for a detailed description of this property.

Activity outputs

The Information Activity does not return any value.

Escalation & Timeout

You can use this property to specify escalation and timeout settings for the activity.

Timeout for Activity - This property is used for setting timeouts for the information activity. If specified, the activity will be timed out after the specified time span and will be moved from the **Inbox**.

The timeout setting must be specified in the below format:

0.00:00:00 which indicates Day.HH:MM:SS

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Info - Delivery Channels

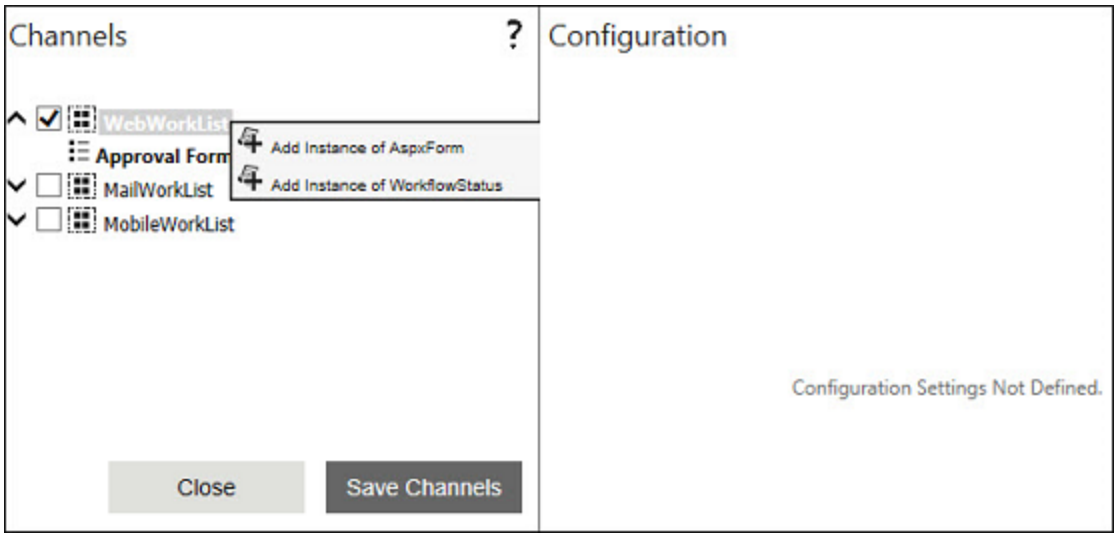
Purpose

The purpose of this chapter is to describe the feature of WorkflowStatus Form of Delivery Channels property under the category Configuration of any human activity available in AVEVA Work Tasks.

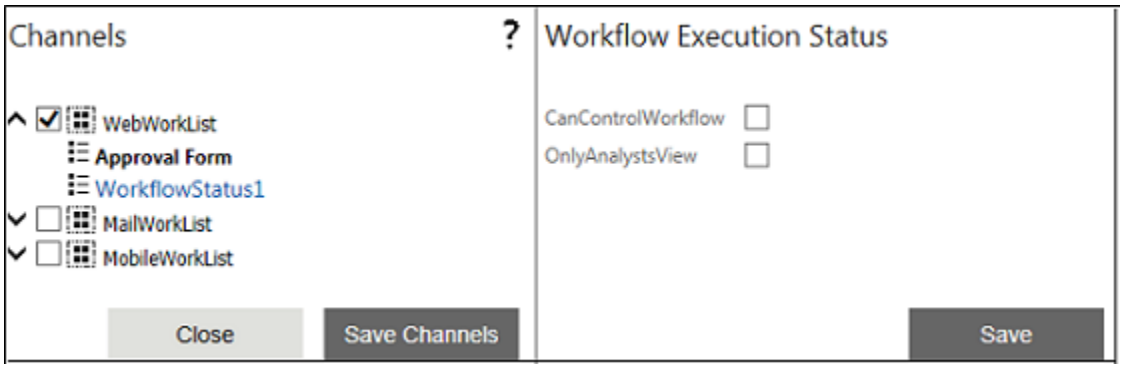
Description: This feature enables us to view the status of the workflow in the work item list in the graphical form. [Optional]

1. Select the Human Activity.
2. Click the Delivery Channels Property.
3. This will open a window like the one shown below.

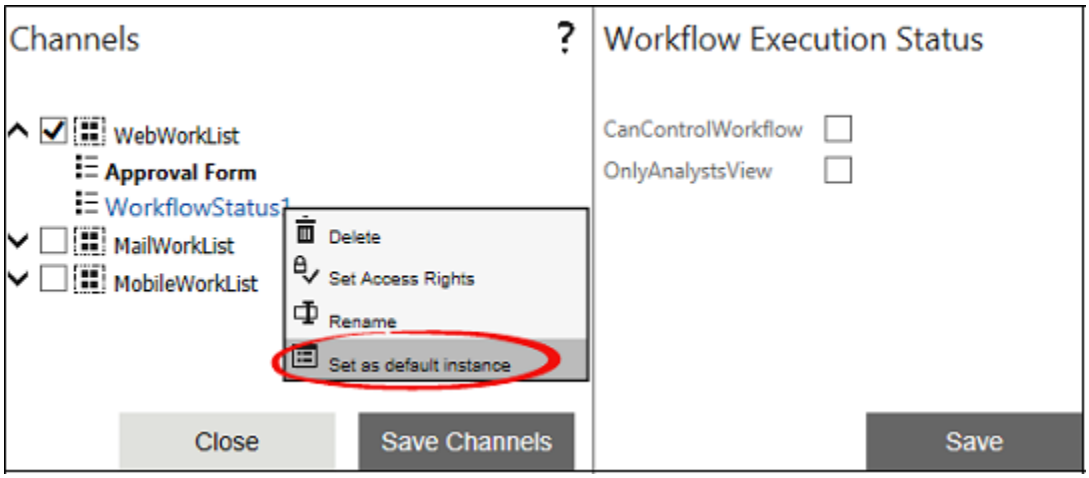
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



4. Right-click the WebWorkList to display the context menu with the **Add Instance of Workflowstatus** option as shown below.



5. Set the Properties for the Execution View Status Graphical view. Click **Save**.
6. If you want to set the workflowstatus form as the default instance, you can do it by right-clicking workflowstatus1 and selecting **the Set as default instance** from the context menu.



7. Click the **Save** and **Save Channels** buttons.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Output

Now, when you click any row in the Work items list, you will find this Execution view page being displayed as the detailed view if you set it as a default instance.

Otherwise, the workflow status form will appear in the context menu of each row in the Work items list and you can open the execution view by selecting it from the context menu.

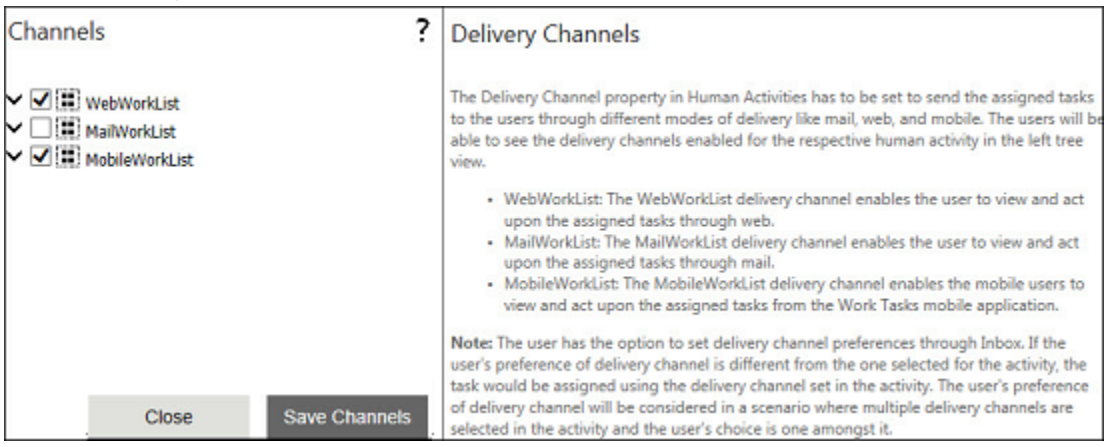
Delivery Channels - MobileWorkList

To deliver the work items to the mobile device, the **MobileWorkList** option in the Delivery Channel property needs to be selected. This allows mobile users to view and act upon the tasks from the Inbox that is rendered on their mobile devices.

This topic describes the steps required to configure the **MobileWorkList** option in the **Delivery Channel** property of the Information activity.

Configuring 'MobileWorkList' option in the Delivery Channels Property

1. In the **Process Designer**, drag and drop the Approval activity on to the design area. The activity properties pane of the selected activity appears on the right side of the design area.
2. In the **Activity Properties** pane, expand the **Configuration** node and select the **Delivery Channels** property.
3. In this window, select the **MobileWorkList** check box in the **Channels** section.



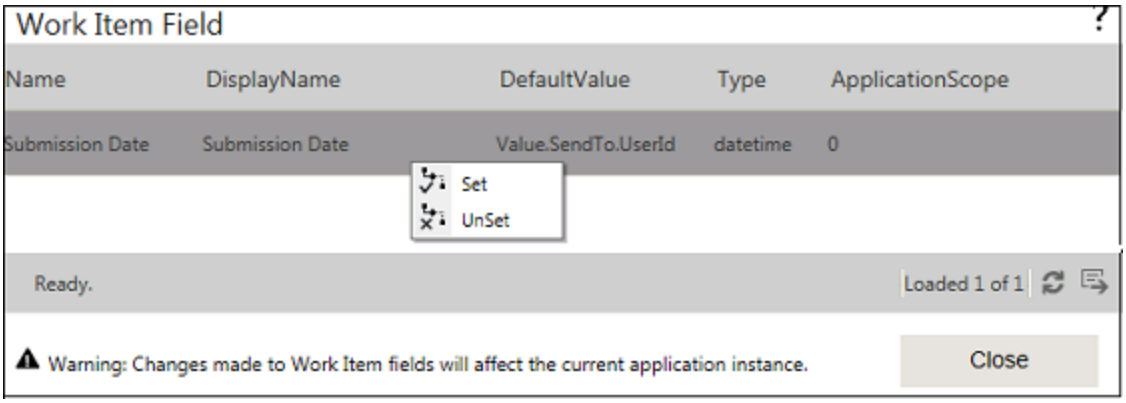
4. Click **Save Channels** and **Close**.
5. Set all the other properties and click **Publish and Test Run**.

Info - Work Item Fields

Work Item Fields Property Window

You can set the Work Item Fields for this activity in this property window. To open this window, click the button for the 'Work Item Fields' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a grid listing all the Work Item Fields that have been set for the process.

- Each row in the grid represents a Work Item Field. You can click a Work Item Field to select it.
- The context menu has the Set WorkItemField option, which you can use to set the selected Work Item Field for this activity. It also has the UnSet WorkItemField option to remove the Work Item Field setting.
- **Close** - You can use this button to close the window.

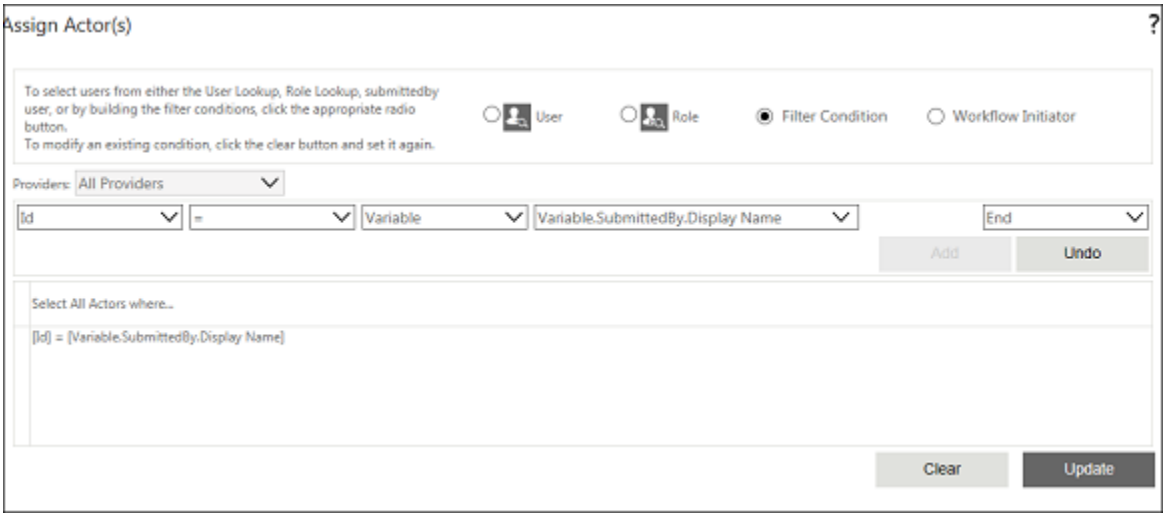
Note: While entering a value in the Work Item field, it is recommended to keep it less than the specified database column size. If there is a need to increase the size of the Work Item field, the user should manually increase the column size in the database.

Info - Assign Actor(s)

Assign Actor(s) Property Window

You can set the actors to whom the Information work item is to be sent in this property window. To open this window, click the button for the 'Assign Actor(s)' property.

See Assigning a Task to a Manager for more details.



The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Different Methods of Assigning an Actor

- Assigning an Actor by building an expression:
  Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.
  - The Providers drop-down- You can select the required provider from the Providers drop-down.
  - The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
  - The Operator drop down - You can select the operator for the expression here. E.g., "="
  - The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
  - The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
  - The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
    - **Add** - You can click this button to add the new expression to the text area.
    - **Undo** - You can click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

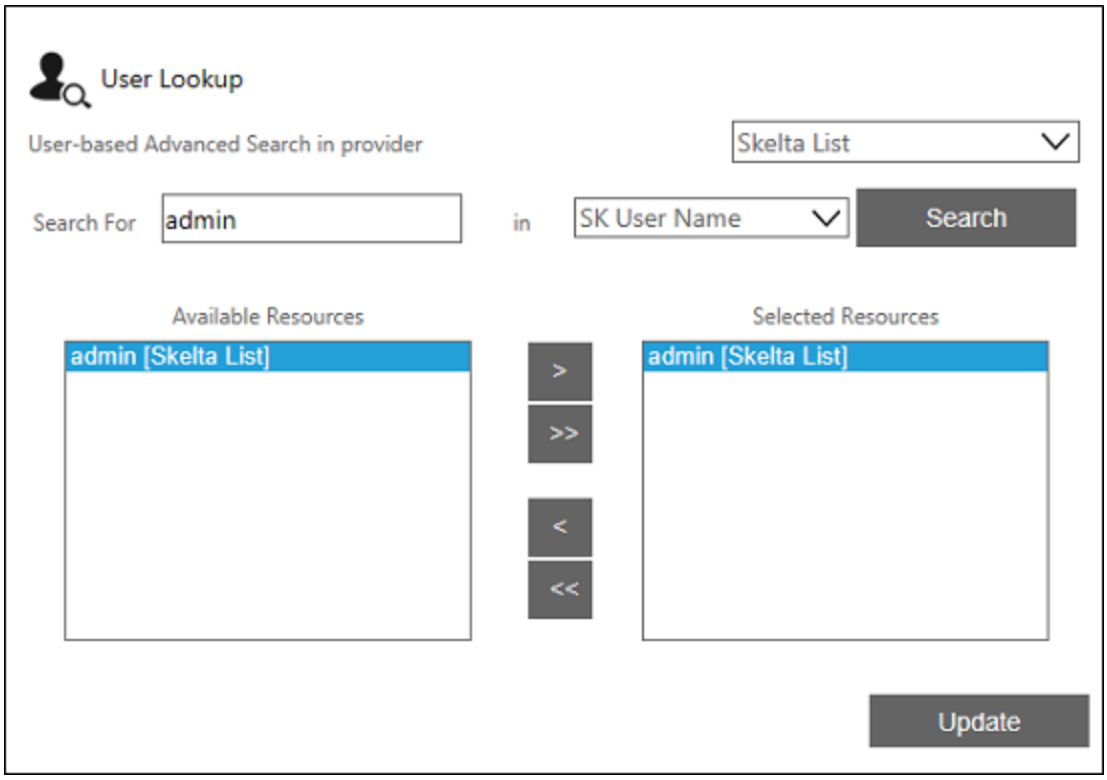
- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

- Assigning an Actor through User Lookup

Through the User Lookup you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.
2. The **User Lookup** dialog is displayed, search and select the required actor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



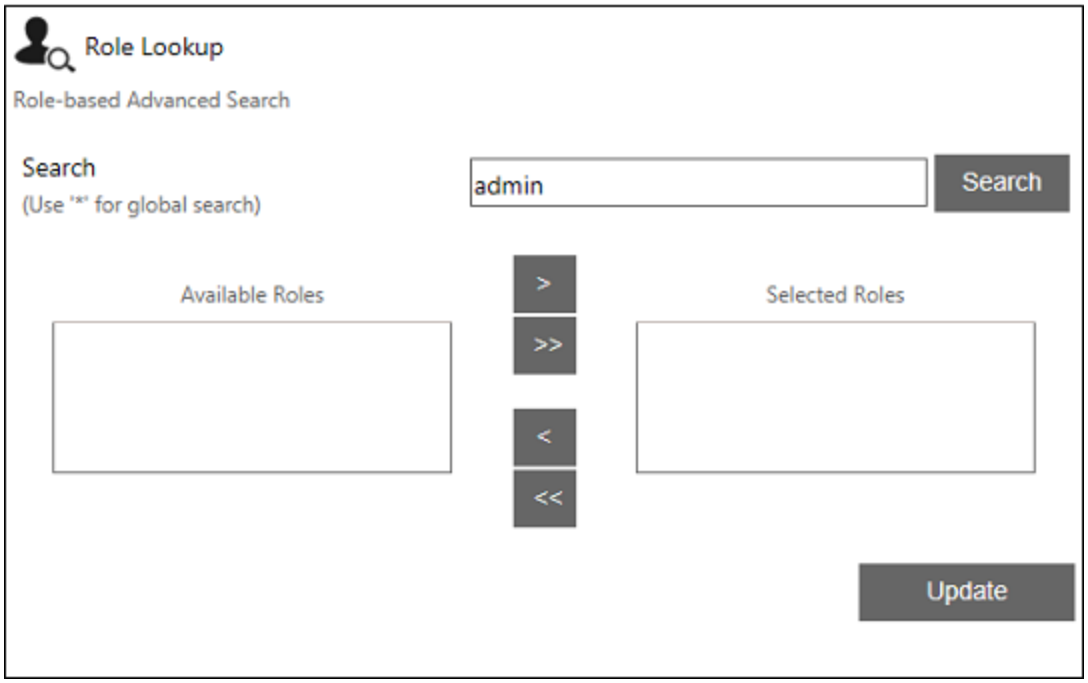
3. Click **Update** in the User Lookup dialog and verify the Assign Actor window.
4. In the Assign Actor window, you will see that an expression is displayed, refer the image below:
5. Click **Update** in the Assign Actor(s) window to update the selected actor(s) in the Assign Actor property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

- Assigning an Actor through Role Lookup

Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

1. Select the **Role Lookup** option.
2. The **Role Lookup** dialog is displayed, search and select for the required role.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



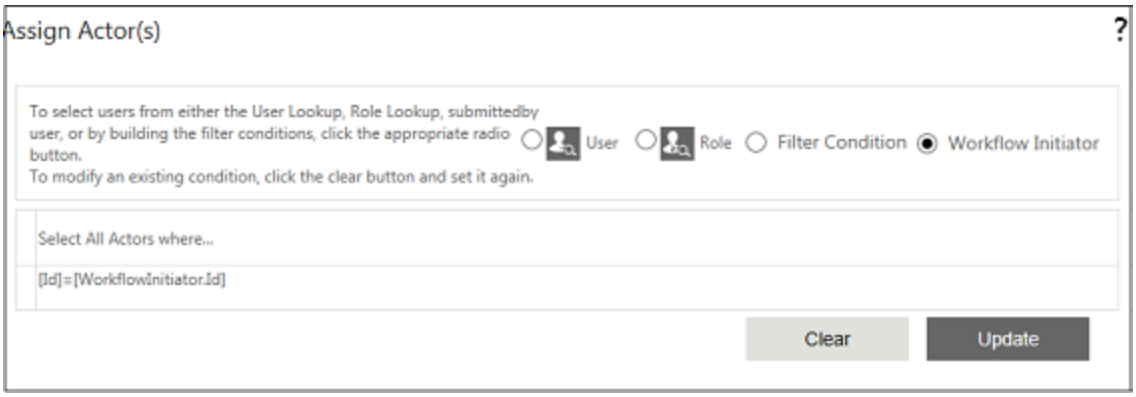
3. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

4. Click **Update** in the Assign Actor(s) window to update the **Assign Actor** property.

- Assigning an Actor through Workflow Initiator

Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

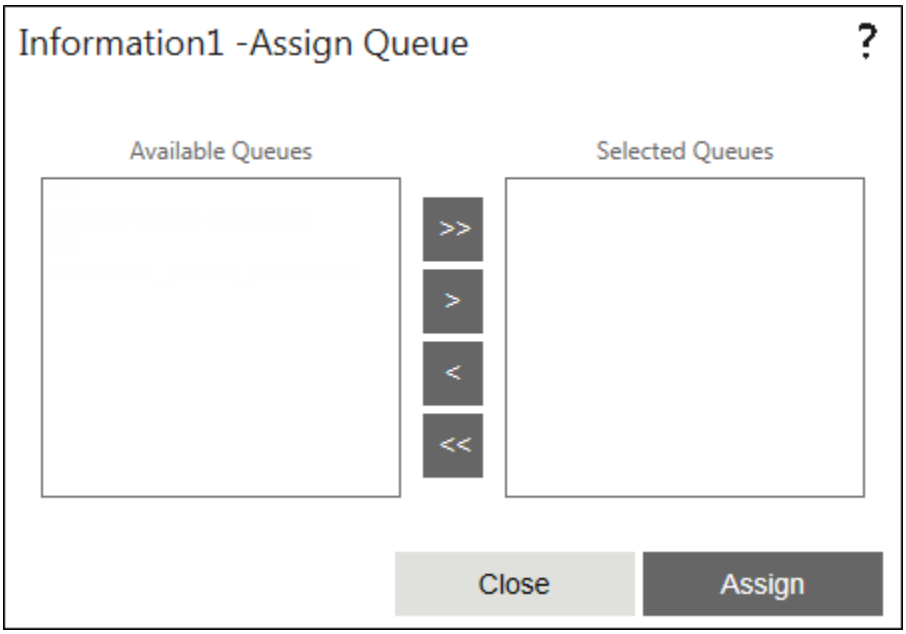
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Info - Assign Queue(s)

Assign Queue(s) Property Window

You can select the queues to which the Information work item is to be sent in this property window. To open this window, click the button for the 'Assign Queue(s)' property.



The window has a pair of list boxes to specify the queue(s). The Available Queues list shows all the available queues. You can select the required queue(s) from this list box. The Selected Queues list shows all the selected queues to which the activity will be assigned.

You can use the following buttons located between the list boxes to move the queues to and from the list boxes.

- [>>] - You can click this button to add all the queues in the Available Queues list to the Selected Queues list
- [>] - You can click this button to add the selected queue(s) to the Selected Queues list
- [<] - You can click this button to add the selected queue(s) from the Selected Queues list to the Available Queues list
- [<<] - You can click this button to add all the queues in the Selected Queues list to the Available Queues list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

At the bottom of the window you will find the following buttons:

- **Assign** - You can click this button to assign the activity to the selected queues.
- **Close** - You can click this button to close this window.

Assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

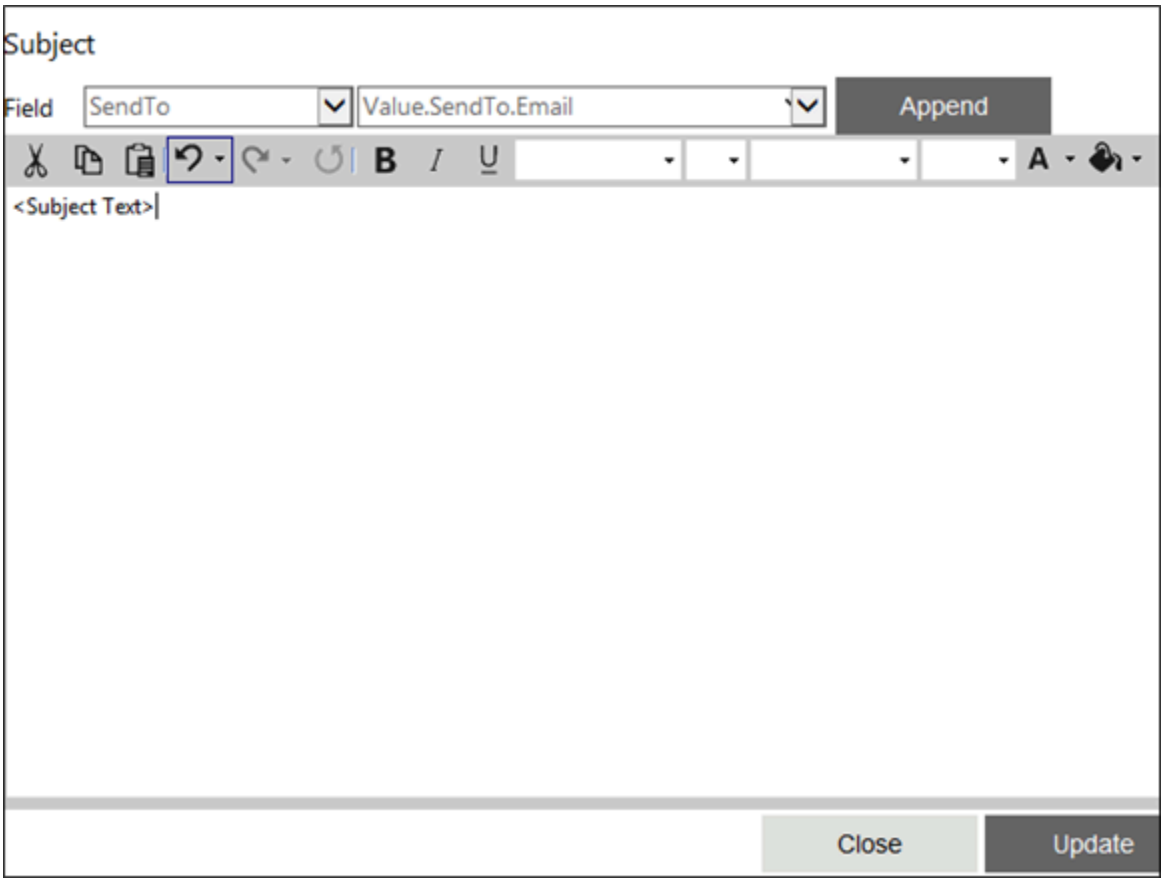
For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the Start activity with the Type as **String**. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

To assign multiple queues, specify the queue-ids with semicolon (;) separator. **Skelta.HWS.Queue.QueueCollection** class can be used to retrieve queue information.

Info - Subject

Subject Property Window

You can specify a subject for the Information work item assignment in this property window. To open this window, click the button for the 'Subject' property.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The window has a text editor to enter the subject. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the subject. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the subject in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the subject.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the subject using a combination of discrete value and variable or content. For example, you can specify the subject as "Submitted By <%#Variable.SubmittedBy.Email%>". Here the string "Submitted By" should be typed directly in the text editor. The second half of the subject is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the Field drop down and clicking Append.

The text tool bar above the text editor allows you to manage and format the subject. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the subject.
- **Close** - You can click this button to close this window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Info - Body

Body Property Window

You can specify a message Body for the Information work item assignment in this property window. To open this window, click the button for the 'Body' property.

The window has a text editor to enter the body text. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the body using a combination of discrete value and variable or content. For example, you can specify the body message as "Hi, Please approve the leave application submitted by <%#Variable.SubmittedBy.name%>". Here, the string "Hi, Please approve the leave application submitted by" must be typed directly in the text editor. The second half of the body is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the Field drop down and clicking Append.

The text tool bar above the text editor allows you to manage and format the body text. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

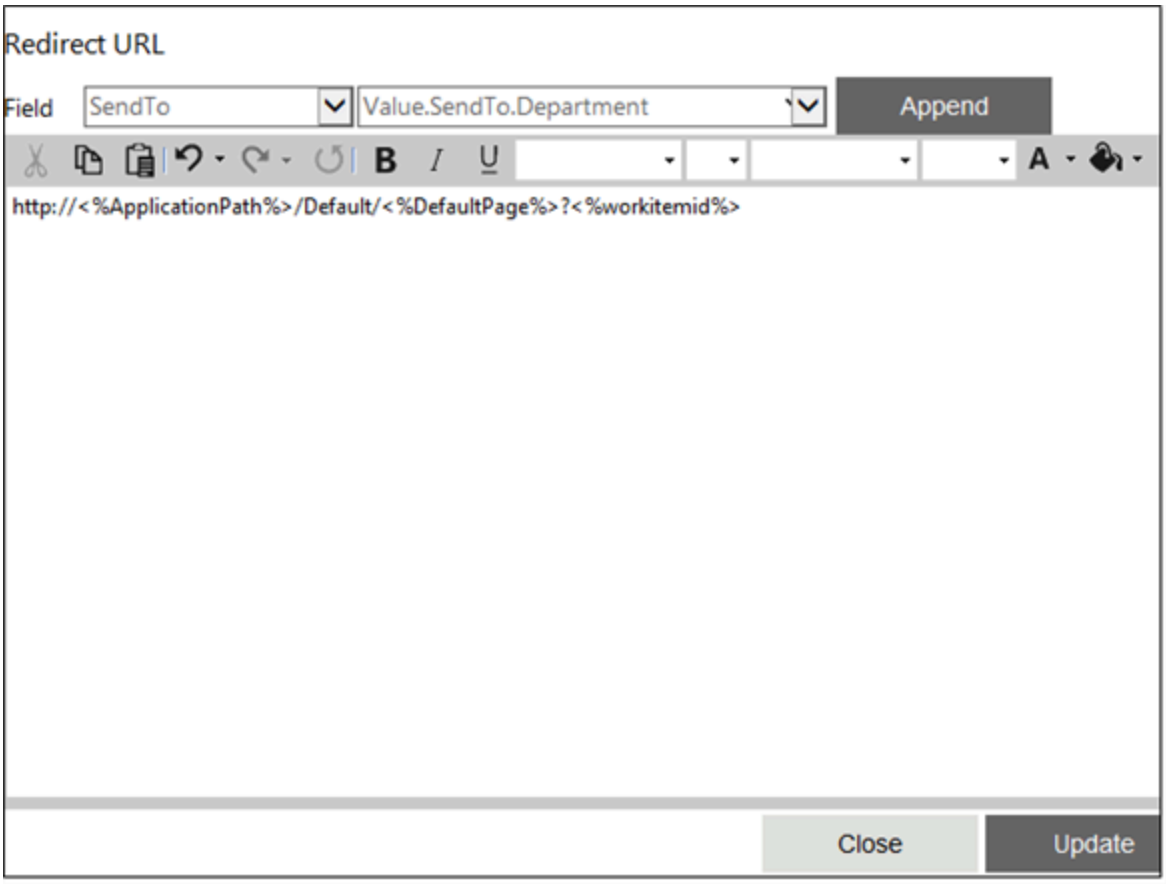
- **Update** - You can click this button to save the body message.
- **Close** - You can click this button to close this window.

Info - Redirect URL

Redirect URL Property Window

You can specify a message Redirect URL for the Information activity in this property window. To open this window, click the button for the 'Redirect URL' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the Redirect URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/ drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Redirect URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Redirect URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Redirect URL. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

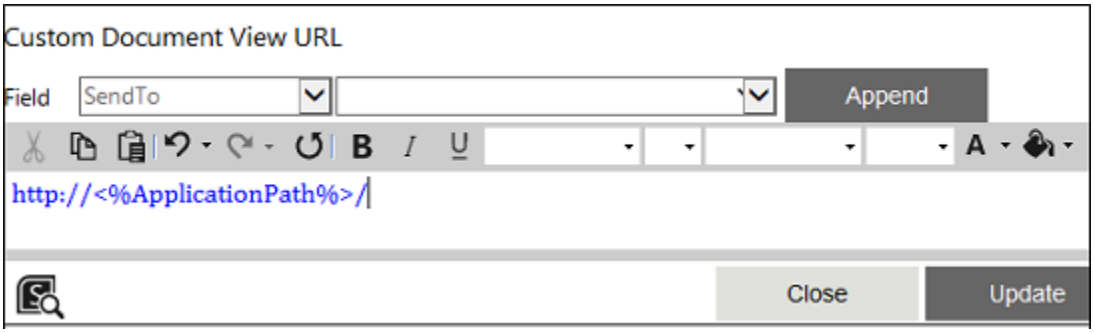
- **Update** - You can click this button to save the Redirect URL.
- **Close** - You can click this button to close this window.

Info - Custom Document View URL

Custom Document View URL Property Window

You can specify a message Custom Document View URL for the Information activity in this property window. To open this window, click the button for the 'Custom Document View URL' property.

For more details, see Custom Document View URL.



The window has a text editor to enter the Custom Document View URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

here. You can enter the Custom Document View URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Custom Document View URL.

- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Custom Document View URL. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the Custom Document View URL.
- **Close** - You can click this button to close this window.

Info - HTML Template

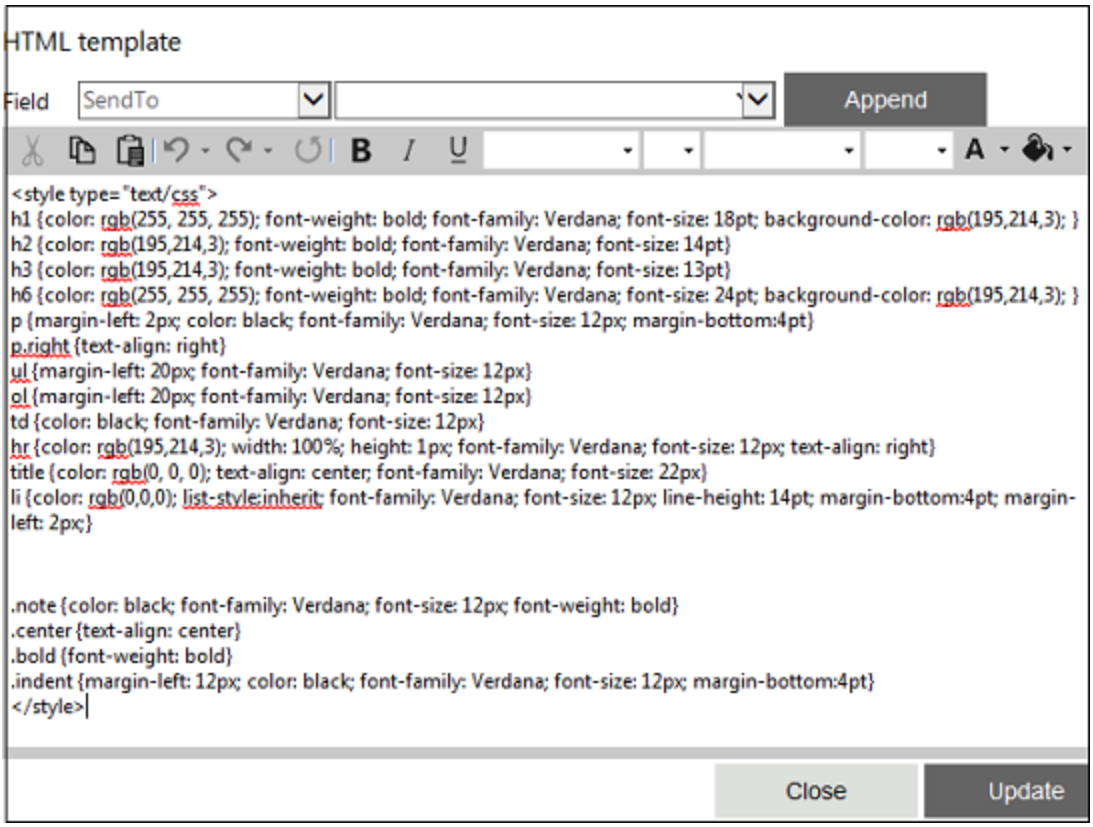
HTML template Property Window

You can specify a html template for the Information activity in this property window. To open this window, click the edit button for the 'HTML template' property.

In this property window, users can write their own html template for Information activity or make use of default Information activity template for customizing the email.

See Using 'HTML Template' property of Information Activity for more information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the body text. You can do this either by entering a html template directly in the text editor or by using the Field controls at the top of the window. These controls are described below:

The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.

The Value text box/drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.

Append - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the html template. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

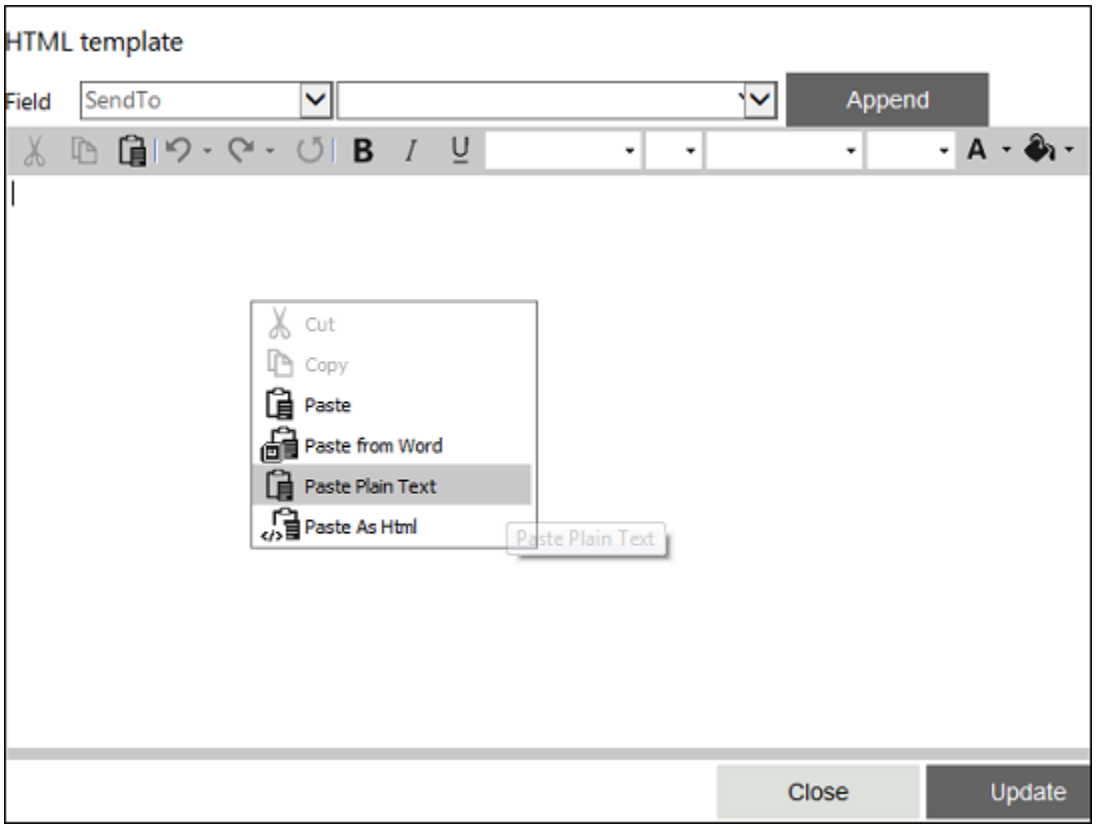
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

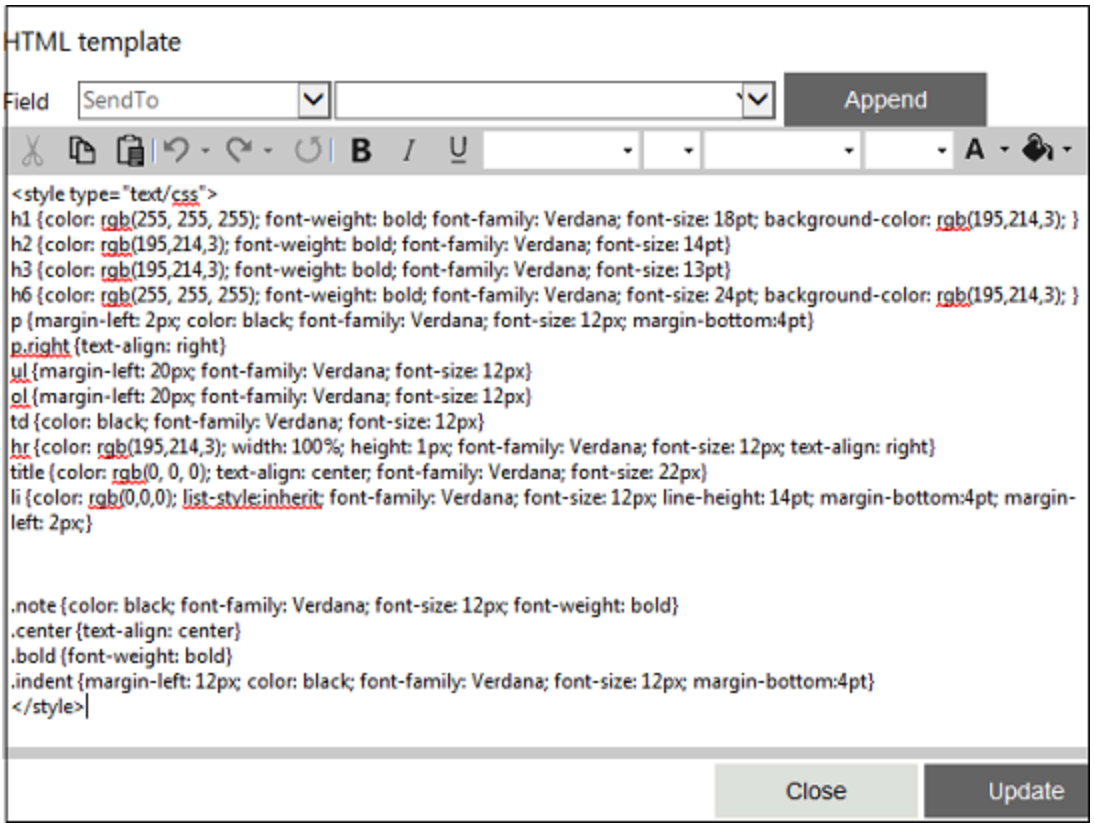
- **Update** - You can click this button to save the HTML template.
- **Close** - You can click this button to close this window.

To set HTML temple, do the following steps:

1. To specify the HTML template, click the edit icon to open the property window.
2. In the **HTML template** window, provide template which should be in html format. If you are pasting a copied html content, then right-click the **HTML template** window and paste it as "**Paste Plain Text**".



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



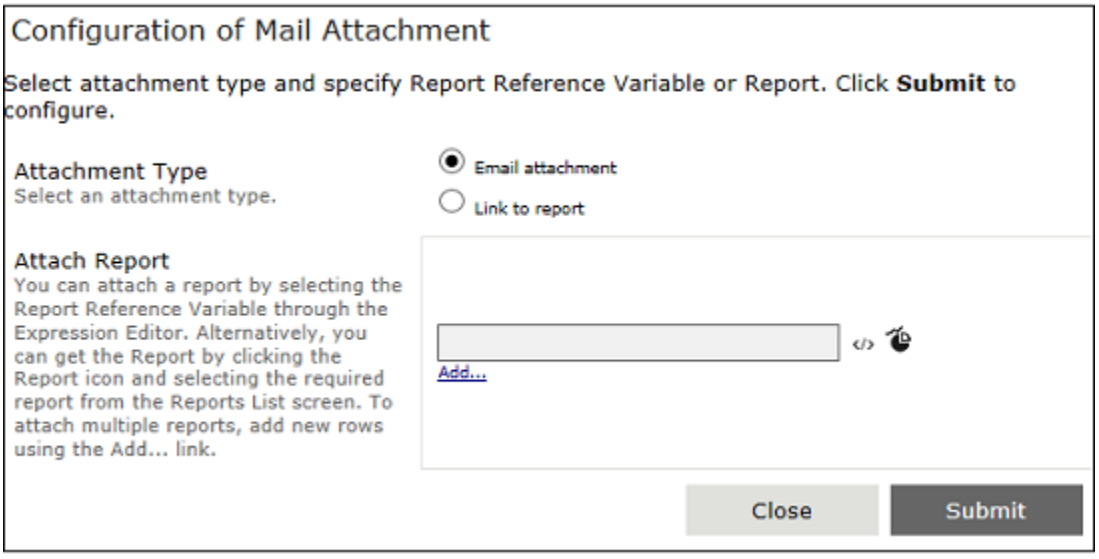
3. Click **Update**.

Info - Attach Reports

Configuration of Mail Attachment property window

You can configure the **Attachment** property for the **Approval** activity in this property window.

In this property window, you can configure the report attachments to be emailed.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

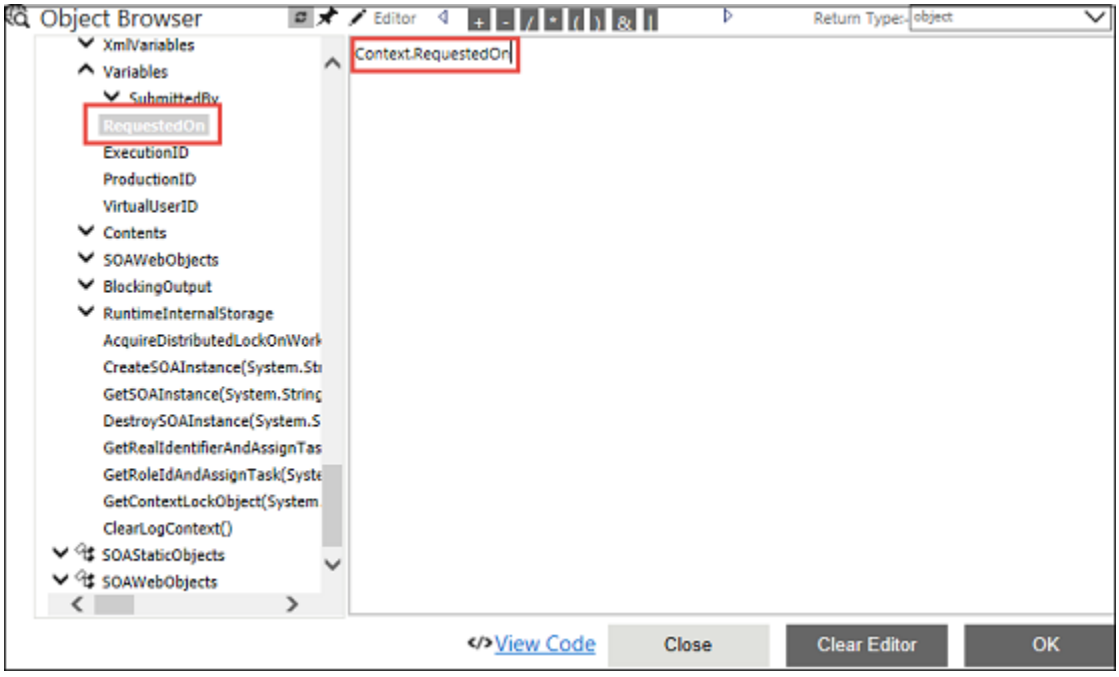
- **Attachment Type:** Use this property to select the type of attachment for sending the report as an email.
  - **Email attachment:** Select this type to send the report as an email attachment.
  - **Link to report:** Select this type to embed a link in the email to download the configured report.
- **Attach Report:** Use this property to attach a report generated by the **Report Generator** activity within the workflow or to select a report from the list of generated reports.

  You can attach a report by selecting the required **Report Reference Variable** using the **Expression Editor**. Alternatively, you can select the report from the Reports List window which consists of the already generated reports.

There must be at least one **Report Generator** activity before an **Approval** activity in the workflow to select the **Report Reference Variable** (using Expression Editor) as an **Attachment**. The **Report Reference Variable** is generated while configuring the **Report Generator** activity. By default, the report reference variable is **[Template Name]_[activity Name]**. For more information about **Report Reference Variable**, see Configure Report Template.

To configure mail attachment

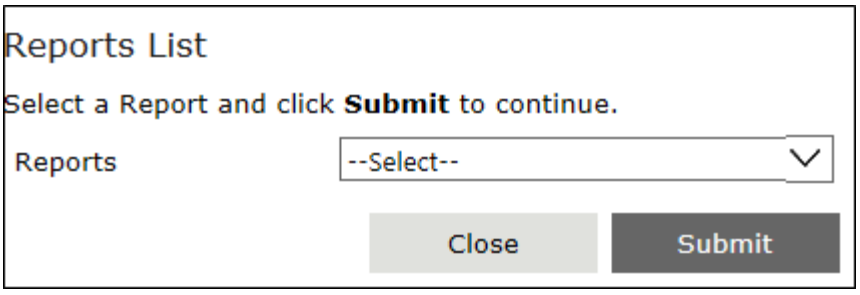
1. Click the edit button for the **Attachment** property.
2. Select the required **Attachment Type**.
3. In the **Attach Report** text box, you can either use the expression editor to get the variable of the report or use the **Reports List** to select the report.
4. To get the variable, click the **Expression Editor** icon given beside the text box to select the required **Report Reference Variable**.
5. In the **Expression Editor** window, select the required variable.



6. Click **OK.**
7. Click **Close**.

   The variable appears in the **Attach Report** text box.
8. To select the report from the Reports List dialog box, click the **Reports** icon.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



9. In the Reports List dialog box, select the required report, and then click **Submit**.
   This window closes automatically.
10. Click **Close** to close the configuration window.

Resource Set Activity

Activity Description

The Resource Set Activity is used in the AVEVA Work Tasks workflow process design when the actor for executing the subsequent activity needs to be selected by another actor in the previous activity of the workflow. This activity displays a list of actors in the Work items list of the actor who will make the selection.

Application Scenarios

A typical application scenario could be a Technical Support Request Workflow Process, in which the support request needs to be assigned to the appropriate technical support person for action. The Resource Set activity can be used to allow a supervisor to assign the support request on the basis of some condition.

Variables for Capturing Resource Details

The details of the resource submitting the ResourceSet response can be accessed

by declaring a variable of type resource with the following syntax:

'^'+Action Name+'RespondingUser'

This variable stores the responded user information. If it is more than one user, it will retrieve the last user details.

Note: For a single user, you can declare this variable as **Resource** type. If the task is assigned to more than one user and to get all responded users information, then declare this variable as an array.

Variables for assigning queue system variable:

To assign queues instead of setting the queue through **Assign Queue** property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the Start activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

To assign multiple queues, specify the queue-ids with semicolon (;) separator.
Skelta.HWS.Queue.QueueCollection class can be used to retrieve queue information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Variable for Queue Acting Users

You can declare an array type variable to get updated information about the name of the actor to whom an activity gets assigned from a queue.

In the Workflow Variables, define the variable in the following format:

^+CurrentActivityName+'QueueActingUsers'

Ex: ^HRFormQueueActingUsers

HRForm is the activity name.

For example, the value of the user who acts on the HRForm activity will be stored in the variable. The Approval1 task will be assigned to the same user.

Variable for storing timeout warning value

Suppose you want to store the timeout warning value for Task activity in your workflow. The naming convention for the system variable to be declared is

^<ActivityName>ResponseByWarningTimeout

The value returned by the variable will be in UTC format. For multiple user scenarios the highest timeout value calculated for a user will be available in the variable.

Note: Multiple timeout warnings cannot be tracked using this feature.

Variables for capturing Comments

To capture the comments entered by an actor, add a variable with a name prefixed by '^', with the following syntax:

'^'+_WorkItem.HWSActivity.ActivityName+'Comments'

For example, if the activity name is 'ResourceSet1' then declare the variable as ^ ResourceSet1Comments in the Start activity.

Note: By default, the comment field is not available in the "ResourceSet" activity.

To add a comment field to the ResourceSet activity, create a Textbox control named 'txtComment' in the 'ResourceSetUI.htm' page which is available under ("[AVEVA Work Tasks Installed Path]\WorkflowElements\Default\en-US\Activities\Activity folder").

Activity Properties

The Resource Set activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right-click menu for the Resource Set activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if necessary.)

Display Name - This property specifies the name to be displayed for the activity in the Work items list.

Property Type: Optional (This property need not be set if the display name is already configured in the Actions.xml. The name entered here overrides any display name that is configured for this activity in the Actions.xml file.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Delivery Channels - This property is used to specify the delivery channels to deliver this activity to the assigned actor(s). Multiple channels can be specified giving the actor(s) different options to act on the work item.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Delivery Channels property

1. To select the Delivery Channels, click the button for the 'Delivery Channels' property to open the property window.
2. In this window, the Delivery Channels that have been configured are listed on the left along with a check box against each channel. Click and select the check box corresponding to the channel you wish to select. You can select multiple channels by selecting the respective check box.
3. Click the **Save Channels** button.

See Approval Activity for a detailed description of the Delivery Channels property window.

Actor Selection Filter - You can define the list of actors who will be available to the user acting on the Resource Set activity using the Actor Selection Filter. The output of the Resource Set activity is the actor(s) set by the user who acts on the Resource Set activity. The actor receives the list of available actors as input through the Assign Actor drop-down.

Property Type: Mandatory (This property must be set for the activity to be used in the workflow.)

Steps to set the Actor Selection Filter property

1. To specify the actors, click the button for the 'Actor Selection Filter' property to open the property window.
2. In this window build an expression to select the actors using the actors properties. Select an attribute from the first drop down list. Select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See ResourceSet - Subject for a detailed description of the Actor Selection Filter property window.

See the Using XML Variables section in the Approval Activity topic for details on how to use the XMLVariables to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

build the expression.

Assign Actor - This property allows you to select the workflow variable to which the actor will be set by the actor acting on the Resource Set. The variables will have to be created in the properties for the Start activity. The variables can be one of the three types - String, Array or Resource. String variables take one UserID at a time, while Arrays take in a number of UserIDs pointing to different actors. Resource variables will take the complete resource properties. In a group action scenario, where more than one actor acts on the Resource Set activity, the selection of the last actor to act will be assigned to the variable.

Property Type: Mandatory (This property must be set for the activity to be used in the workflow.)

Work Item Fields - This property is used to add the Work Item Fields to be used to display application data as well as value in variables and content in the Work items list when this activity is executed.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Work Item Fields property

1. To select the Work Item Fields, click the button for the 'Work Item Fields' property to open the property window.
2. In this window, the Work Item Fields that have been set are listed in a grid. Click a row to select the corresponding Work Item Field.
3. Right click the row and select "SetWorkItemField" from the context menu to set it.

See ResourceSet - Work Item Fields for a detailed description of the Work Item Fields property window.

Actors

The properties in the Actors group have been described below. You can use these properties to specify the actors who will work on the activity and related properties.

Assign Actor(s) - This property specifies the actor(s) who has to take up the Resource Set activity. It is tagged with the Ignore Unavailable Resources property which in turn is tagged with Alternate Resource Allowed. This will be explained in detail later.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any actor(s).)

Steps to set the Assign Actor(s) property

1. To specify the actors, click the button for the 'Assign Actor(s)' property to open the property window.
2. In this window build an expression to select the actors using the actors properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking Undo.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See ResourceSet - Subject for a detailed description of the Assign Actor(s) property window.

See the Using XML Variables section in the About the XML Variable User Interface topic for details on how to use the XMLVariables to build the expression.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Assign Queue(s) - This property specifies the queue(s) to which the Resource Set activity is to be assigned.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any Queue(s).)

Steps to set the Assign Queue(s) property

1. To specify the queue(s), click the button for the 'Assign Queue(s)' property to open the property window.
2. In this window, select the queue(s) from the **Available Queues** list box. You can select multiple queues by selecting with the <Ctrl> key pressed.
3. After selecting the required queue(s), click the [>] button to move it to the **Selected Queues** list box. You can click the [>>] button to select all the queue(s) and move them to the **Selected Queues** list box.
4. You can remove any queue from the selected list by selecting it and clicking the [<] button. You can remove all the queues by clicking the [<<] button.
5. After adding the required queue(s), click **Assign** to assign the activity to the selection.
6. Click **Close** to close the window without saving the selection.

See ResourceSet - Subject for a detailed description of the Assign Queue(s) property window.

Min. Ownerships [% or Number] - This property, which can be specified as a number or percentage, is the minimum number of actors who should take ownership of the Resource Set request. This property is related to the 'Timeout warning for Ownership' and 'Timeout for Ownership' properties. To illustrate, when 'Timeout for Ownership' is reached, the activity first checks if the ownership count is less than Min.Ownerships (if it is set). If it is, the work item will be removed from all Work items lists. If the 'Min.Ownerships' has already been satisfied, then the work item will be removed from the Work items lists of those who have not taken ownership. If the Min.Ownerships value is 0, then the activity checks if the ownership count is less than Max.Ownerships (if it is set). If it is, the work item will be removed from all Work items lists. Else the work item will be removed from the Work items lists of those who have not taken ownership. If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work items lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Max. Ownerships [% or Number] - This property, which can be specified as a number or percentage, is the maximum number of actors who should take ownership of the Resource Set request. If 'Timeout for Ownership' is specified and achieved, and the 'Min.Ownerships' is also satisfied, then the activity waits for action from those who have taken ownership. If the Min.Ownerships value is set to 0, then it checks if the ownership count is less than Max.Ownerships. If it is, the work item will be removed from all Work items lists. Else the work item will be removed from the Work items lists of those who have not taken ownership. If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work items lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Enable Take Ownership? - If this property is set to **Yes**, then the actor is provided with a **Take Ownership** option in the Work items list to take ownership of the task. If you are not required **Take Ownership** option in the Work items list, then set this property as **No**.

Note: By default, this property is set as Yes.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to set the Enable Take Ownership? property

1. Select **Yes** or **No** option from the **Enable Take Ownership?** drop-down list.

Re-Assign Allowed? - If this property is set then the resource is provided with a Re-Assign option. Using this the actor can forward the work item to another actor.

Property Type: Optional (This property needs to be set only if necessary.)

Ignore Unavailable Actors? - This property provides the following options - **Yes**, **Yes** - **holidays excluded** and **No**. If this property is set to 'No', when the Approval activity is assigned to the actor's Work items list, a check is first done to see whether the actor is available. If the actor is not available then a check is made for an alternate actor assigned to the unavailable actor. Simultaneously, the value of the property 'Alternate Actor Allowed' is checked. If that is also set, then the Approval work item is assigned to this actor. If this property is set to Yes - holidays excluded, the workflow engine will ignore the fact that actors are not available if the day is an organization holiday and the output Not Enough Resources will not be generated.

Property Type: Optional (This property needs to be set only if necessary.)

Alternate Actor Allowed? - This property is set to assign a particular task to an alternate actor in the absence or unavailability of any actor.

Property Type: Optional (This property needs to be set only if necessary.)

Notification Contents

The properties in the Notification Contents group have been described below. You can use these properties to specify the content used to notify actors about the activity.

Subject - This is the subject of the Resource Set assignment. The subject can be customized for each actor.

Property Type: Optional (This property need not be set. If it is not set, the default subject line from the Actions.xml file is displayed.)

Steps to set the Subject property

1. To specify the subject, click the button for the 'Subject' property to open the property window.
2. In this window, enter the subject text directly in the text editor.
3. If you want to customize the subject you can do so using the **Field** controls to add resource property values. For example, to add a subject that includes the email id of the actor submitting the Resource set, you can do as follows: Type in the text "Submitted By". Select "Variable" and "Variable.SubmittedBy.Email" from the two drop downs for **Field** and click **Append**. This will add the placeholder for the variable after the text entered in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the email id.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the subject, click **Update** to save it.
7. Click **Close** to close the window without saving the subject.

See ResourceSet - Subject for a detailed description of the Subject property window.

Body - This is the actual message of the Resource Set sent to the actor(s). The body can be customized for each actor. There is an **Enterprise Console Explorer**(plug-in) icon in the Body property window to get the work item details URL. See *Enterprise Console Control (EC Plug-in)* for more information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property need not be set.)

Steps to set the Body property

1. To specify the message body, click the button for the 'Body' property to open the property window.
2. In this window, enter the body text directly in the text editor.
3. If you want to customize the body you can do so using the **Field** controls to add resource property values. For example, to add a body that includes the name of the actor submitting the Resource set, you can do as follows: Type in the text "Hi, Please cast your vote in the poll conducted by". Select "Variable" and "Variable.SubmittedBy.name" from the two drop downs for **Field** and click **Append**. This will add the placeholder for the variable after the text entered in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the name.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the body, click **Update** to save it.
7. Click **Close** to close the window without saving the body.

See ResourceSet - Body for a detailed description of the Body property window.

From Email Address? - This property is used to specify the email address from which notification messages will be sent for the activity.

Property Type: Optional (This property needs to be set only if necessary. If it is not set, the default value in Activities.xml will be used.)

Hide Responses? - If this property is set to 'No', then actors can see the responses of other actors in a group Resource Set scenario. To view these responses i.e., ownerships, Resource Sets etc., select the 'Take Ownership' option. This will open a new pop up with the responses of other actors who have taken ownership.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Priority - This property is set to indicate the priority of the Resource Set request. Enter the values 0-33 to set as Low, 34-66 for Medium, or 67-99 for High. The default value is set to 34.

Property Type: Optional (This property needs to be set only if necessary.)

Show Custom Document View Window - If this property is set to 'Yes', then the Custom Document View Window will be shown to actors in their Work items lists.

Property Type: Optional (This property needs to be set only if necessary.)

Send Notification Email - If this property is set to 'Yes', then notification emails will be sent to the actors. If it is set to 'Yes-HTML', then the emails will be sent in HTML format.

Property Type: Optional (This property needs to be set only if necessary.)

Notify if alternate actor is considered - If this property is set to 'Yes', then notification emails will be sent to the actors when an alternate actor is considered.

Property Type: Optional (This property needs to be set only if necessary.)

Redirect URL - This property is used to get the URL of the web page to redirect the client to, in the case of synchronous execution of the activity.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to set the Redirect URL property

1. To specify the Redirect URL, click the button for the 'Redirect URL' property to open the property window.
2. In this window, enter the Redirect URL directly in the text editor.
3. If you want to customize the Redirect URL you can do so using the **Field** controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for **Field**, and clicking **Append**. This will add the placeholder for the variable in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the URL.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Redirect URL, click **Update** to save it.
7. Click **Close** to close the window without saving the Redirect URL.

See ResourceSet - Redirect URL for a detailed description of the Redirect URL property window.

Custom Document View URL - This property is used to get the URL of the page to which the custom document dlls does a post. The data posted to the page are application, workflow, executionid, executiondetailsid, and workitemid. The posted data can be retrieved as Request.Form ["workitemid"].

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Custom Document View URL property

1. To specify the Custom Document View URL, click the button for the 'Custom Document View URL' property to open the property window.
2. In this window, enter the Custom Document View URL directly in the text editor.
3. If you want to customize the Custom Document View URL you can do so using the **Field** controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for **Field**, and clicking **Append**. This will add the placeholder for the variable in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the URL.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Custom Document View URL, click **Update** to save it.
7. Click **Close** to close the window without saving the Custom Document View URL.

See ResourceSet - Custom Document View URL for a detailed description of the Custom Document View URL property window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Decision Influencers

The properties in the Decision Influencers group have been described below. You can use these properties to specify some parameters that influence the outcome of the activity.

Max Actors Required For ResourceSet Completion [% or Number] - This property is used to check what percentage or number of actors should take action for Resource Set. Decision on the Resource Set is taken only after the specified number or percentage of actor(s) have taken the action. The activity waits for the actors' action until the 'Timeout for activity' is reached.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Min Actors Required For ResourceSet Completion [% or Number] - This property specifies the minimum number of actors to act on the Resource Set before the 'Timeout Warning for Activity'. If the activity limit is reached before the 'Timeout Warning for Activity' then the warning message is not shown.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Escalation & Timeout

The properties in the Escalation & Timeout group have been described below. You can use these properties to specify escalation and timeout settings for the activity.

Set Calendar for Timeout - This property is used to specify the calendar to be used to determine activity timeouts.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to Set Calendar for Timeout

1. To specify the calendar for timeout, click the button for the 'Set Calendar for Timeout' property to open the property window.
2. In this window select the required calendar by clicking the calendar name in the tree structure.
3. Click **Set Calendar** to set the selected calendar.
4. You can specify that a resource calendar should be used, if available, by clicking to select the corresponding check box.

See ResourceSet - Set Calendar for Timeout for Timeout for a detailed description of the Set Calendar for Timeout property window.

Minimum available time for Activity - This property is used to check the availability of an actor to whom a task must be assigned while executing the activity. The availability check for the actor is done based on the Calendar settings (Global or Resource Calendar). If the time taken to complete the task is within the value set for the **Minimum available time for Activity** property, then the task will be assigned to that actor. However, if the time taken to complete the task exceeds the value set for the **Minimum available time for Activity** property, then the task will be assigned based on the values set for the **Ignore Unavailable Actors?** and **Alternate Actor Allowed?** properties.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout warning for Ownership - This property specifies the time at which a warning is provided to an actor to take ownership of the work item assigned to him/her. If the actor does not take ownership within this time, the engine is notified with a warning. Thus if 'Min.Ownerships' is set, and it is not satisfied before the given time then a warning is sent. If Min.Ownerships is 0, it will check for Max.Ownerships. If the ownership count does not reach the Max.Ownerships value within the set time, a warning is sent. If both Min.Ownerships and

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Max.Ownerships are not set, this property is ignored.

Property Type: Optional (This property needs to be set only if necessary.)

Understand how Ownership Properties relate to each other:

- Resource Set activity is assigned to the actors in the Resource Set list by the workflow engine. Resource Set activity has a default property set that at least one actor has to make Take Ownership of the activity.
- A drop-down interface has been provided for the resource to Take Ownership. However, the actor can act on the Resource Set without taking ownership first, in which case ownership will be considered to be done. i.e. if the actor acts upon the Resource Set activity without ownership, then it is considered as an ownership with action taken.
- After the actor has taken ownership, a link is provided to the actor, which opens a pop up with the details of the other actors. This is again tagged with the 'Hide Responses?' property. The link will be provided only If 'Hide Responses?' is set as 'No'. The pop up will have details like Actor name, Ownership Status, Action taken, Comments.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] then the 'Timeout Warning For Ownership' property is not considered.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] and the 'Time Out For Ownership' is also reached then the work item is removed from other actors' Work items list and the work item waits for action from those who have taken ownership.
- If the 'Min.Ownerships' limit is not reached for the Resource Set activity, and the 'Timeout Warning For Ownership' has been exceeded then the actor is notified with a warning by including an email or an information activity, with the Timeout Warning for Ownership link to it.
- If 'Timeout For Ownership' is set, if the Resource Set activity has not reached the minimum limit (Min.Ownerships) and the 'Time Out For Ownership' has been reached, then the work item is removed from the Work items lists of all the actors.
- If 'Min.Ownerships' = 0 and if 'Max.Ownerships' is set, the activity waits for actors to take ownership or action until the 'Time Out For Ownership' is reached (if this is set).
- If none of the Ownership properties are set then the activity waits until the actor takes ownership or acts upon it.

Timeout for Ownership - This property specifies the time until which actors are allowed to take ownership, i.e. the time until which the Resource Set action will wait for actors to take ownership. If Min.Ownerships is set and is not reached, the work item is removed from all Work items lists. If Min.Ownerships is 0, it will check if ownership count is less than Max.Ownerships. If it is, the work item is removed from all Work items lists. Else the work item will be removed from the Work items lists of those who have not taken ownership. The work item will remain in the Work items list of those who have taken ownership and wait for action from them.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout warning for Activity - This property specifies when a reminder has to be provided for the actors to act on the Resource Set request, if the action count has not been reached. If properties such as Max.Ownerships, Affirmative Action for Resource Set (AAA), and Action Limit (AL) are set, once the property values are reached then a warning is sent to the actor. When the timeout is reached, the following checks are made: If AL is set, when the action count i.e. the no. of Resource Sets (AC) < AL, then a warning is sent. If AL = 0 and AAA is set, and when AC < AAA, then a warning is sent to the actor.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Timeout for Activity - This property's value is used to decide until what time the activity should wait for the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

actors' action. If properties such as Max.Ownerships, AAA, AL are set, once the property values are reached the work item is removed from the Work Item lists of all the actors. When the timeout is reached, the following checks are made: If AL is set, when the action count i.e. the no. of Resource Sets or rejections (AC) < AL, then the timeout for action is performed and the work item is removed from all Work items lists. If AC > AL, it checks with the no. of Resource Sets or 'decision count' (DC) and AAA. If DC >= AAA (if it is set), the action is considered completed. If AAA = 0 and if DC >= AL, the action is considered completed. If AL = 0 and AC < AAA, then the timeout for action is performed and the activity is removed from all activity lists. If AL = 0 and AC > AAA, it will check with DC. If DC > AAA it is completed. If AL = 0 and AAA = 0, then if DC is at least 1, then it is considered completed.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Multiple Timeout Warning - This is used to set recurring timeout warnings that will be sent to the actors who have to take action. To set multiple timeout warnings, click the 'Multiple TimeOut Warning' button in the Properties pane for the Resource Set activity. A new window pops up. In this window, set the interval for the recurring warning as well as the output message. You can also set the Recurring Time Interval. This can be used to set a different recurring time out after the first timeout happens. For example, if 'Time Interval' for the property is set as 7 days and 'Recurring Time Interval' is set as 2 days, then the first time out will happen after 7 days and there will be a recurring time out every second day after the seventh day (first) time out.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Steps to set the Multiple TimeOut warning property

1. To set the multiple timeout warning, click the button for the 'Multiple TimeOut Warning' property to open the property window.
2. Enter the time interval for the timeout in the **Time Interval** field.
3. Select "Yes" from the drop down for the **Recurring**? field if you want the timeout warning to recur. Select "No" otherwise.
4. Enter the output message corresponding to the timeout warning in the **Output** field.
5. Enter the time interval for the recurring timeout in the **Recurring Time Interval** field.
6. After entering the content details, click **Add Row** to add this timeout setting and display a new row for entering another timeout setting.
7. Remove rows if necessary by selecting them and clicking **Remove Selected**.
8. After specifying the timeout settings, click **Update**.

See ResourceSet - Multiple TimeOut Warning for a detailed description of the Multiple TimeOut Warning property window.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Redirection Enabled? - This property enables redirection. When this property is set to Yes, the workflow redirects the user to the URL in the Redirect URL property in the next activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Outputs

The Resource Set Activity returns the following values -

Action Completed: Displayed when all the actors have taken part.

Action Completed - Affirmative Percentage Not Reached: Displayed when all the actors have not taken part.

Not Enough Resources to Take Acknowledge: Displayed when the filter condition did not yield enough actors to whom the request can be sent.

Timeout - Acknowledge: Displayed when the specified minimum number of actors did not take ownership of the request in the given time.

Timeout - Action: Displayed when some or all of the specified actors did not perform the required action in the given time.

Timeout Warning - Acknowledge: Displayed when a warning is generated that the specified minimum actors have not taken ownership of the request in the given time.

Timeout Warning - Action: Displayed when a warning for timeout is generated because some or all the actors did not perform the required action in the given time.

Note: The following outputs will return only if user assigned tasks to a queue user.

Resource Unavailable for Queue - If a queue is assigned to a task & if there are no resources available in the queue then this output will get triggered.

TimeOut for Queue Task - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting the completion of that task.

Timeout Warning -Queue - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting in sending a warning out put. However, the task will still be sleeping.

Timeout Warning -Queue Participant - This output will get triggered when the task in the queue is picked by user but still not acted on that task within the queue escalation time out settings resulting in triggering a warning output. However, the task will still be sleeping.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

ResourceSet - Delivery Channels

Purpose

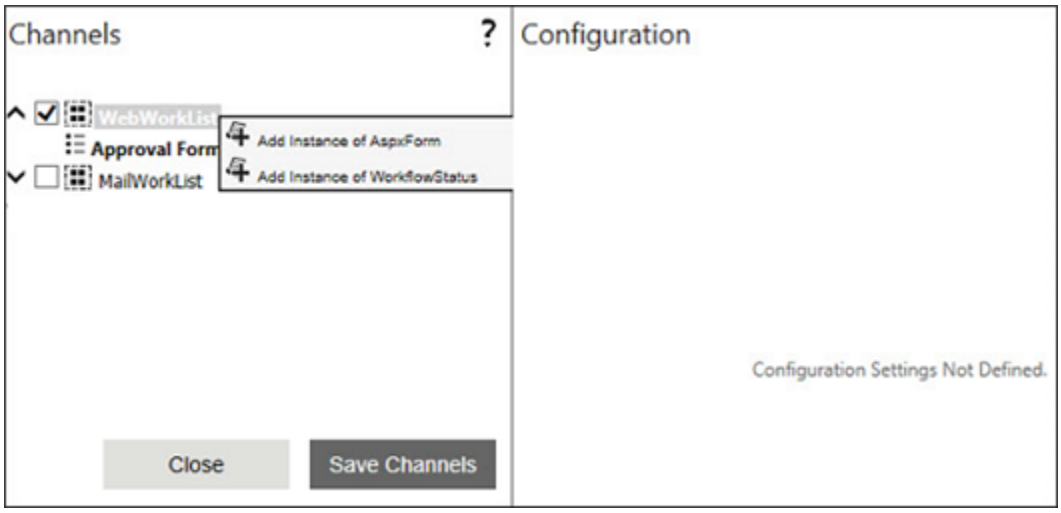
The purpose of this chapter is to describe the feature of WorkflowStatus Form of Delivery Channels property under the category Configuration of any human activity available in AVEVA Work Tasks.

Description: This feature enables us to view the status of the workflow in the work item list in the graphical form. [Optional]

1. Select the Human Activity.
2. Click the **Delivery Channels** Property.

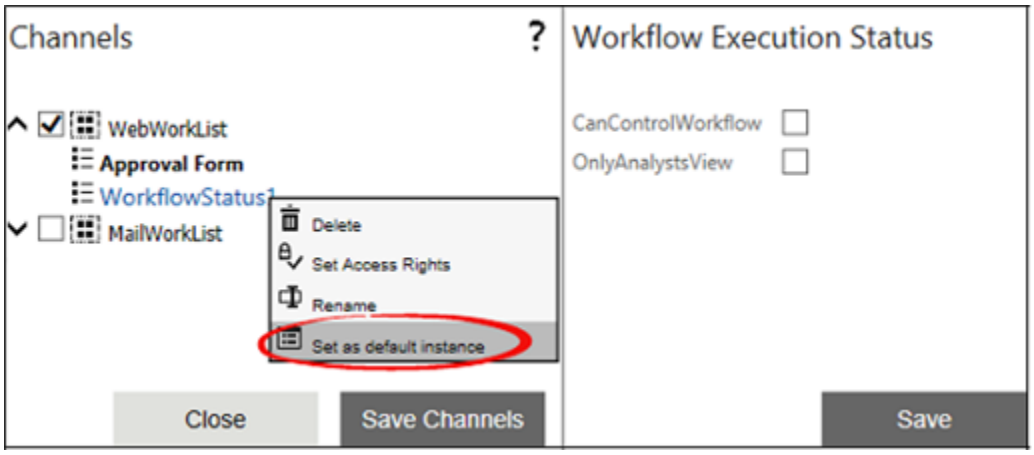
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

This will open a window like the one shown below.



3. Right-click **WebWorkList** to display the context menu with the **Add Instance of Workflowstatus** option as shown below.

4. Set the Properties for the Execution View Status Graphical view.
5. Click **Save**.
6. If you want to set the workflowstatus form as the default instance, right-click **WorkflowStatus1**, and select the **Set as default instance** option from the context menu.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. Click **Save** and **Save Channels**.

Output

Now, when you click on any row in the Work items list, you will find this Execution view page being displayed as the detailed view if you set it as a default instance.

Otherwise, workflow status form will appear in the context menu of each row in the Work items list and you can open the execution view by selecting it from the context menu.

ResourceSet - Actor Selection Filter

Actor Selection Filter Property Window

You can set the actors to whom the Resource Set work item is to be sent in this property window. To open this window, click the button for the 'Actor Selection Filter' property.

The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

Different Methods of Assigning an Actor

- Assigning an Actor by building an expression:

Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- Add - You can click this button to add the new expression to the text area.
- Undo - You can click this button to remove a condition you have added to the expression.

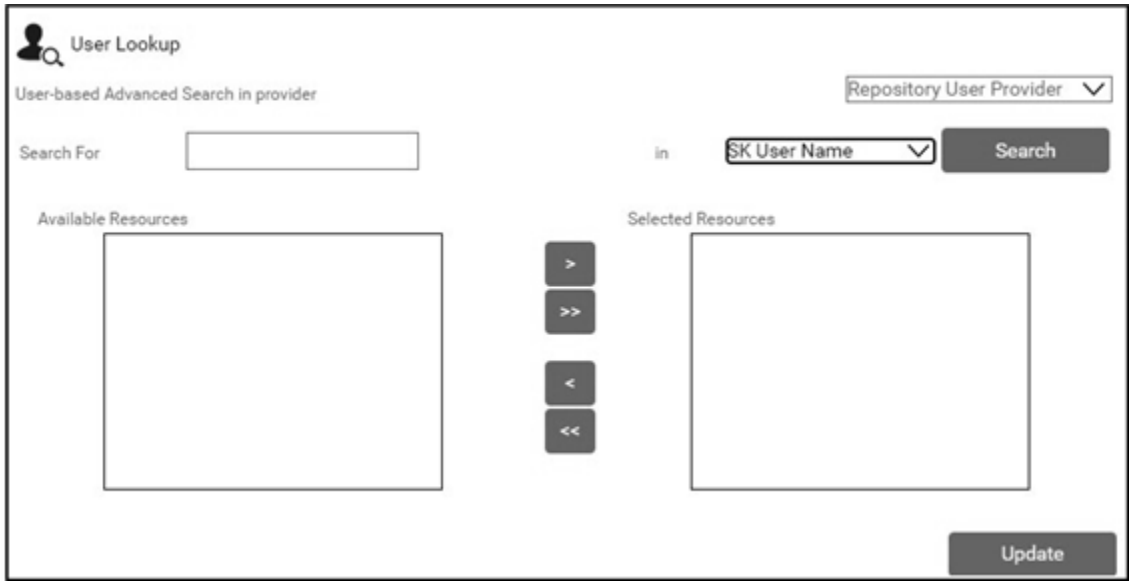
At the bottom of the window you will find the following buttons:

- Clear - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- Update - You can click this button to save the expression that has been added to the text area.

- Assigning an Actor through User Lookup

Through the User Lookup you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.
2. The **User Lookup** dialog is displayed, search and select the required actor.



3. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

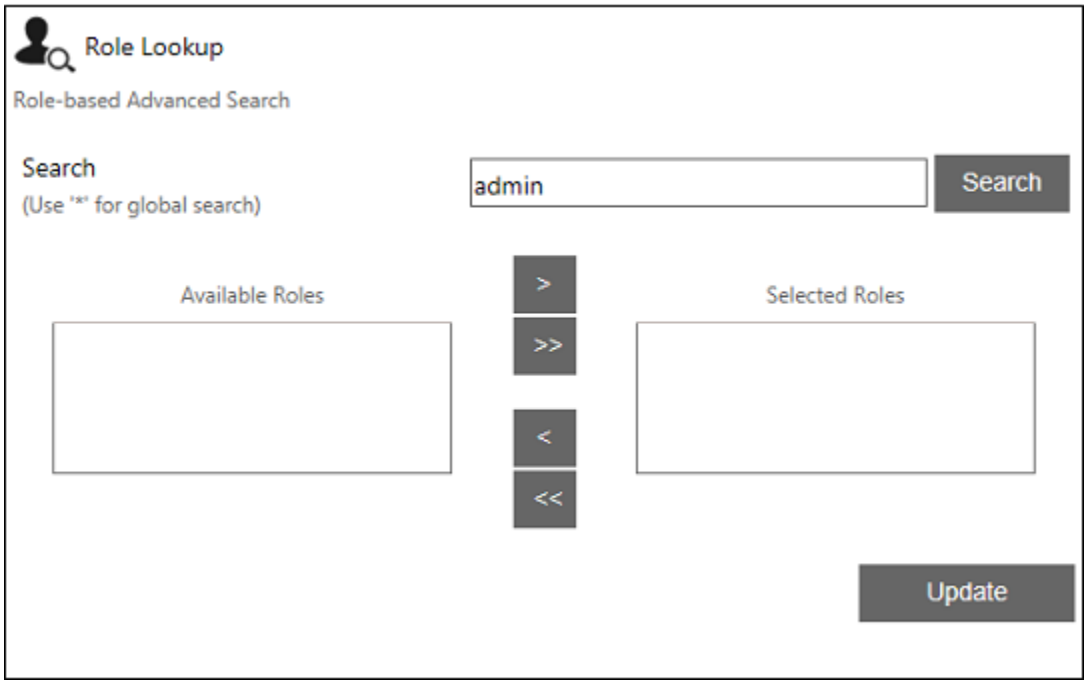
4. Click Update in the Assign Actor(s) window to update the selected actor(s) in the Assign Actor property.
5. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

- Assigning an Actor through Role Lookup

  Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

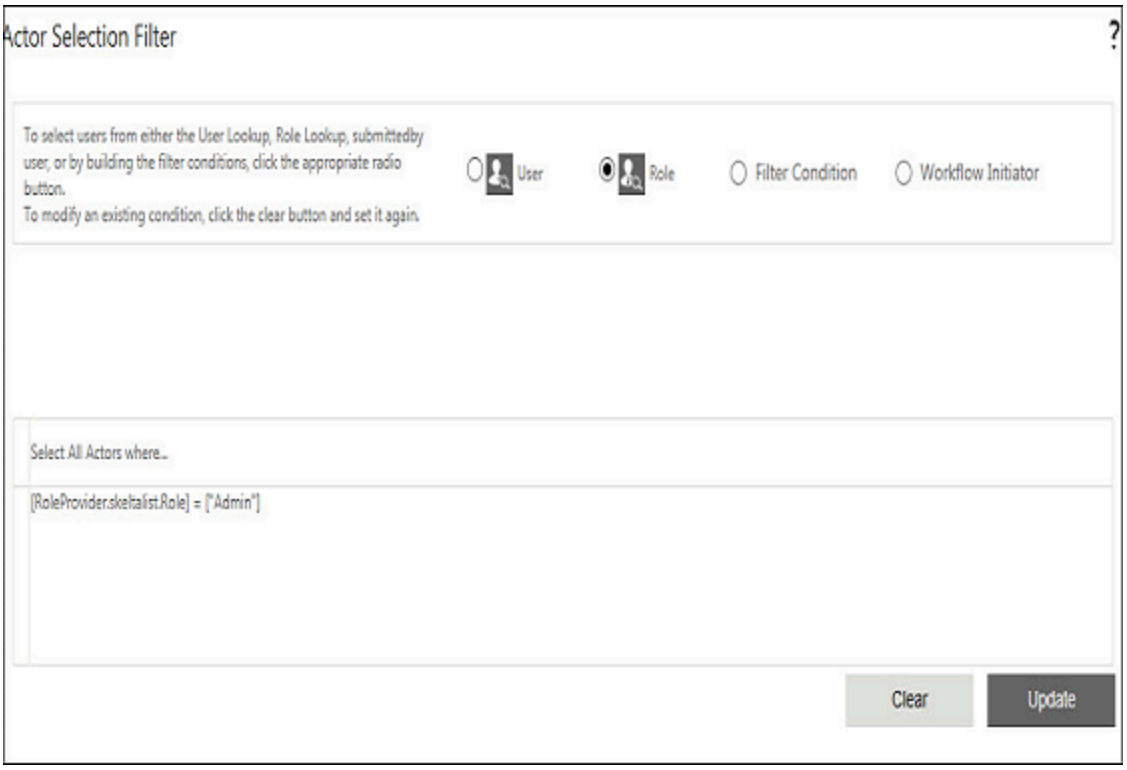
  Select the **Role Lookup** option.

  The **Role Lookup** dialog is displayed, search and select for the required role.



1. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
   In the Assign Actor window, you will see that an expression is displayed, refer the image below:

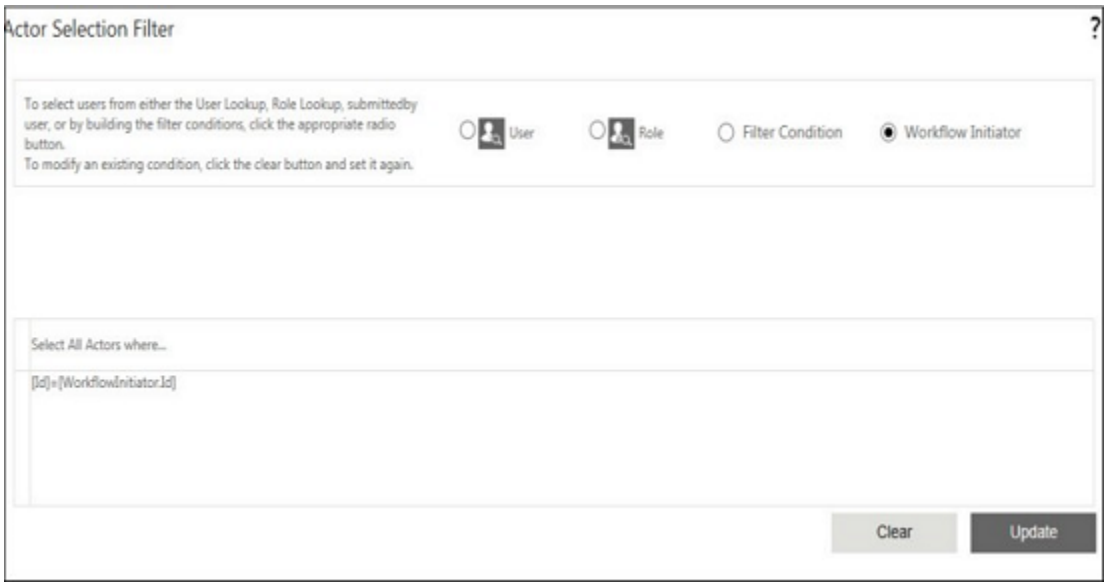
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



2. Click Update in the Assign Actor(s) window to update the Assign Actor property.

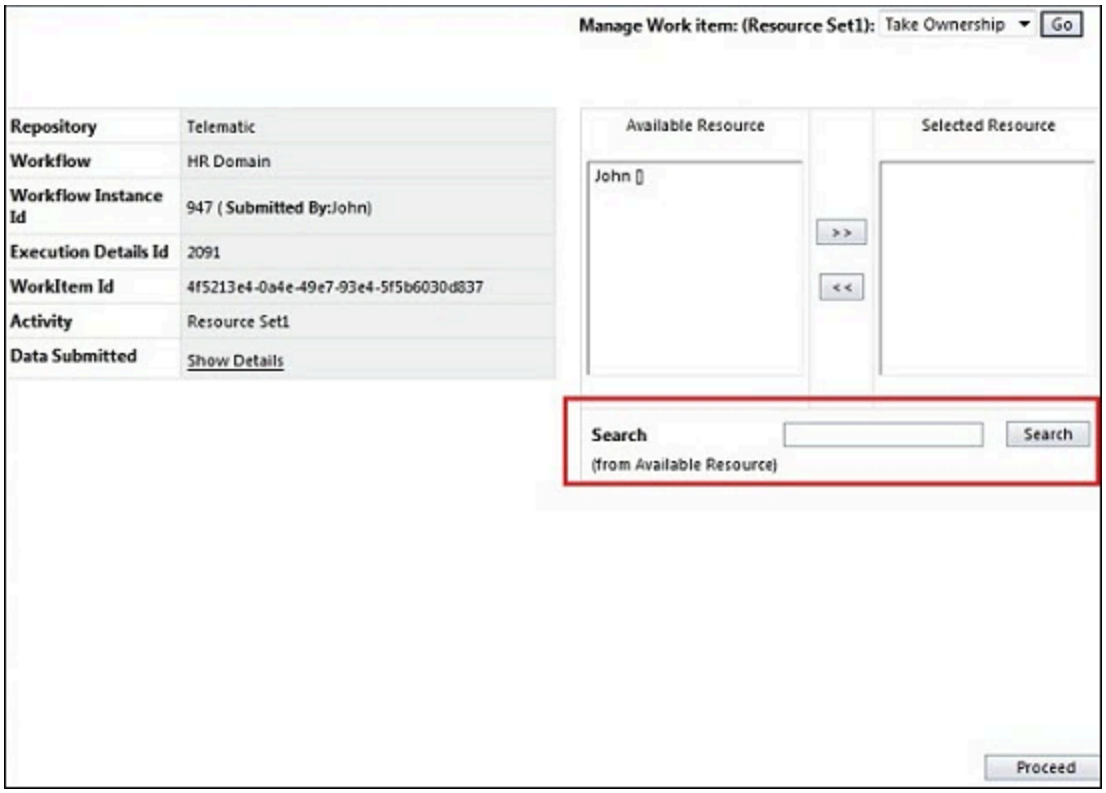
- Assigning an Actor through Workflow Initiator

  Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:



Note: When Actor Selection Filter is set through User Lookup and Filter Condition, the Work Item Detail view will have a Search option available. This search option allows you to perform a search within the Available Resource. Refer image below:

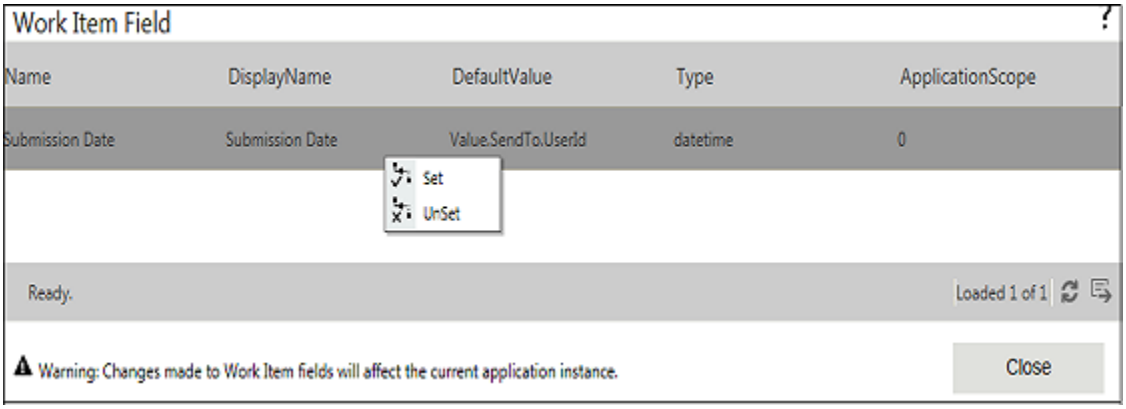
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



ResourceSet - Work Item Fields

Work Item Fields Property Window

You can set the Work Item Fields for this activity in this property window. To open this window, click the button for the 'Work Item Fields' property.



The window has a grid listing all the Work Item Fields that have been set for the process.

- Each row in the grid represents a Work Item Field. You can click a Work Item Field to select it.
- The context menu has the Set WorkItemField option, which you can use to set the selected Work Item Field for this activity. It also has the UnSet WorkItemField option to remove the Work Item Field setting.
- **Close** - You can use this button to close the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: While entering a value in the Work Item field, it is recommended to keep it less than the specified database column size. If there is a need to increase the size of the Work Item field, the user should manually increase the column size in the database.

ResourceSet - Assign Actor(s)

Assign Actor(s) Property Window

You can set the actors to whom the Resource Set work item is to be sent in this property window. To open this window, click the button for the 'Assign Actor(s)' property.

See Assigning a Task to a Manager for more details.

The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

Different Methods of Assigning an Actor

- Assigning an Actor by building an expression:

Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

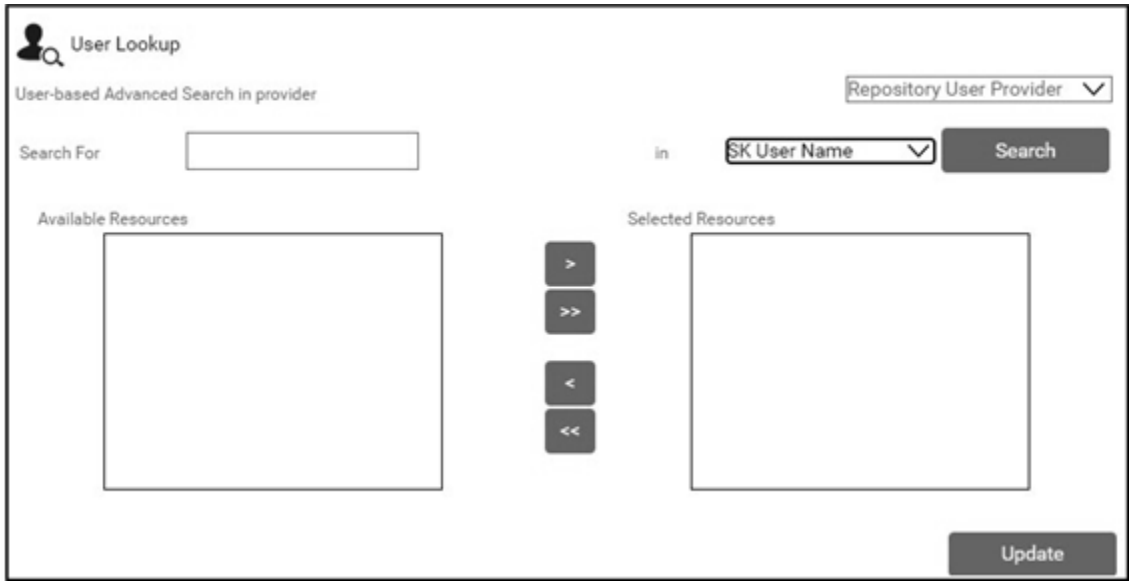
At the bottom of the window you will find the following buttons:

- Clear - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- Update - You can click this button to save the expression that has been added to the text area.

- Assigning an Actor through User Lookup

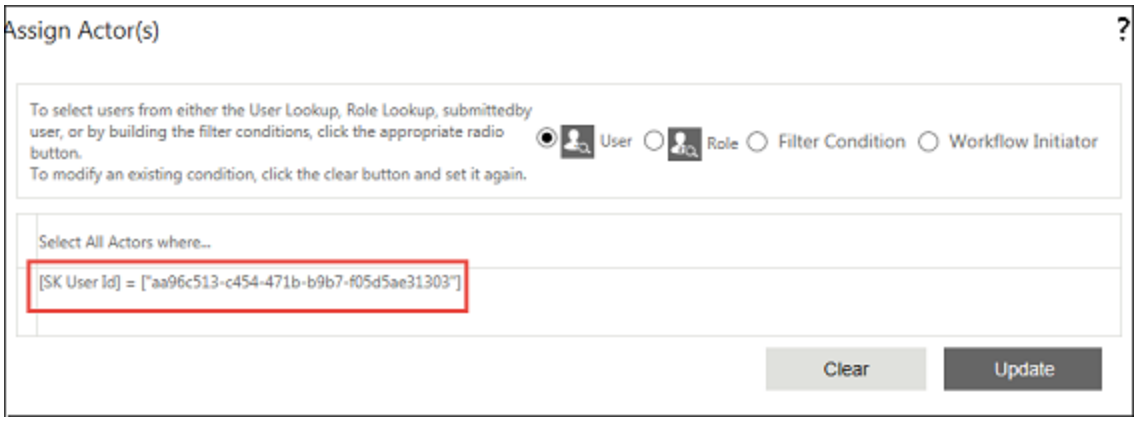
Through the User Lookup you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup.

1. Select the **User Lookup** option.
2. The **User Lookup** dialog is displayed, search and select the required actor.



3. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
4. In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

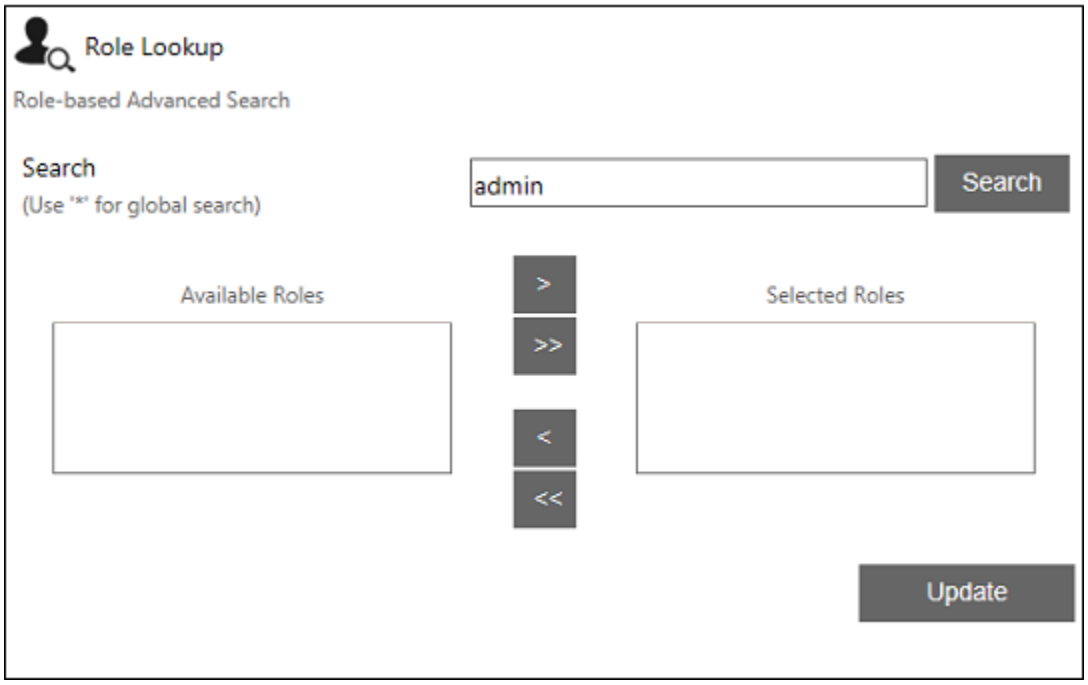


5. Click **Update** in the Assign Actor(s) window to update the selected actor(s) in the **Assign Actor** property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

- Assigning an Actor through Role Lookup

Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

1. Select the **Role Lookup** option.
2. The **Role Lookup** dialog is displayed, search and select for the required role.



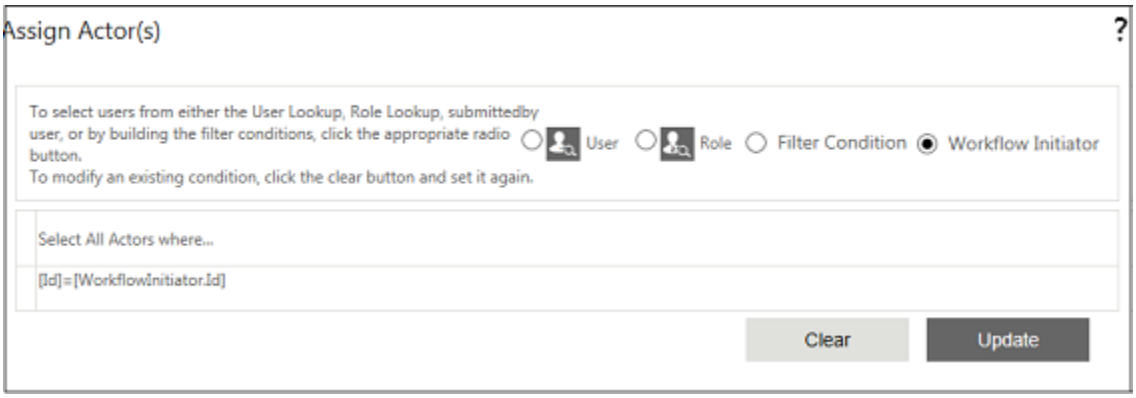
3. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click **Update** in the Assign Actor(s) window to update the `Assign Actor` property.

- Assigning an Actor through Workflow Initiator

Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

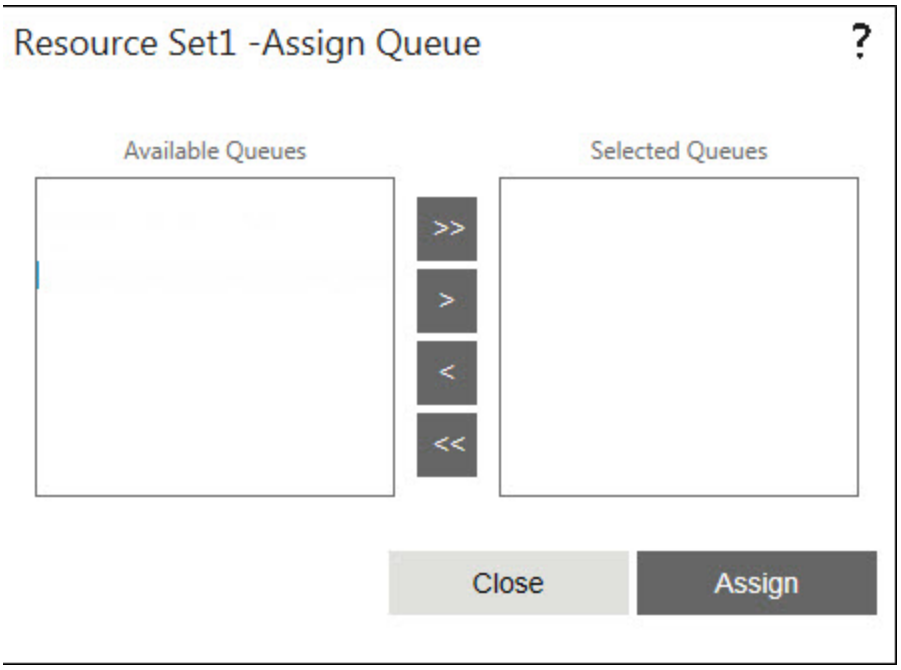


ResourceSet - Assign Queue(s)

Assign Queue(s) Property Window

You can select the queues to which the Resource Set work item is to be sent in this property window. To open this window, click the button for the 'Assign Queue(s)' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Assign Queue(s) Properties window for Resource Set

The window has a pair of list boxes to specify the queue(s). The Available Queues list shows all the available queues. You can select the required queue(s) from this list box. The Selected Queues list shows all the selected queues to which the activity will be assigned.

You can use the following buttons located between the list boxes to move the queues to and from the list boxes.

- [>>] - Click this button to add all the queues in the Available Queues list to the Selected Queues list
- [>] - Click this button to add the selected queue(s) to the Selected Queues list
- [<] - Click this button to add the selected queue(s) from the Selected Queues list to the Available Queues list
- [<<] - Click this button to add all the queues in the Selected Queues list to the Available Queues list

At the bottom of the window you will find the following buttons:

- **Assign** - You can click this button to assign the activity to the selected queues.
- **Close** - You can click this button to close this window.

Assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the Start activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

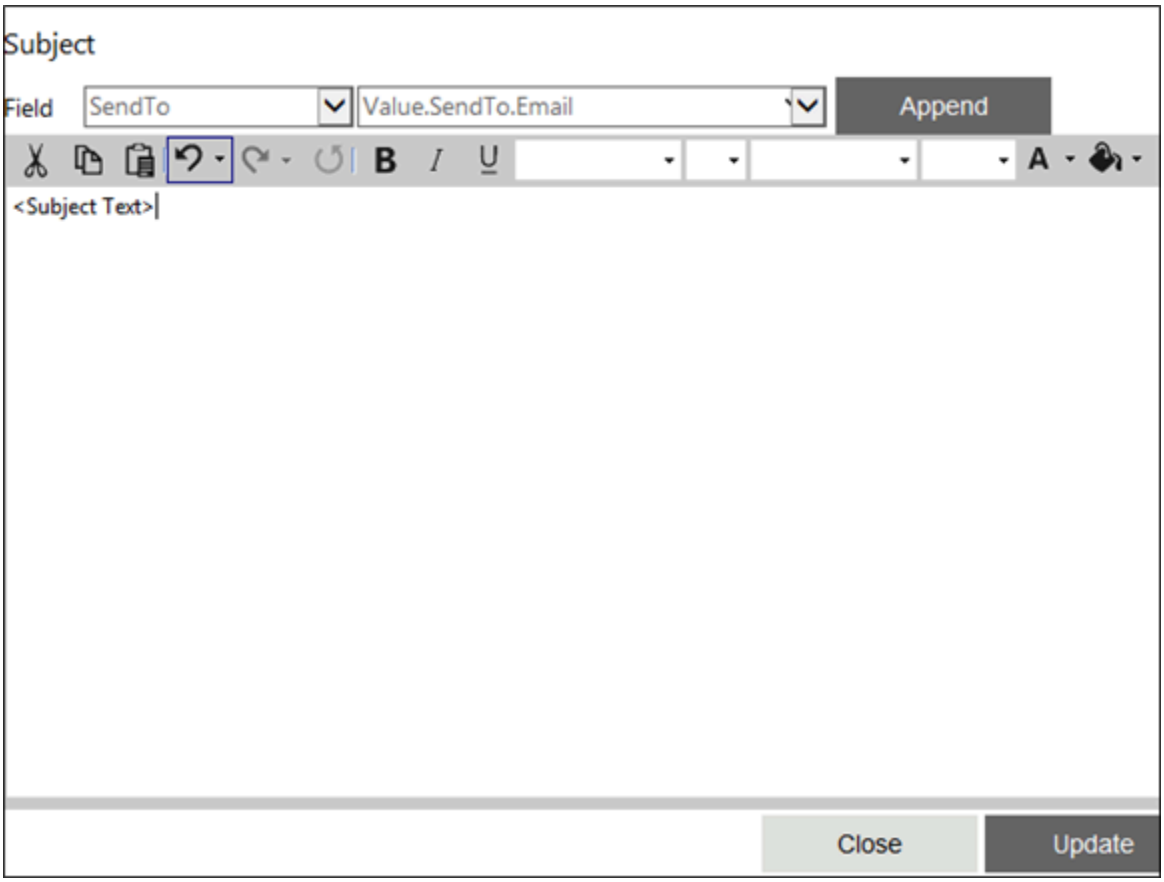
To assign multiple queues, specify the queue-ids with semicolon (;) separator. Skelta.HWS.Queue.QueueCollection class can be used to retrieve queue information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

ResourceSet - Subject

Subject Property Window

You can specify a subject for the Resource Set work item assignment in this property window. To open this window, click the button for the 'Subject' property.



The window has a text editor to enter the subject. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the subject. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the subject in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the subject.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the subject using a combination of discrete value and variable or content. For example, you can specify the subject as "Submitted By <%#Variable.SubmittedBy.Email%>". Here, the string "Submitted By" should be typed directly in the text editor. The second half of the subject is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the Field drop down and clicking Append.

The text tool bar above the text editor allows you to manage and format the subject. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

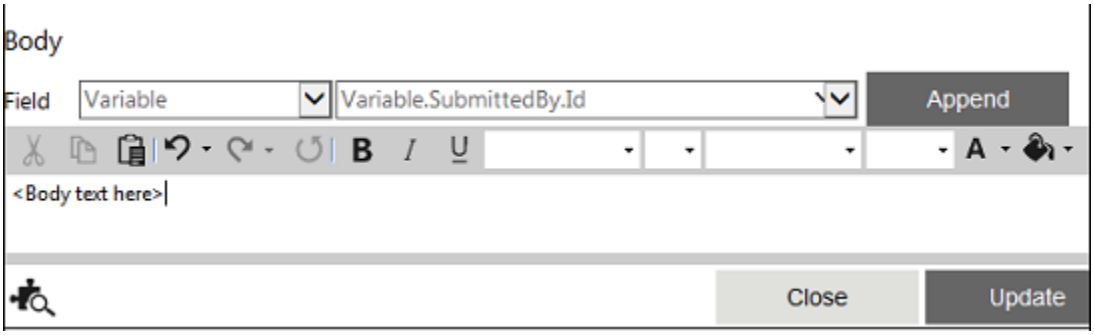
It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text
- At the bottom of the window, you will find the following buttons:
- Update - You can click this button to save the subject.
- Close - You can click this button to close this window.

ResourceSet - Body

Body Property Window

You can specify a message Body for the Resource Set work item assignment in this property window. To open this window, click the button for the 'Body' property.



The window has a text editor to enter the body text. You can do this either by entering a discrete value directly in

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/ drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the body using a combination of discrete value and variable or content. For example, you can specify the body message as "Hi, Please approve the leave application submitted by <%#Variable.SubmittedBy.name%>". Here, the string "Hi, Please approve the leave application submitted by" must be typed directly in the text editor. The second half of the body is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the Field drop down and clicking Append.

The text tool bar above the text editor allows you to manage and format the body text. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

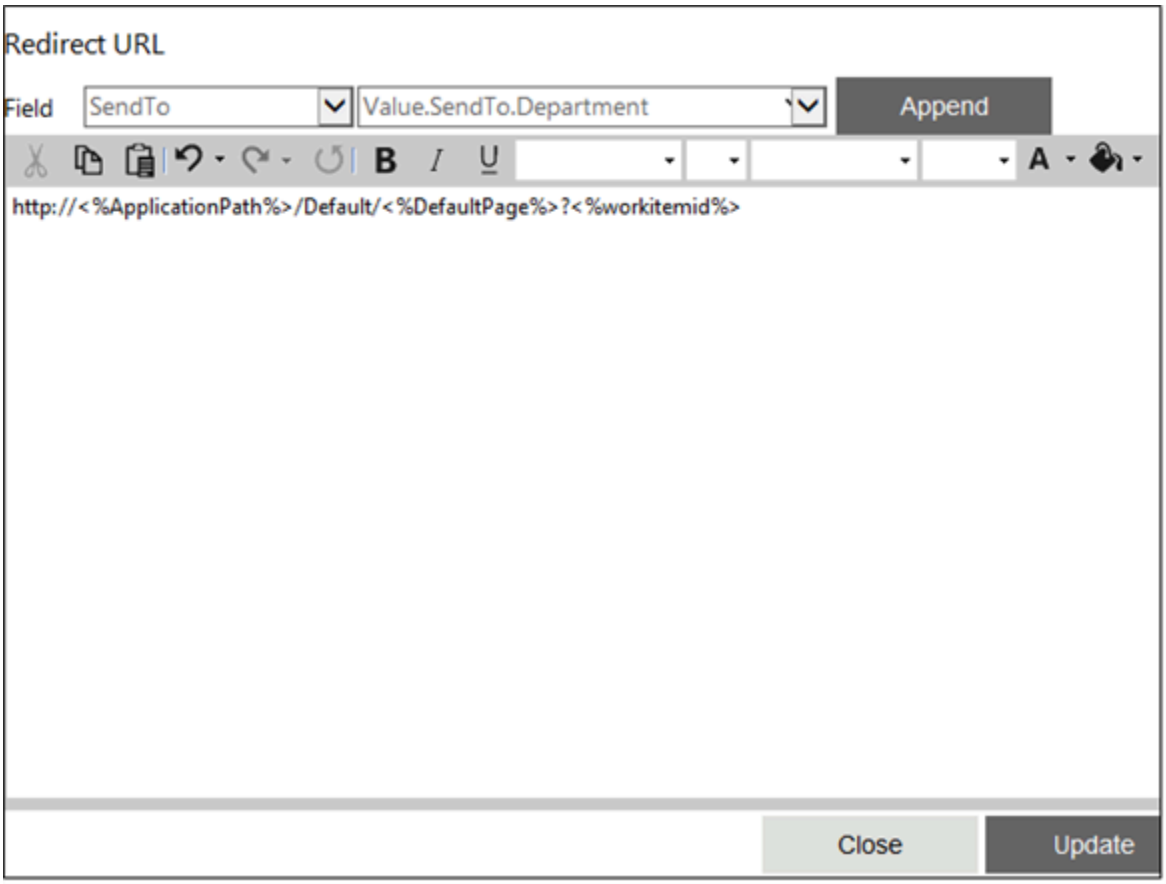
- **Update** - You can click this button to save the body message.
- **Close** - You can click this button to close this window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

ResourceSet - Redirect URL

Redirect URL Property Window

You can specify a message Redirect URL for the Resource Set activity in this property window. To open this window, click the button for the 'Redirect URL' property



The window has a text editor to enter the Redirect URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Redirect URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Redirect URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Redirect URL. It has the following text tools –

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the Redirect URL.
- **Close** - You can click this button to close this window.

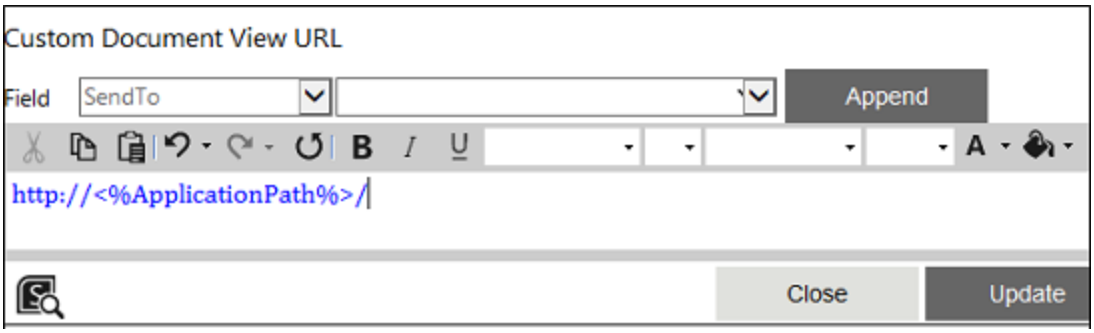
ResourceSet - Custom Document View URL

Custom Document View URL Property Window

You can specify a message Custom Document View URL for the Resource Set activity in this property window. To open this window, click the button for the 'Custom Document View URL' property.

For more details, see Custom Document View URL.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the Custom Document View URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Custom Document View URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Custom Document View URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Custom Document View URL. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Print - to print the text

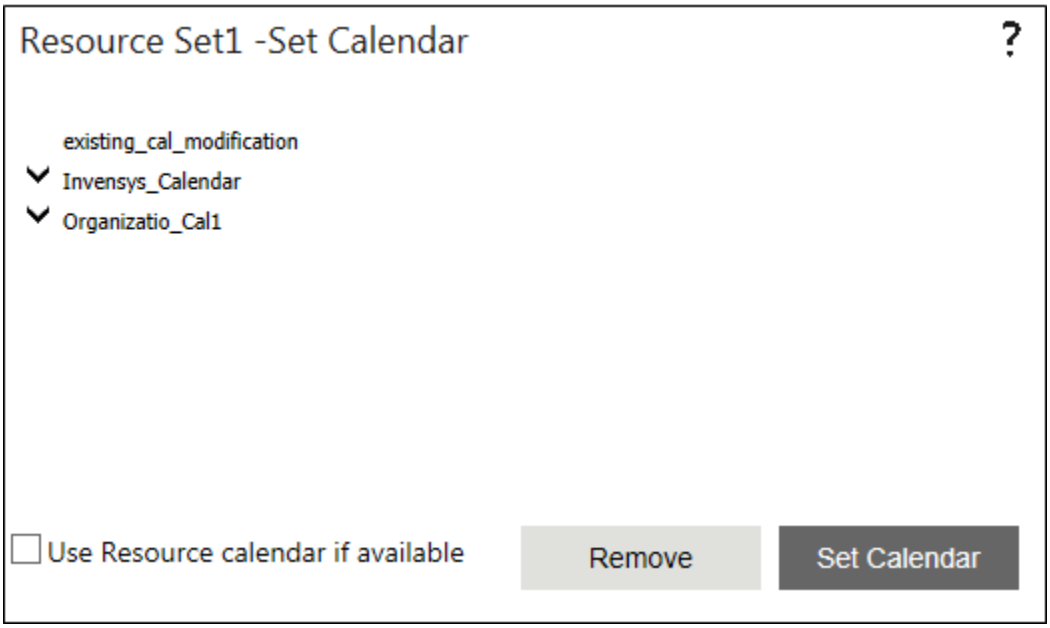
At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the Custom Document View URL.
- **Close** - You can click this button to close this window.

ResourceSet - Set Calendar for Timeout

Set Calendar for Timeout Property Window

You can set a Calendar to calculate the timeout for the ResourceSet activity in this property window. To open this window, click the button for the 'Set Calendar for Timeout' property.



This window lists the available calendars in a tree structure, with lower-level calendars nested within the respective top-level calendar. You can click on a calendar to select it for setting.

There are two buttons in this window.

- **No Calendar** - You can click this button to indicate that there is no calendar set for the activity.
- **Set Calendar** - You can click this button to set the selected calendar.

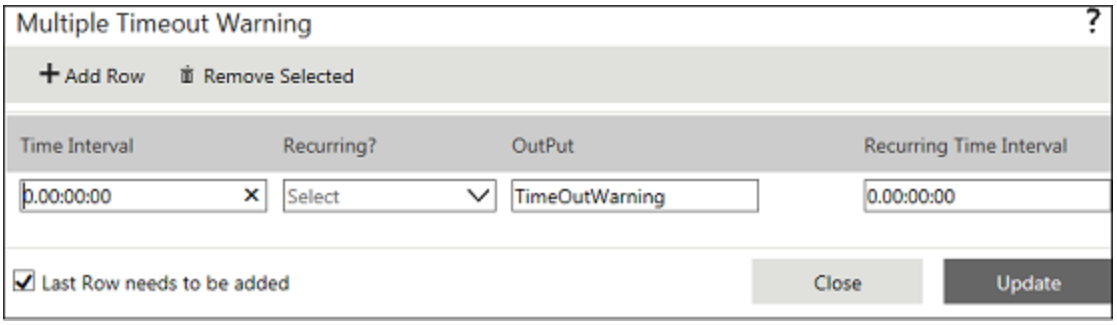
The **User Resource calendar if available** check box can be selected to enable the use of resource calendars if none are displayed in this window for selection.

ResourceSet - Multiple TimeOut Warning

Multiple TimeOut Warning Property Window

You can set the Multiple Timeout Warning in this property window. To open this window, click the button for the 'Multiple TimeOut Warning' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has two tabs at the top to manage the rows of timeout settings in the grid.

1. **Add Row** - You can click this tab to add the newly entered timeout setting and display a new row to the grid where you can add another timeout.
2. **Remove Selected** - You can click this tab to remove the timeout setting row selected in the grid.

The grid has the following parameters that you need to specify for the new timeout setting.

- **Time Interval** - You can use this field to enter the time interval in which you want to generate a timeout warning. You must enter the time in d.hh:mm:ss format.
- **Recurring?** - You can use the drop down in this field to indicate whether the timeout is recurring or not. You can select "Yes" to set the timeout warning to recur.
- **Output** - You can enter the output message corresponding to the timeout warning here.
- **Recurring Time Interval** - You can set the time interval in which the timeout warning has to recur here. Enter the time in d.hh:mm:ss format.

There are two buttons and a check box at the bottom of the screen.

- **Update** - You can click this button to save the newly added timeout settings.
- **Close** - You can click this to close the property window without saving the newly added timeout settings.
- **Last Row needs to be added** - You need to select this check box to save the last row added.

Task Activity

Activity Description

The Task Activity is used in a AVEVA Work Tasks workflow process design in order to assign a task to an actor or set of actors. This activity is used to define the task type and the associated status.

Application Scenarios

A typical application scenario could be a Bug Tracking Workflow Process, in which bug fixing tasks will be assigned to developers. The task can have a drop-down that will function as a task completion maker. From the drop-down, the developer can select the status of the task, selecting "Closed" when the bug is fixed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Workflow Variables for Task

Steps to define variables for use in the Task activity

This section describes how to define variables for use in Task activities. It gives the syntax for creating the variables and typical examples for different cases.

Variables for capturing WorkItemIds

The Activityids for the Task work item can be retrieved using a variable of type array, with the following syntax:

'^'+CurrentActivity.Name+'WorkItemIds'.

For example, **^SolveNetworkIssueWorkItemIds**

Variables for capturing All Actors' Details

The details of the actor(s) to whom the Task work item gets assigned can be accessed by declaring a variable of type array with the following syntax:

'^'+CurrentActivity.Name+'ActingUsers'

For example, **^SolveNetworkIssueActingUsers.**

Variables for capturing Responding Actor Details

The details of the actor submitting the Task response can be accessed by declaring a variable of type actor with the following syntax:

'^'+HWSActivity.ActivityName+'RespondingUser'

For example, **^SolveNetworkIssueRespondingUser**.

This variable stores the responded user information. If it is more than one user, it will retrieve the last user details.

Note: For single user, you can declare this variable as Resource type. If task is assigned to more than one user and to get all responded users information, then declare this variable as array.

Variables for capturing Comments

Comments entered in a Task can be retrieved using a variable with the following syntax:

'^'+_HWSActivity.ActivityName+'CommentTaskStatus'.

For example, if a comment entered along with a task response submission has to be retrieved from an activity named 'SolveNetworkIssue', add a variable with ^SolveNetworkIssueCommentTaskStatus as its name. The variable gets initialized with the comment entered by the resource after the activity submission.

Variables for assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the Start activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

To assign multiple queues, specify the queue-ids with semicolon (;) separator. Skelta.HWS.Queue.QueueCollection class can be used to retrieve queue information.

Variable for Queue Acting Users

You can declare an array type variable to get updated information about the name of the actor to whom an activity gets assigned from a queue.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

In the Workflow Variables, define the variable in the following format:

^+CurrentActivityName+'QueueActingUsers'

Ex: **^HRFormQueueActingUsers**

HRForm is the activity name.

For example, the value of the user who acts on the HRForm activity will be stored in the variable. The Approval1 task will be assigned to the same user.

Variables for storing timeout warning value

Suppose, you want to store the timeout warning value for a task activity in your workflow. The naming convention for the system variable to be declared is

^<ActivityName>ResponseByWarningTimeout

The value returned by the variable will be in UTC format. For multiple user scenarios the highest timeout value calculated for a user will be available in the variable.

Activity Properties

The Task activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right-click menu for the Task activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Display Name - This property specifies the name to be displayed for the activity in the Work Item List.

Property Type: Optional (This property need not be set if the display name is already configured in the Actions.xml. The name entered here overrides any display name that is configured for this activity in the Actions.xml file.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Delivery Channels - This property is used to specify the delivery channels to deliver this activity to the assigned actor(s). Multiple channels can be specified giving the actor(s) different options to act on the work item.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to set the Delivery Channels property

1. To select the Delivery Channels, click the button for the 'Delivery Channels' property to open the property window.
2. In this window, the Delivery Channels that have been configured are listed on the left along with a check box against each channel. Click and select the check box corresponding to the channel you wish to select. You can select multiple channels by selecting the respective check-box.
3. Click the **Save Channels** button.

See Task - Delivery Channels for a detailed description of the Delivery Channels property window.

Select Task Type - You can define Task Type and Status under Select Task Type in a Task Assignment. The Status added under Task Types is used to maintain the status of the Task assigned.

Property Type: Mandatory (This property must be set for the activity to be used in the workflow.)

Attributes of the select Task Type property

- The user interface displays the Task Type and Status under them.
- Task Completion Maker functionality.
  - There can be a Task Completion Status associated with a Task Type. A Task Status marked as ! is said to be the Task Completion Status for the Task Activity.
  - The outcome of the Task Assigned is determined by the Task Completion Status tagged with Task Type.
  - Whenever a actor acts upon (submits) the Task, the engine is notified with the Task Completion Status of the Task, provided the status is marked as Task Completion status. For example, if the TASK COMPLETION Task Type has COMPLETED and ON-HOLD as status, and COMPLETED is tagged with the Task Completion Maker, then, when the Task is submitted with COMPLETED as the selected status, the engine is alerted about the 'COMPLETED' status and it executes the next process in the workflow.
- Edit and Delete options have been provided.
- First the actor clicks the Add New button. (The option to add more than one Task Type is also provided.)
- Then the actor has to enter the Task Type and the Number of Status to be added under the Task Type.
- For the Task Type, the actor has to enter the Status.

Steps to set the Select Task Type property

1. To add Task Types and their Status, click the button for the 'Select Task Type' property to open the property window.
2. In this window, specify the number of task types to add in the **No. of additional Task Types** text box and click Add New. This will open a new window.
3. In the second window, enter the task type name in the text box under the **Task Type** column.
4. Specify the number of status options for the new task type in the **No. of Options** text box.
5. Repeat steps 3 and 4, if you are adding more than one task type for the task.
6. After setting the parameters for all the task types, click **Next** to go to the next window.
7. In the third window, specify the status options that will be presented to the actor to set the status for the task, in the text boxes under the **Task Type & Status** column. The number of rows displayed for specifying the status options correspond to the number specified in step 4.
8. Specify the status option that will set the task status as completed by checking the corresponding check box

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

under the **Task Completion Maker!** column.

9. You can delete the status if required in the **Delete?** column.
10. After completing all the details you can click on the **Finish** button to save the new task types.

The first window is displayed again when you click **Finish**. It shows the newly added task type(s). You can modify or delete the new task types or add more new task types. If you do not wish to perform any more operations, click **Close** to exit this window.

See Task - Task Type for a detailed description of the Select Task Type property windows.

Work Item Fields - This property is used to add the custom fields to be used to display application data as well as value in variables and content in the Work items list when this activity is executed.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Work items fields property

1. To select the Work Item Fields, click the button for the 'Work Item Fields' property to open the property window.
2. In this window, the Work Item Fields that have been set are listed in a grid. Click a row to select the corresponding Work Item Field.
3. Right-click the row, and select "SetWorkItemField" from the context menu to set it.

See Task - Work Item Fields for a detailed description of the Work Item Fields property window.

Actors

The properties in the Actors group have been described below. You can use these properties to specify the actors who will work on the activity and related properties.

Assign Actor(s) - This property specifies the actor(s) who has to take up the Task activity. It is tagged with the Ignore Unavailable Resources property which in turn is tagged with Alternate Resource Allowed. This will be explained in detail later.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any actor(s).)

Steps to set the Assign Actor's property

1. To specify the actors, click the button for the 'Assign Actor(s)' property to open the property window.
2. In this window, build an expression to select the actors using the actors properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking Undo. You can also clear an existing expression if necessary by clicking Clear. This is useful if you want to clear large expressions in the text area in order to build a new expression.
5. After building the expression, click **Update** to save it.

See Task - Assign Actor(s) for a detailed description of the Assign Actor(s) property window.

See the Using XML Variables section in the About XML Variables topic for details on how to use the XMLVariables to build the expression.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Assign Queue(s) - This property specifies the queue(s) to which the Task activity is to be assigned.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any Queue(s).)

Steps to set the Assign Queue(s) property

1. To specify the queue(s), click the button for the 'Assign Queue(s)' property to open the property window.
2. In this window, select the queue(s) from the Available Queues list box. You can select multiple queues by selecting with the
3. After selecting the required queue(s) click the **[>]** button to move it to the Selected Queues list box. You can click the [>>] button to select all the queue(s) and move them to the Selected Queues list box.
4. You can remove any queue from the selected list by selecting it and clicking the [<] button. You can remove all the queues by clicking the [<<] button.
5. After adding the required queue(s), click **Assign** to assign the activity to the selection.
6. Click **Close** to close the window without saving the selection.

See Task - Assign Queue(s) for a detailed description of the Assign Queue(s) property window.

Min. Ownerships [% or Number] - This property, which can be specified as a number or percentage, is the minimum number of actors who should take ownership of the Task assignment. This property is related to the 'Timeout warning for Ownership' and 'Timeout for Ownership' properties. To illustrate, when 'Timeout for Ownership' is reached, the activity first checks if the ownership count is less than Min.Ownerships (if it is set). If it is, the work item will be removed from all Work Item lists.

If the 'Min.Ownerships' has already been satisfied, then the work item will be removed from the Work Item lists of those who have not taken ownership. If the Min.Ownerships value is 0, then the activity checks if the ownership count is less than Max.Ownerships (if it is set). If it is, the work item will be removed from all Work Item lists. Else, the work item will be removed from the Work Item lists of those who have not taken ownership. If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work Item lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Max. Ownerships [% or Number] - This property, which can be specified as a number or percentage, is the maximum number of actors who should take ownership of the Task assignment. If 'Timeout for Ownership' is specified and achieved, and the 'Min.Ownerships' is also satisfied, then the activity waits for action from those who have taken ownership. If the Min.Ownerships value is set to 0, then it checks if the ownership count is less than Max.Ownerships. If it is, the work item will be removed from all Work Item lists. Else the work item will be removed from the Work Item list of those who have not taken ownership. If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work Item lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Enable Take Ownership? - If this property is set to Yes, then the actor is provided with a Take Ownership option in the Work items list to take ownership of the task. If you do not required the Take Ownership option in the Work items list, then set this property as No.

Note: By default, this property is set as Yes.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to set the Enable Take Ownership? property

1. Select **Yes** or **No** option from the Enable Take Ownership? drop-down list.

Re-Assign Allowed? - If this property is set then the actor is provided with a Re-Assign option. Using this, the actor can forward the work item to another actor.

Property Type: Optional (This property needs to be set only if necessary.)

Ignore Unavailable Actors?

This property provides the following options - **Yes**, **Yes** - **holidays excluded** and **No**. If this property is set to 'No', when the Approval activity is assigned to the actor's Work Item list, a check is first done to see whether the actor is available. If the actor is not available then a check is made for an alternate actor assigned to the unavailable actor. Simultaneously, the value of the property 'Alternate Actor Allowed' is checked. If that is also set, then the Approval work item is assigned to this actor. If this property is set to Yes - holidays excluded, the workflow engine will ignore the fact that actors are not available if the day is an organization holiday and the output Not Enough Resources will not be generated.

Property Type: Optional (This property needs to be set only if necessary.)

Alternate Actor Allowed? - This property is set to assign a particular task to an alternate actor in the absence or unavailability of any actor.

Property Type: Optional (This property needs to be set only if necessary.)

Notification Contents

The properties in the Notification Contents group have been described below. You can use these properties to specify the content used to notify actors about the activity.

Subject - This is the subject of the Task assignment. The subject can be customized for each actor.

Property Type: Optional (This property need not be set. If it is not set, the default subject line from the Actions.xml file is displayed.)

Steps to set the Subject property

1. To specify the subject, click the button for the 'Subject' property to open the property window.
2. In this window, enter the subject text directly in the text editor.
3. If you want to customize the subject you can do so using the Field controls to add resource property values. For example, to add a subject that includes the email id of the actor submitting the Task, you can do as follows: Type in the text "Submitted By". Select "Variable" and "Variable.SubmittedBy.Email" from the two drop downs for Field and click Append. This will add the placeholder for the variable after the text entered in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the email id.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the subject, click Update to save it.
7. Click Close to close the window without saving the subject.

See Task - Subject for a detailed description of the Subject property window.

Body - This is the actual message of the Task sent to the actor(s). The message can be customized for each actor. There is an **Enterprise Console Explorer(plug-in)** icon in the Body property window to get the work item details

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

URL.

See Enterprise Console Control (EC Plug-in) for more information.

Property Type: Optional (This property need not be set.)

Steps to set the Body property

1. To specify the message body, click the button for the 'Body' property to open the property window.
2. In this window, enter the body text directly in the text editor.
3. If you want to customize the body you can do so using the **Field** controls to add resource property values. For example, to add a body that includes the name of the actor submitting the Task, you can do as follows: Type in the text "Hi, Please act on the task assigned to you by". Select "Variable" and "Variable.SubmittedBy.name" from the two drop downs for **Field** and click **Append**. This will add the placeholder for the variable after the text entered in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the name.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the body, click **Update**.
7. Click **Close** to close the window without saving the body

See Task - Body for a detailed description of the Body property window.

From Email Address? - This property is used to specify the email address from which notification messages will be sent for the activity.

Property Type: Optional (This property needs to be set only if necessary. If it is not set, the default value in Activities.xml will be used.)

Hide Responses? - If this property is set to 'No', then actors can see the responses of other actors in a group Task scenario. To view these responses i.e., ownerships, Task status etc., select the 'Take Ownership' option. This will open a new pop up with the responses of other actors who have taken ownership.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Priority - This property is set to indicate the priority of the Task assignment. Enter the values 0-33 to set as Low, 34-66 for Medium, or 67-99 for High. The default value is set to 34.

Property Type: Optional (This property needs to be set only if necessary.)

Show Custom Document View Window - If this property is set to 'Yes', then the Custom Document View Window will be shown to actors in their Work Item Lists.

Property Type: Optional (This property needs to be set only if necessary.)

Send Notification Email - If this property is set to 'Yes', then notification emails will be sent to the actors. If it is set to 'Yes-HTML', then the emails will be sent in HTML format.

Property Type: Optional (This property needs to be set only if necessary.)

Notify if alternate actor is considered - If this property is set to 'Yes', then notification emails will be sent to the actors when an alternate actor is considered.

Property Type: Optional (This property needs to be set only if necessary.)

Redirect URL - This property is used to get the URL of the web page to redirect the client to, in the case of

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

synchronous execution of the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Redirect URL property

1. To specify the Redirect URL, click the button for the 'Redirect URL' property to open the property window.
2. In this window, enter the Redirect URL directly in the text editor.
3. If you want to customize the Redirect URL you can do so using the Field controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for Field, and clicking Append. This will add the placeholder for the variable in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the URL.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Redirect URL, click Update to save it.
7. Click Close to close the window without saving the Redirect URL.

See Task - Redirect URL for a detailed description of the Redirect URL property window.

Custom Document View URL - This property is used to get the URL of the page to which the custom document dlls does a post. The data posted to the page are application, workflow, executionid, executiondetailsid, and workitemid. The posted data can be retrieved as Request.Form["workitemid"].

Property Type: Optional (This property needs to be set only if necessary.)

See Task - Custom Document View URL for a detailed description of the Custom Document View URL property window.

Steps to set the Custom Document View URL property

1. To specify the Custom Document View URL, click the button for the 'Custom Document View URL' property to open the property window.
2. In this window, enter the Custom Document View URL directly in the text editor. If you want to customize the Custom Document View URL you can do so using the Field controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for Field, and clicking Append. This will add the placeholder for the variable in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the URL.
3. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
4. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
5. After adding the Custom Document View URL, click Update to save it.
6. Click Close to close the window without saving the Custom Document View URL.

Mail Channel

The properties in the Mail Channel group have been described below. You can use these properties to customize

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the Task activity(activities) with different mail channel templates.

Use html template from property - If this property is set to **'No'**, then it will take the mail channel template from the physical path '[AVEVA Work Tasks Installed Path]\BPM.NET\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\TaskEmail.html.
If this property is set to '**Yes'** and the **"HTML Template"** property is empty, then it will take the mail channel template from the physical path '[AVEVA Work Tasks Installed Path]\BPM.NET\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\TaskEmail.html.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the HTML templates

1. To specify the HTML template, click the edit icon to open the property window.
2. In the **HTML template** window, provide the template which should be in html format. If you are pasting a copied html content, then right-click the **HTML template** window and paste it as "**Paste Plain Text**".
3. Click **Update**

See Task - HTML Template for a detailed description of the property HTML template window.

Attachment

This property is used to configure report to be sent as an Email attachment.

Property Type: Optional (This property needs to be set only if necessary.)

See Task - Attach Reports for a detailed description of this property.

Decision Influencers

The properties in the Decision Influencers group have been described below. You can use these properties to specify some parameters that influence the outcome of the activity.
Max Actors Required for TaskAssignment Completion [% or Number] - This property is used to check what percentage or number of actors should take action on the Task. Decision on the Task is taken only after the specified number or percentage of actor(s) have taken the action. The activity waits for the actors' action until the 'Timeout for action' is reached.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Min Actors Required for TaskAssignment Completion [% or Number] - This property specifies the minimum number of actors to act on the Task before the 'Timeout Warning for Action'. If the action limit is reached before the 'Timeout Warning for Activity' then the warning message is not shown.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Escalation & Timeout

The properties in the Escalation & Timeout group have been described below. You can use these properties to specify escalation and timeout settings for the activity.

Set Calendar for Timeout - This property is used to specify the calendar to be used to determine activity timeouts.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to Set Calendar for Timeout

1. To specify the calendar for timeout, click the button for the 'Set Calendar for Timeout' property to open the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

property window.

2. In this window, select the required calendar by clicking the calendar name in the tree structure.
3. Click **Set Calendar** to set the selected calendar.
4. You can specify that a resource calendar should be used, if available, by clicking to select the corresponding check box.

See Task - Set Calendar for Timeout for a detailed description of the Set Calendar for Timeout property window.

Minimum available time for Activity - This property is used to check the availability of an actor to whom a task must be assigned while executing the activity. The availability check for the actor is done based on the Calendar settings (Global or Resource Calendar). If the time taken to complete the task is within the value set for the **Minimum available time for Activity** property, then the task will be assigned to that actor. However, if the time taken to complete the task exceeds the value set for the **Minimum available time for Activity** property, then the task will be assigned based on the values set for the **Ignore Unavailable Actors?** and **Alternate Actor Allowed?** properties.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout Warning for Ownership - This property specifies the time at which a warning is provided to an actor to take ownership of the work item assigned to him/her. If the actor does not take ownership within this time the engine is notified with a warning. Thus if 'Min.Ownerships' is set, and it is not satisfied before the given time then a warning is sent. If Min.Ownerships is 0, it will check for Max.Ownerships. If the ownerships count does not reach the Max.Ownerships value within the set time, a warning is sent. If both Min.Ownerships and Max.Ownerships are not set, this property is ignored.

Property Type: Optional (This property needs to be set only if necessary.)

Understand how Ownership Properties relate to each other:

- Task activity is assigned to the actors in the Task assignment list by the workflow engine. Task activity has a default property set that at least one actor has to make Take Ownership of the Task activity.
- A drop-down interface has been provided for the actor to Take Ownership. However, the actor can act on the Task without taking ownership first, in which case ownership will be considered to be done. i.e. if the actor acts upon the Task activity without taking ownership, then it is considered as an ownership with action taken.
- After the actor has taken ownership, a link is provided to the actor, which opens a pop up with the details of the other actors. This is again tagged with the 'Hide Responses?' property. The link will be provided only If 'Hide Responses?' is set as 'No'. The pop up will have details like Actor name, Ownership Status, Action taken, Comments.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] then the 'Timeout Warning For Ownership' property is not considered.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] and the 'Time Out For Ownership' is also reached then the work item is removed from other actors' Work Item list and the work item waits for action from those who have taken ownership.
- If the 'Min.Ownerships' limit is not reached for the Task activity, and the 'Timeout Warning For Ownership' has been exceeded then the actor is notified with a warning by including an email or an information activity, with the Timeout Warning for Ownership link to it.
- If 'Timeout For Ownership' is set, if the Task activity has not reached the minimum limit (Min.Ownerships) and the 'Time Out For Ownership' has been reached, then the work item is removed from the Work Item lists of all the actors.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- If 'Min.Ownerships' = 0 and if 'Max.Ownerships' is set, the activity waits for actors to take ownership or action until the 'Time Out For Ownership' is reached (if this is set).
- If none of the Ownership properties are set then the activity waits until the actor takes ownership or acts upon it.

Timeout for Ownership - This property specifies the time until which actors are allowed to take ownership, i.e. the time until which the Task activity will wait for actors to take ownership. If Min.Ownerships is set and is not reached, the work item is removed from all Work Item lists. If Min.Ownerships is 0, it will check if ownership count is less than Max.Ownerships. If it is, the work item is removed from all Work Item lists. Else the work item will be removed from the Work Item lists of those who have not taken ownership. The work item will remain in the Work Item list of actors who have taken ownership and wait for action from them.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout Warning for Activity - This property specifies when a reminder has to be provided for the actors to act on the Task request, if the action count has not been reached. If properties such as Max.Ownerships, Affirmative Action for Task (AAA), and Action Limit (AL) are set, once the property values are reached then a warning is sent to the participant. When the timeout is reached, the following checks are made: if AL is set, when the action count i.e. the no. of Tasks or rejections (AC) < AL, then a warning is sent, if AL = 0 and AAA is set, and when AC < AAA, then a warning is sent to the participant.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Timeout for Activity - This property's value is used to decide until what time the activity should wait for the actors' action. If properties such as Max.Ownerships, AAA, AL are set, once the property values are reached the work item is removed from the Work Item lists of all the actors. When the timeout is reached, the following checks are made: If AL is set, when the action count i.e. the no. of Tasks or rejections (AC) < AL, then the timeout for action is performed and the work item is removed from all Work Item lists. If AC > AL, it checks with the no. of Tasks or 'decision count' (DC) and AAA. If DC >= AAA (if it is set), the action is considered completed. If AAA = 0 and if DC >= AL, the action is considered completed. If AL = 0 and AC < AAA, then the timeout for action is performed and the activity is removed from all Work Item lists. If AL = 0 and AC > AAA, it will check with DC. If DC > AAA it is completed. If AL = 0 and AAA = 0, then if DC is at least 1, then it is considered completed.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Multiple Timeout Warning - This is used to set recurring timeout warnings that will be sent to the actors who have to take action. To set multiple timeout warnings, click the 'Multiple TimeOut Warning' button in the Properties pane for the Task activity. A new window pops up. In this window, set the interval for the recurring warning as well as the output message. You can also set the Recurring Time Interval. This can be used to set a different recurring time out after the first timeout happens. For example, if 'Time Interval' for the property is set as 7 days and 'Recurring Time Interval' is set as 2 days, then the first time out will happen after 7 days and there will be a recurring time out every second day after the seventh day (first) time out.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Steps to set the Multiple Timeout Warning property

1. To set the multiple timeout warning, click the button for the 'Multiple Timeout Warning' property to open the property window.
2. Enter the time interval for the timeout in the Time Interval field.
3. Select "Yes" from the drop down for the Recurring? field if you want the timeout warning to recur. Select "No" otherwise.
4. Enter the output message corresponding to the timeout warning in the Output field.
5. Enter the time interval for the recurring timeout in the Recurring Time Interval field.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. After entering the content details, click Add Row to add this timeout setting and display a new row for entering another timeout setting.
7. Remove rows if necessary by selecting them and clicking Remove Selected.
8. After specifying the timeout settings, click Update.

See Task - Multiple Timeout Warning for a detailed description of the Multiple TimeOut Warning property window.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Miscellaneous

Redirection Enabled? - This property enables redirection. When this property is set to Yes, the workflow redirects the user to the URL in the Redirect URL property in the next activity.

Activity Outputs:

The Task Activity returns the following values -

Action Completed: Displayed when all the actors have taken part.

Action Completed - Affirmative Percentage Not Reached: Displayed when all the actors have not taken part.

Not Enough Resources to Take Acknowledge: Displayed when the filter condition did not yield enough actors to whom the request can be sent.

Timeout - Take Acknowledge: Displayed when the specified minimum number of actors did not take ownership of the request in the given time.

Timeout - Action: Displayed when some or all of the specified actors did not perform the required action in the given time.

Timeout Warning - Take Acknowledge: Displayed when a warning is generated that the specified minimum actors have not taken ownership of the request in the given time.

Timeout Warning - Action: Displayed when a warning for timeout is generated because some or all the actors did not perform the required action in the given time.

Note: The following outputs will return only if user assigned tasks to a queue user.

Resource Unavailable for Queue - If a queue is assigned to a task & if there are no resources available in the queue then this output will get triggered.

TimeOut for Queue Task - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting the completion of that task.

Timeout Warning -Queue - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting in sending a warning out put. However, the task will still be sleeping.

Timeout Warning -Queue Participant - This output will get triggered when the task in the queue is picked by user but still not acted on that task within the queue escalation time out settings resulting in triggering a warning output. However, the task will still be sleeping.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Task - Delivery Channels

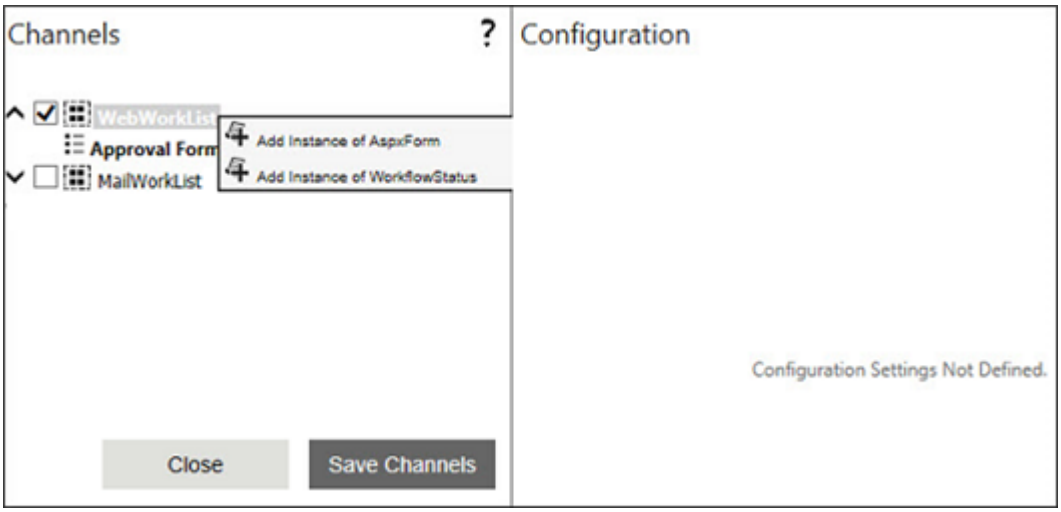
Purpose

The purpose of this chapter is to describe the feature of WorkflowStatus Form of Delivery Channels property under the category Configuration of any human activity available in AVEVA Work Tasks.

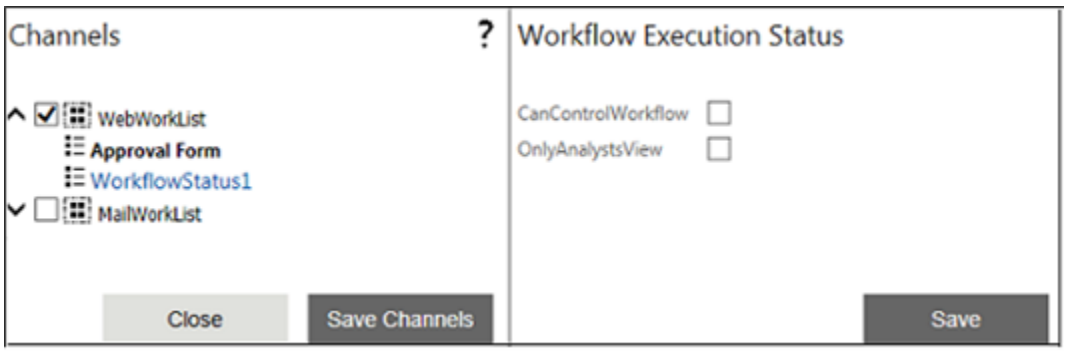
Description: This feature enables us to view the status of the workflow in the work item list in the graphical form. [Optional]

1. Select the Human Activity.
2. Click the **Delivery Channels Property**.

   This will open a window like the one shown below.



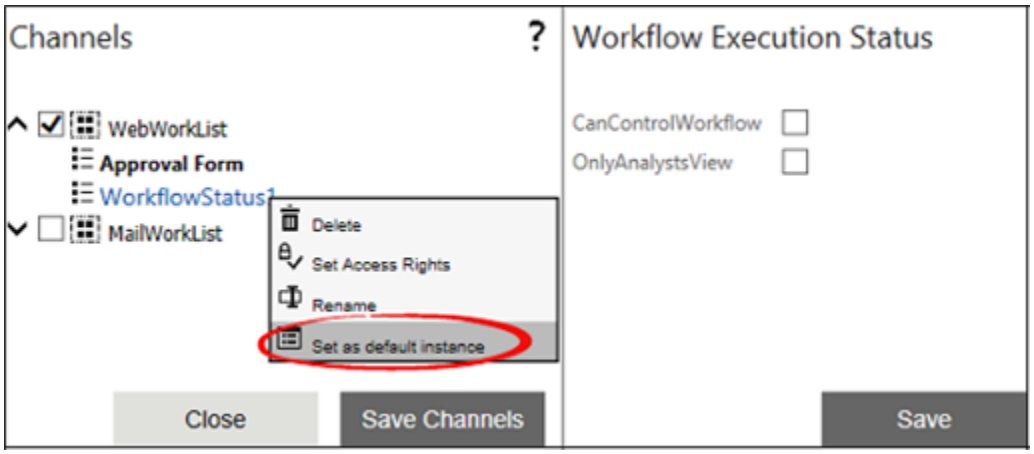
3. Right-click **WebWorkList** to display the context menu with the **Add Instance of Workflowstatus** option as shown below.



4. Set the Properties for the Execution View Status Graphical view.
5. Click **Save**.
6. If you want to Set the workflowstatus form as the default instance, right-click **WorkflowStatus1**, and

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

select the **Set as default instance** option from the context menu.



7. Click **Save** and **Save Channels**.

Output

Now, when you click any row in the Work items list, you will find this Execution view page being displayed as the detailed view if you set it as a default instance.

Otherwise, workflow status form will appear in the context menu of each row in the Work items list and you can open the execution view by selecting it from the context menu

Task - Task Type

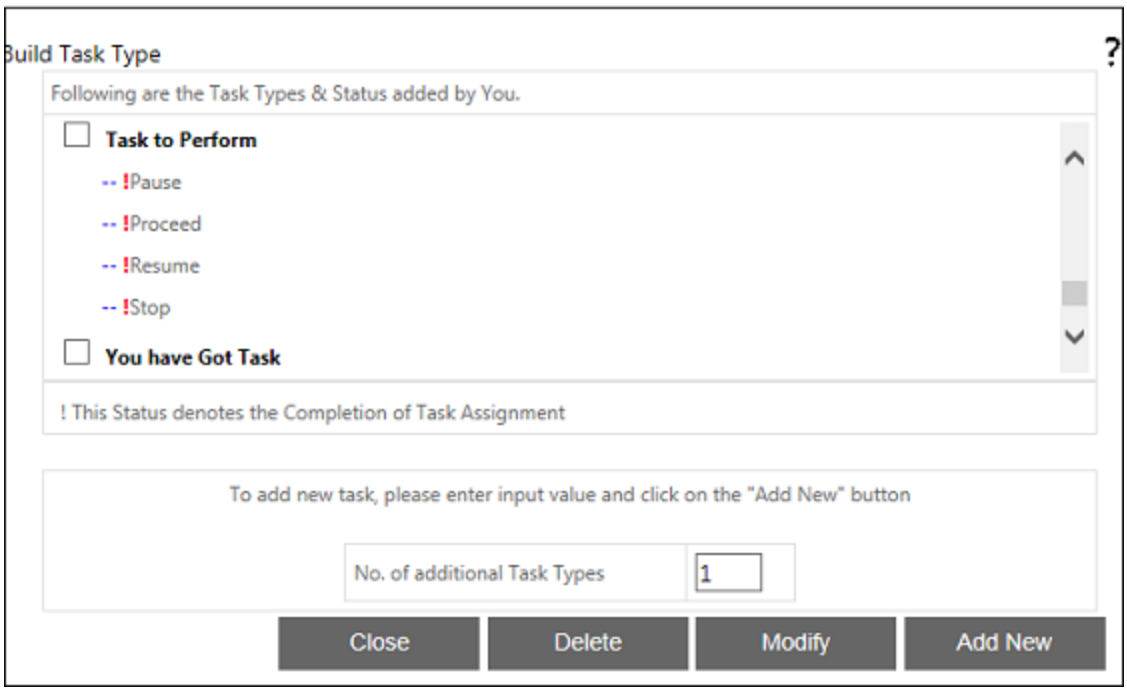
Task Type Property Window

You can add the task types and the corresponding status in this property window. To open this window, click the button for the 'Select Task Type' property.

The window shown in the image below is used to Add, Edit and View the Task Types added for the Task activity. This is the first window in the Select Task Type property user-interface. It lists all the tasks that have been added for the task. Each task in the listing has a check box. You can use this to select the task for editing. The window has the **No. of additional Task Types** field to specify the number of questions to add. This window has the following buttons:

- **Add New** - You can click this button to add new task types. This will open the second window in the Select Task Type user-interface where you can add the tasks. The number of questions you can add will depend on the value set in the No. of additional Task Types field.
- **Modify** - You can click this button to edit the selected task type(s). You can select a task type for editing using the check box next to it.
- **Delete** - You can delete the selected task type(s) by selecting them as described above and clicking this button.
- **Close** - You can click this button to close the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window shown below is the second in the user-interface for adding task types for a task. This interface is displayed after the actor selects **Add New** or **Modify** from the first window shown above. You can add as many task types as were specified the first window and a separate row of fields is displayed for every new task type. For each task type, the following parameters can be set here:

- **Task Type** - You can enter the name of the task type in this text box.
- **No. of Status** - You can specify the number of status options for the Task Type in this text box.

This window has the following buttons:

- **Back** - You can click this to go back to the first window.
- **Next** - You can click this to go to the next window.
- **Close** - You can click this to close the window.

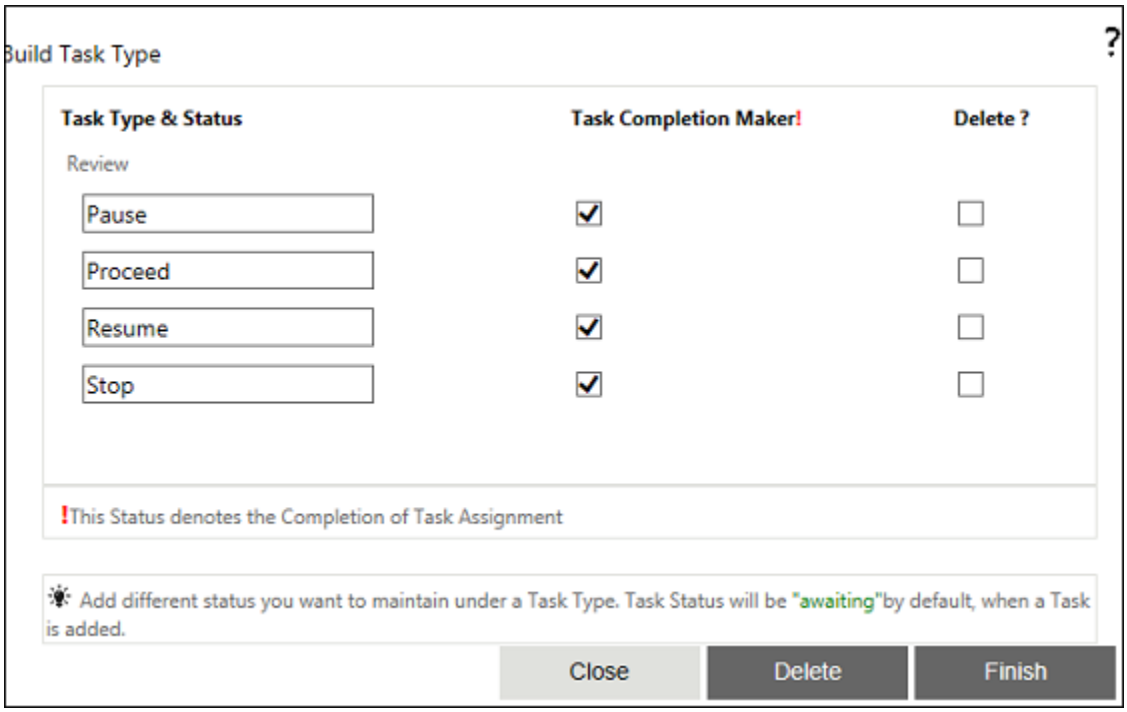
The window shown below is the third in the user-interface for adding task types for a task. In this window, you can set the following parameters:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Task Type & Status** - You can specify the status options that will be available to the actor to set for the task in the fields in this column.
- **Task Completion Maker!** - You can specify the status option that will set the task status as completed.
- **Delete?** - You can delete the status from this column.

This window has the following buttons:

- **Back** - You can click this to go back to the first window.
- **Finish** - You can click this button after completing all the details to save the question.
- **Close** - You can click this button to close this window.



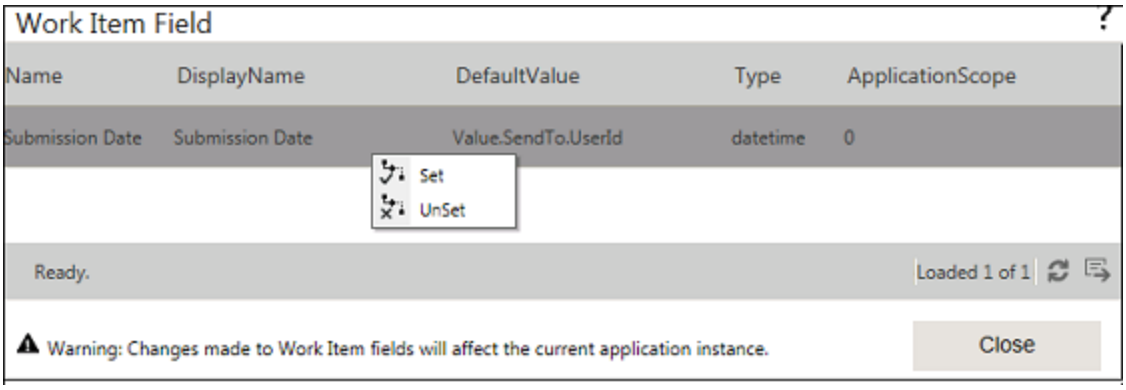
The first window is displayed again when you click Finish in the previous window. It shows the newly added tasks along with their status. You can modify or delete the new task or add a new task. If you do not wish to perform any more operations, click **Close** to exit this user interface.

Task - Work Item Fields

Work Item Fields Property Window

You can set the Work Item Fields for this activity in this property window. To open this window, click the button for the 'Work Item Fields' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a grid listing all the Work Item Fields that have been set for the process.

- Each row in the grid represents a Work Item Field. You can click a Work Item Field to select it.
- The context menu has the Set WorkItemField option, which you can use to set the selected Work Item Field for this activity. It also has the UnSet WorkItemField option to remove the Work Item Field setting.
- **Close** - You can use this button to close the window.

  Note: While entering a value in the Work Item field, it is recommended to keep it less than the specified database column size. If there is a need to increase the size of the Work Item field, the user should manually increase the column size in the database.

Task - Assign Actor(s)

Assign Actor(s) Property Window

You can set the actors to whom the Task work item is to be sent in this property window. To open this window, click the button for the 'Assign Actor(s)' property.

See Assigning a Task to a Manager for more details.

The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Different Methods of Assigning an Actor

- **Assigning an Actor by building an expression:**

Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

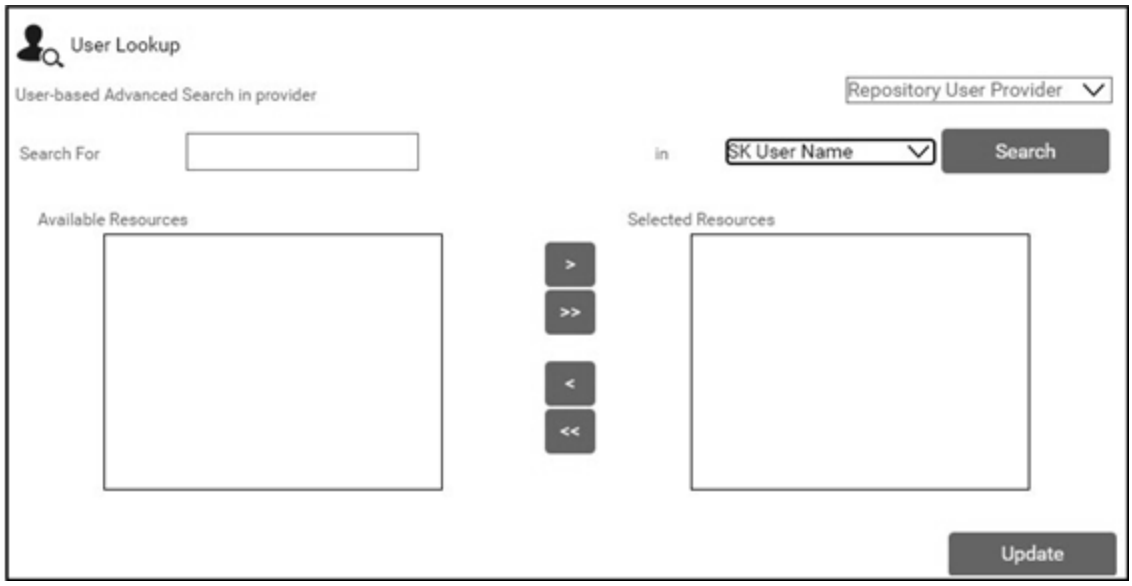
- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

- **Assigning an Actor through User Lookup**

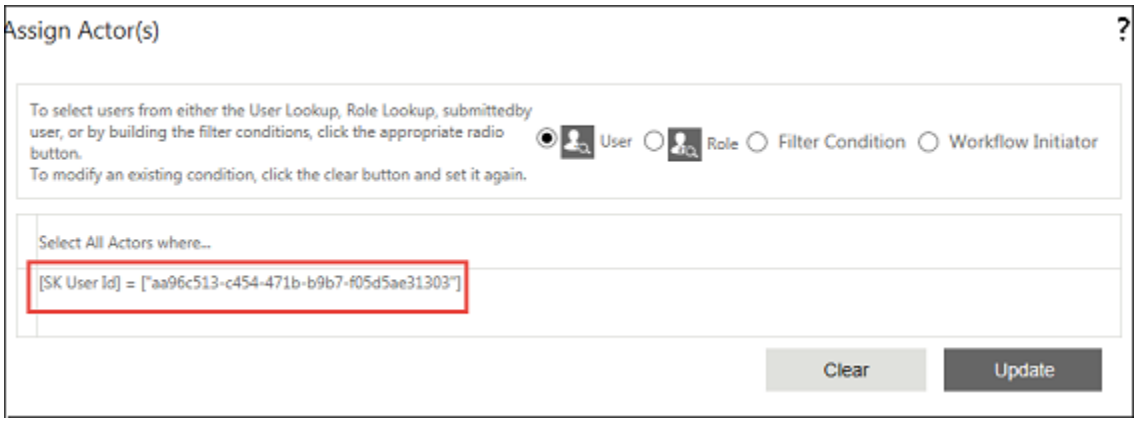
Through the User Lookup you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.
2. The **User Lookup** dialog is displayed, search and select the required actor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:



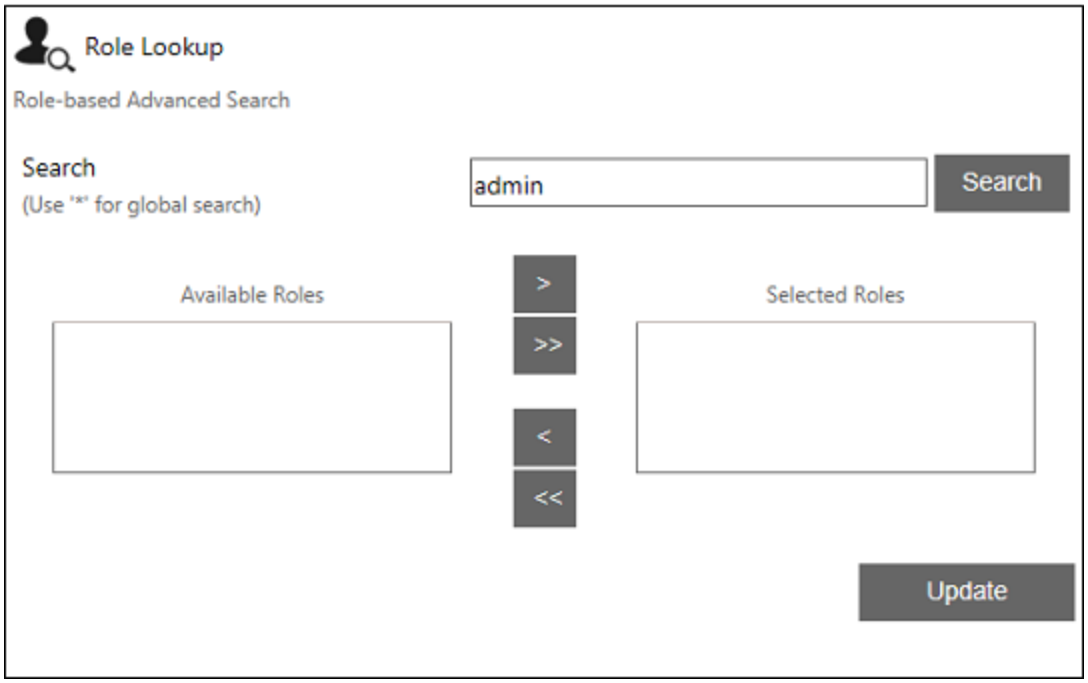
4. Click **Update** in the Assign Actor(s) window to update the selected actor(s) in the Assign Actor property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

- **Assigning an Actor through Role Lookup**

Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

1. Select the **Role Lookup** option.
2. The **Role Lookup** dialog is displayed, search and select for the required role.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



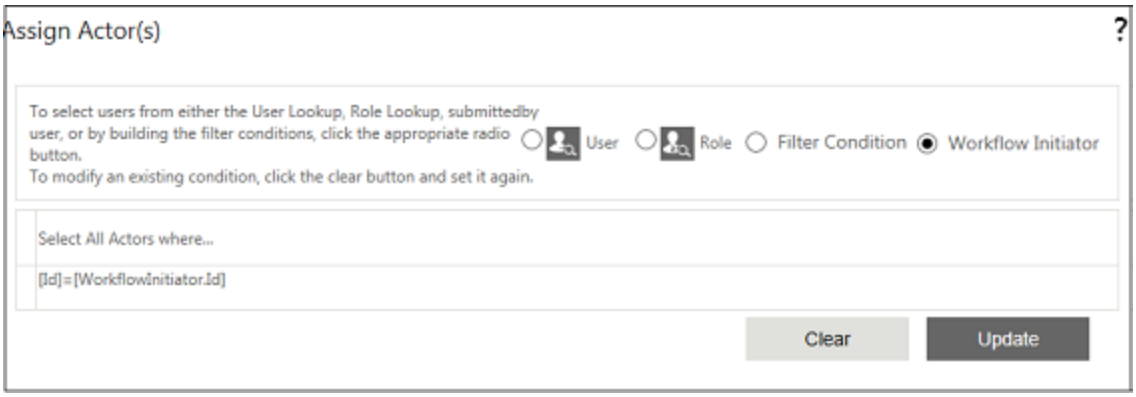
3. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

4. Click **Update** in the Assign Actor(s) window to update the **Assign Actor** property.

- **Assigning an Actor through Workflow Initiator**

Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

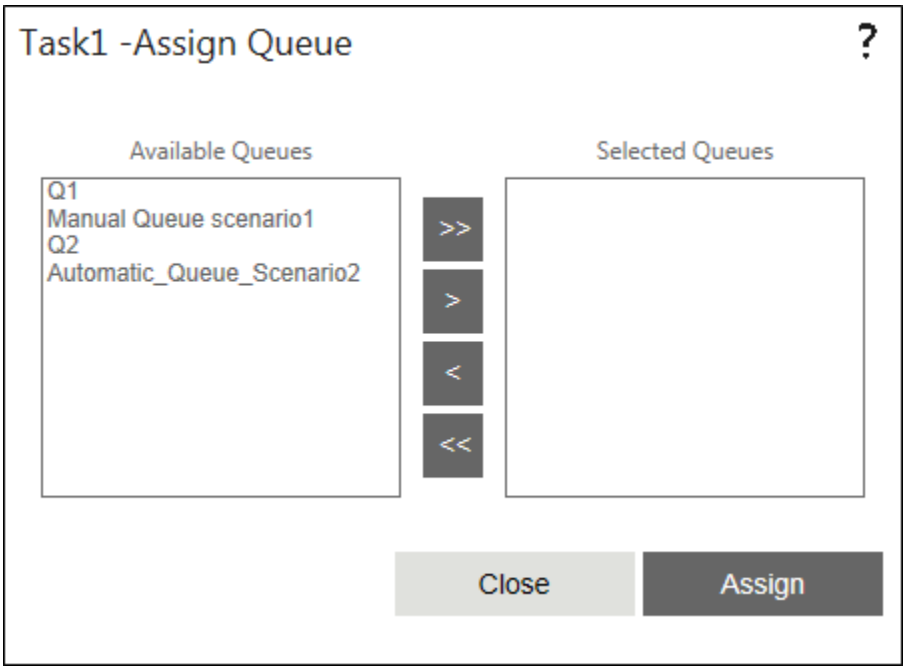
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Task - Assign Queue(s)

Assign Queue(s) Property Window

You can select the queues to which the Task work item is to be sent in this property window. To open this window, click the button for the 'Assign Queue(s)' property.



The window has a pair of list boxes to specify the queue(s). The **Available Queues** list shows all the available queues. You can select the required queue(s) from this list box. The Selected Queues list shows all the selected queues to which the activity will be assigned.

You can use the following buttons located between the list boxes to move the queues to and from the list boxes.

- [>>] - You can click this button to add all the queues in the Available Queues list to the Selected Queues list.
- [>] - You can click this button to add the selected queue(s) to the Selected Queues list.
- [<] - You can click this button to add the selected queue(s) from the Selected Queues list to the Available Queues list.
- [<<] - You can click this button to add all the queues in the Selected Queues list to the Available Queues list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

At the bottom of the window you will find the following buttons:

- **Assign** - You can click this button to assign the activity to the selected queues.
- **Close** - You can click this button to close this window.

Assigning queue system variable

To assign queues instead of setting the queue through **Assign Queue** property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

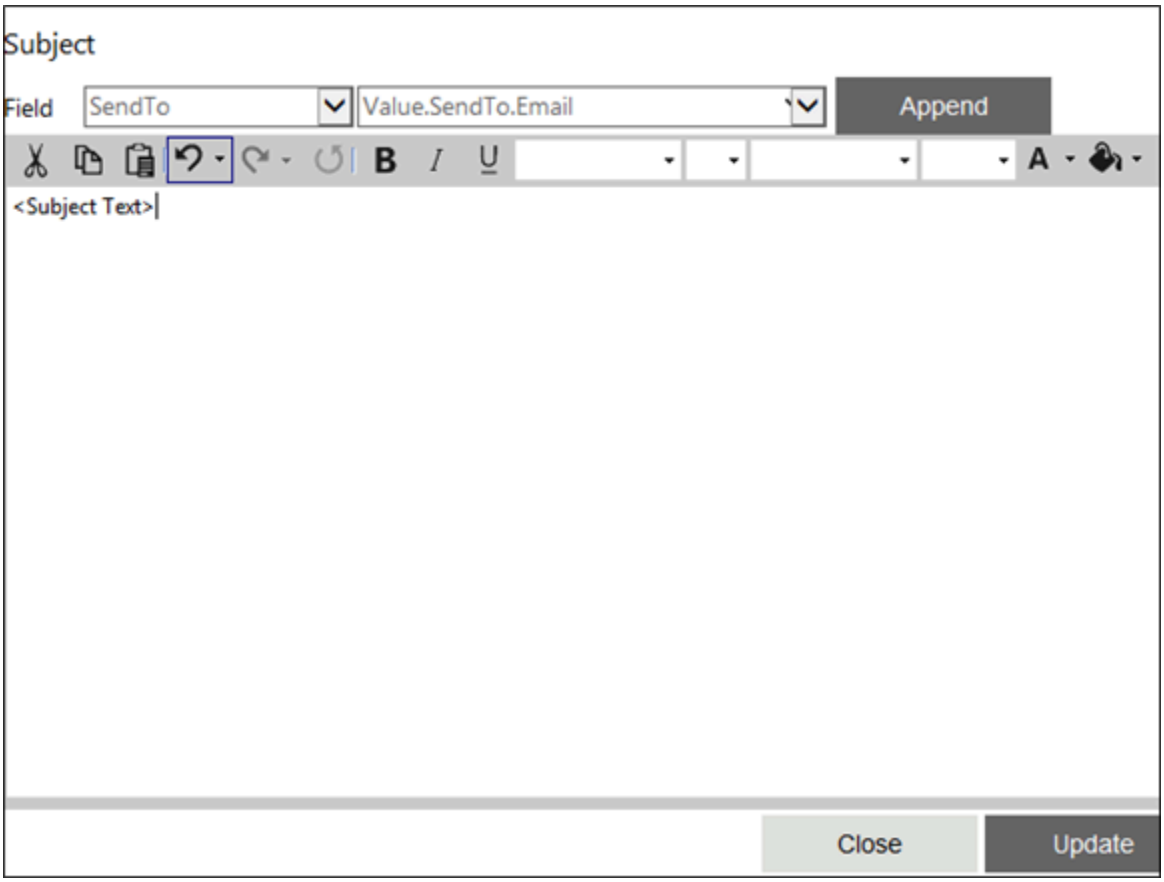
For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the Start activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

To assign multiple queues, specify the queue-ids with semicolon (;) separator. **Skelta.HWS.Queue.QueueCollection** class can be used to retrieve queue information.

Task - Subject

Subject Property window

You can specify a subject for the Task work item assignment in this property window. To open this window, click the button for the 'Subject' property.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The window has a text editor to enter the subject. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the subject. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the subject in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the subject.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the subject using a combination of discrete value and variable or content. For example, you can specify the subject as "Submitted By <%#Variable.SubmittedBy.Email%>". Here the string "Submitted By" should be typed directly in the text editor. The second half of the subject is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the **Field** drop down and clicking **Append**.

The text tool bar above the text editor allows you to manage and format the subject. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the subject.
- **Close** - You can click this button to close this window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task - Body

Body Property window

You can specify a message Body for the Task work item assignment in this property window. To open this window, click the button for the 'Body' property.

The window has a text editor to enter the body text. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/ drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the body using a combination of discrete value and variable or content. For example, you can specify the body message as "Hi, Please cast your vote in the poll conducted by <%#Variable.SubmittedBy.name%>". Here, the string "Hi, Please cast your vote in the poll conducted by" must be typed directly in the text editor. The second half of the body is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the Field drop down and clicking Append.

The text tool bar above the text editor allows you to manage and format the body text. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

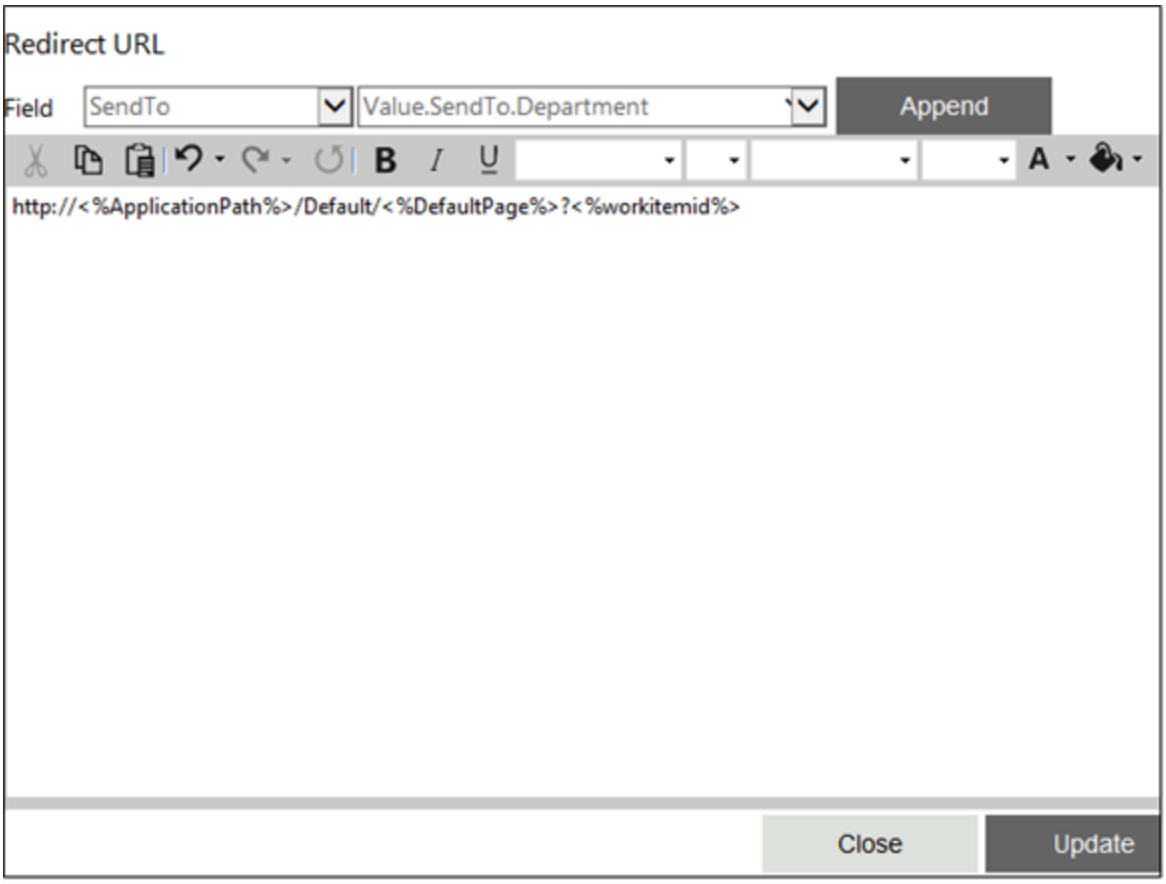
- **Update** - You can click this button to save the body message.
- **Close** - You can click this button to close this window.

Task - Redirect URL

Redirect URL Property window

You can specify a message Redirect URL for the Task activity in this property window. To open this window, click the button for the 'Redirect URL' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the Redirect URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Redirect URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Redirect URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Redirect URL. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

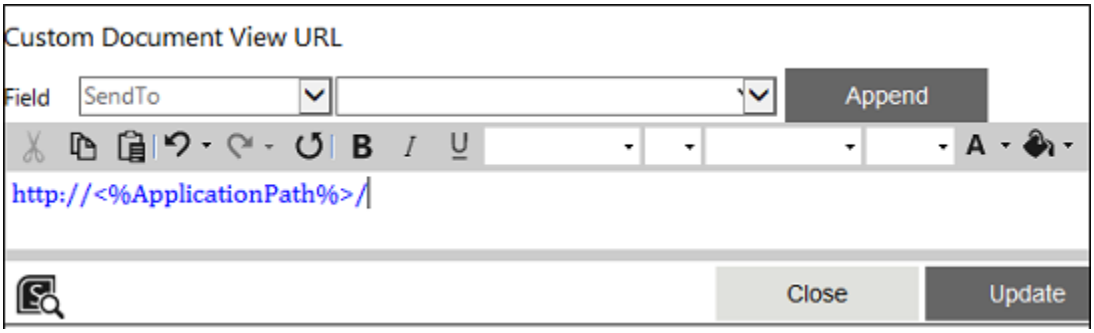
- **Update** - You can click this button to save the Redirect URL.
- **Close** - You can click this button to close this window.

Task - Custom Document View URL

Custom Document View URL Property window

You can specify a message Custom Document View URL for the Task activity in this property window. To open this window, click the button for the 'Custom Document View URL' property.

For more details, see Custom Document View URL.



The window has a text editor to enter the Custom Document View URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Custom Document View URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Custom Document View URL.

- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Custom Document View URL. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the Custom Document View URL.
- **Close** - You can click this button to close this window.

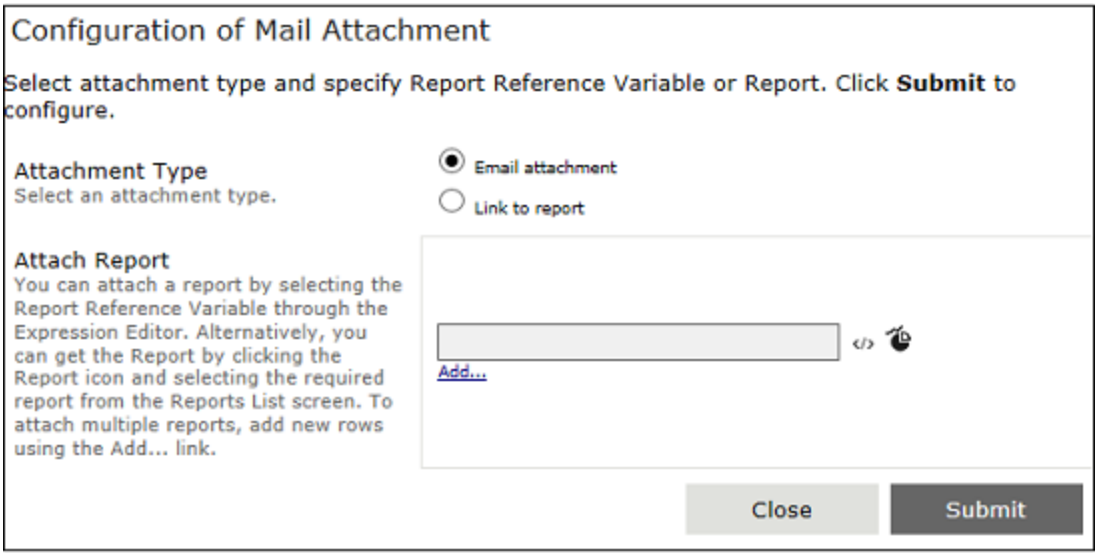
Task - Attach Reports

Configuration of Mail Attachment property window

You can configure the **Attachment** property for the **Approval** activity in this property window.

In this property window, you can configure the report attachments to be emailed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



- **Attachment Type:** Use this property to select the type of attachment for sending the report as an email.
  - **Email attachment:** Select this type to send the report as an email attachment.
  - **Link to report:** Select this type to embed a link in the email to download the configured report.
- **Attach Report:** Use this property to attach a report generated by the **Report Generator** activity within the workflow or to select a report from the list of generated reports.

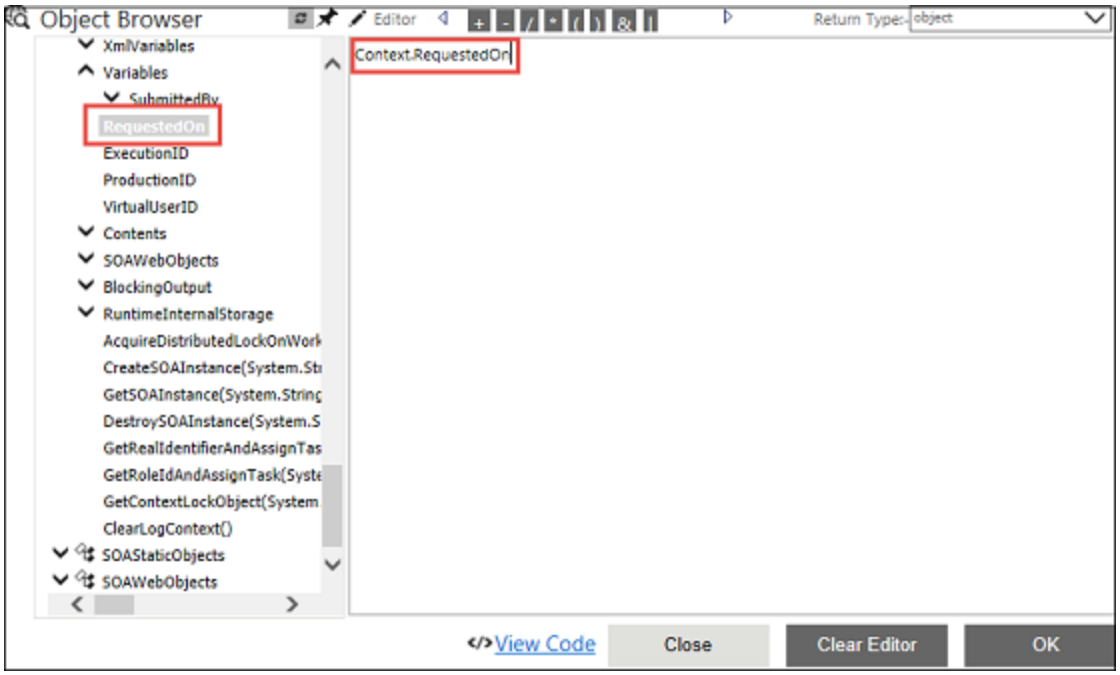
  You can attach a report by selecting the required **Report Reference Variable** using the **Expression Editor**. Alternatively, you can select the report from the Reports List window which consists of the already generated reports.

There must be at least one **Report Generator** activity before an **Approval** activity in the workflow to select the **Report Reference Variable** (using Expression Editor) as an **Attachment**. The **Report Reference Variable** is generated while configuring the **Report Generator** activity. By default, the report reference variable is **[Template Name]_[activity Name]**. For more information about **Report Reference Variable**, see Configure Report Template.

To configure mail attachment

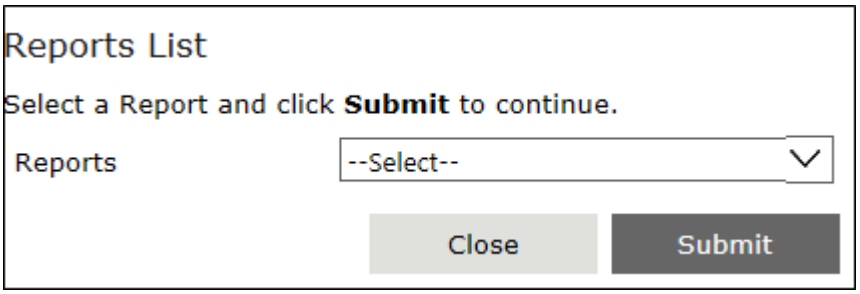
1. Click the edit button for the **Attachment** property.
2. Select the required **Attachment Type**.
3. In the **Attach Report** text box, you can either use the expression editor to get the variable of the report or use the **Reports List** to select the report.
4. To get the variable, click the **Expression Editor** icon given beside the text box to select the required **Report Reference Variable**.
5. In the **Expression Editor** window, select the required variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



6. Click **OK.**
7. Click **Close**.

   The variable appears in the **Attach Report** text box.
8. To select the report from the Reports List dialog box, click the **Reports** icon.



9. In the Reports List dialog box, select the required report, and then click **Submit**.

   This window closes automatically.
10. Click **Close** to close the configuration window.

Task - HTML Template

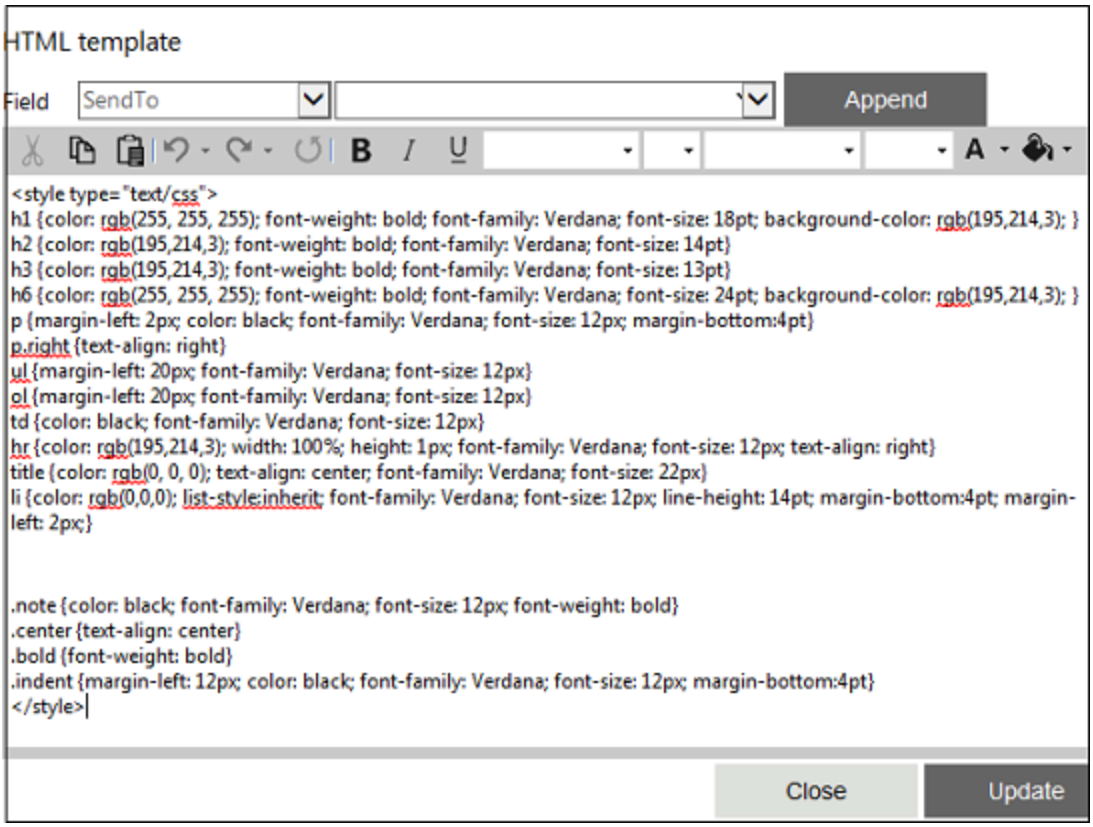
HTML Template Property Window

You can specify a html template for the Task activity in this property window. To open this window, click the edit button for the 'HTML template' property.

In this property window, user can write their own html template for Task activity or user can make use of default Task activity template for customizing the email.

See Using 'HTML Template' property of Task Activity for more information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the body text. You can do this either by entering a html template directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the html template. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

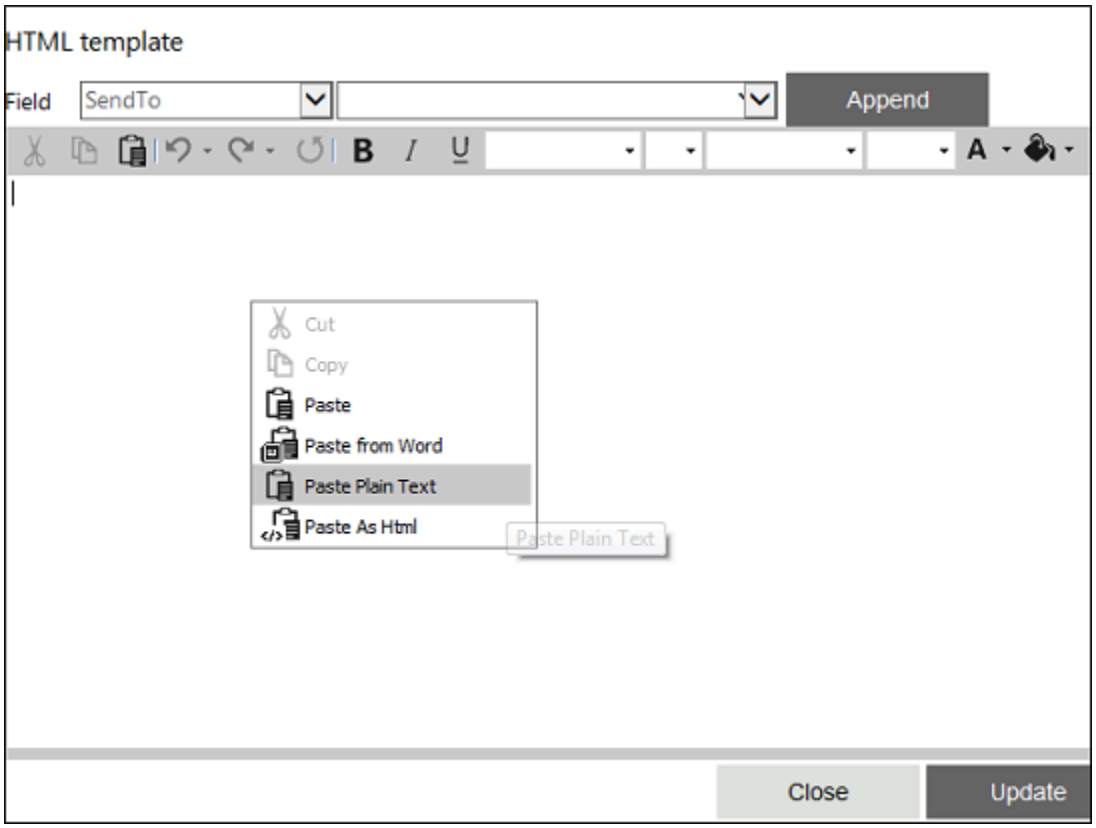
- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

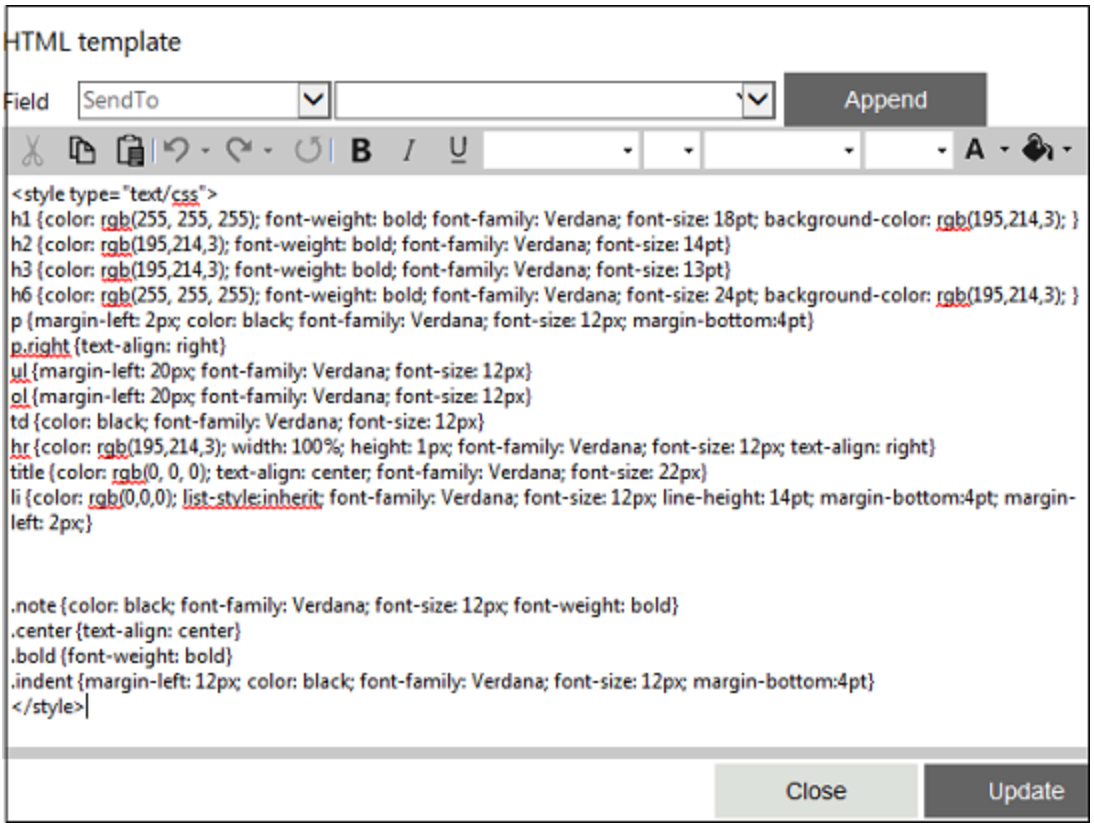
- Update - You can click this button to save the HTML template.
- Close - You can click this button to close this window.

To set HTML temple, follow the steps given below:

1. To specify the HTML template, click the edit icon to open the property window.
2. In the **HTML template** window, provide the template which should be in html format. If you are pasting a copied html content, right-click the **HTML template** window and paste it as "**Paste Plain Text**".



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



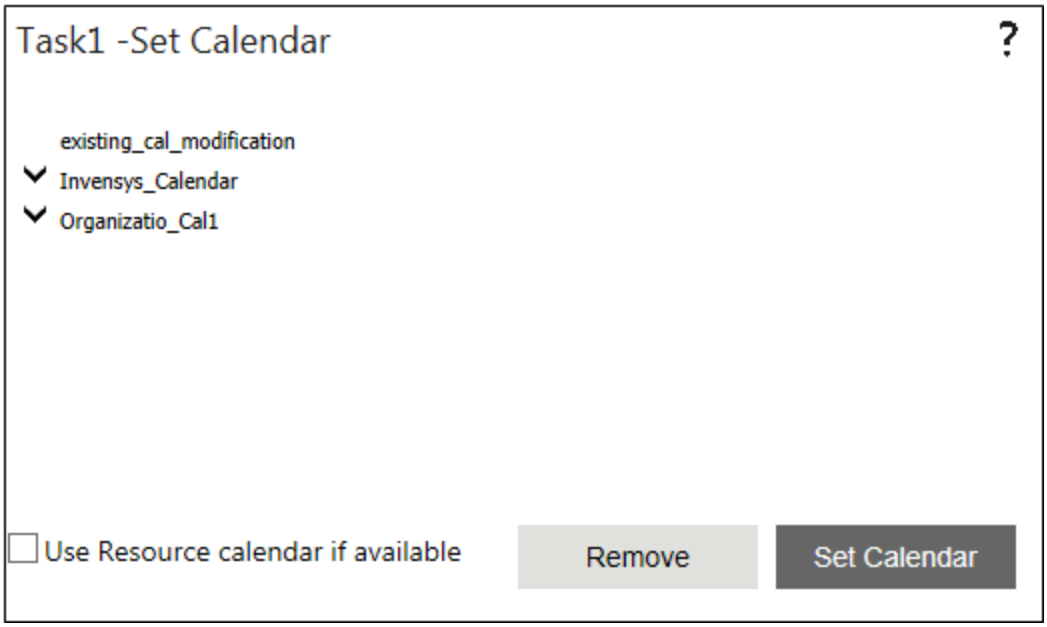
3. Click **Update**.

Task - Set Calendar for Timeout

Set Calendar for Timeout Property window

You can set a Calendar to calculate the timeout for the Task activity in this property window. To open this window, click the button for the 'Set Calendar for Timeout' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



This window lists the available calendars in a tree structure, with lower-level calendars nested within the respective top-level calendar. You can click on a calendar to select it for setting.

There are two buttons in this window.

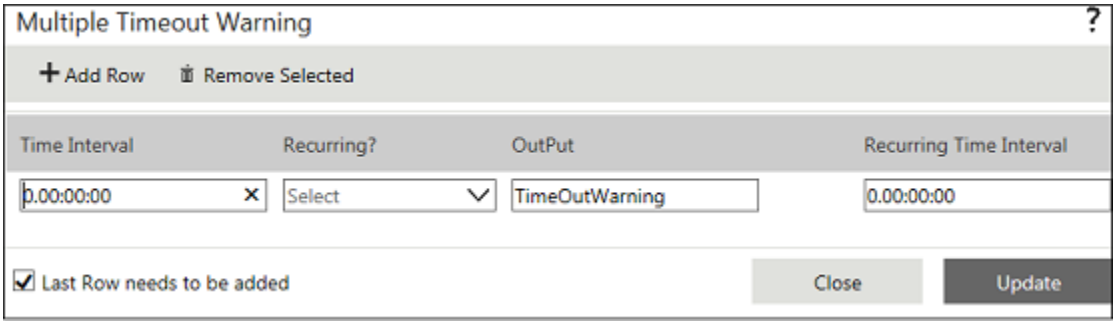
- **No Calendar** - You can click this button to indicate that there is no calendar set for the activity.
- **Set Calendar** - You can click this button to set the selected calendar.

The **User Resource calendar if available** check box can be selected to enable the use of resource calendars if none are displayed in this window for selection.

Task - Multiple Timeout Warning

Multiple TimeOut Warning Property window

You can set the Multiple Timeout Warning in this property window. To open this window, click the button for the 'Multiple TimeOut Warning' property.



The window has two tabs at the top to manage the rows of timeout settings in the grid.

- Add Row - You can click this tab to add the newly entered timeout setting and display a new row to the grid where you can add another timeout.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Remove Selected - You can click this tab to remove the timeout setting row selected in the grid.

The grid has the following parameters that you need to specify for the new timeout setting.

- **Time Interval** - You can use this field to enter the time interval in which you want to generate a timeout warning. You must enter the time in d.hh:mm:ss format.
- **Recurring?** - You can use the drop down in this field to indicate whether the timeout is recurring or not. You can select "Yes" to set the timeout warning to recur.
- **Output** - You can enter the output message corresponding to the timeout warning here.
- **Recurring Time Interval** - You can set the time interval in which the timeout warning has to recur here. Enter the time in d.hh:mm:ss format.

There are two buttons and a check box at the bottom of the screen.

- **Update** - You can click this button to save the newly added timeout settings.
- **Close** - You can click this to close the property window without saving the newly added timeout settings.
- **Last Row needs to be added** - You need to select this check box to save the last row added.

Work Activity

Activity Description

The Work Activity is used in the AVEVA Work Tasks workflow process design in cases where custom activities are required. This activity can be configured using XML configuration files and HTML templates allowing a high degree of flexibility in defining its functionality.

These configuration files are located in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Actions\WorkAction* folder. The Work activity can be configured with Action Outputs, Input Variables, and Output Variables.

Application Scenarios

A typical application scenario would be a Performance Appraisal Workflow where different user-interfaces will have to be shown to evaluators and those being evaluated. The Work activity with its role-based user-interfaces will help implement this functionality.

Workflow Variables for Work:

Steps to define variables for use in the Work activity

This section describes how to define variables for use in Work activities. It gives the syntax for creating the variables and typical examples for different cases.

Variables for capturing WorkItemIds

The Work Item Id's for the Work work item can be retrieved using a variable of type array, with the following syntax:

'^'+CurrentActivity.Name+'WorkItemIds'.

For example, **^WorkActivityWorkItemIds.**

Variables for capturing All Actors' Details

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The details of the actor(s) to whom the Work work item gets assigned can be accessed by declaring a variable of type array with the following syntax:

'^'+CurrentActivity.Name+'ActingUsers'

For example, ^WorkActivityActingUsers.

Variables for capturing Responding Actors' Details

The details of the actors' responding to the Work work item can be accessed by declaring a variable of type resource with the following syntax:

'^'+HWSActivity.ActivityName+'RespondingUser'

For example, ^WorkActivityRespondingUser.

This variable stores the responded user information. If it is more than one user, it will retrieve the last user details.

Note: For a single user, you can declare this variable as Resource type. If task is assigned to more than one user and to get all responded users information, then declare this variable as array.

Variables for capturing Comments

Comments entered in a Work work item can be retrieved using a variable with the following syntax:

'^'+_WorkItem.CurrentActivity.Name+'CommentWork'

For example, **^WorkActivityCommentWork.**

Variables for assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the Start activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

To assign multiple queues, specify the queue-ids with semicolon (;) separator.
Skelta.HWS.Queue.QueueCollection class can be used to retrieve queue information.

Variable for Queue Acting Users

You can declare an array type variable to get updated information about the name of the actor to whom an activity gets assigned from a queue.

In the Workflow Variables, define the variable in the following format:

^+CurrentActivityName+'QueueActingUsers'

Ex: ^HRFormQueueActingUsers

HRForm is the activity name.

For example the value of the user who acts on the HRForm activity will be stored in the variable. The Approval1 task will be assigned to the same user.

Variable for storing timeout warning value

Suppose if you want to store the timeout warning value for Work activity in your workflow. The naming convention for the system variable to be declared is

^<ActivityName>ResponseByWarningTimeout

The value returned by the variable will be in UTC format. For multiple user scenarios the highest timeout value calculated for a user will be available in the variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: Multiple timeout warnings cannot be tracked using this feature.

Activity Properties:

The Work activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Work activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Display Name - This property specifies name to be displayed for the activity in the Work Item List.

Property Type: Optional (This property need not be set if the display name is already configured in the Actions.xml. The name entered here overrides any display name that is configured for this activity in the Actions.xml file.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Delivery Channels - This property is used to specify the delivery channels to deliver this activity to the assigned actor(s). Multiple channels can be specified giving the actor(s) different options to act on the work item.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Delivery Channels property

1. To select the Delivery Channels, click the button for the 'Delivery Channels' property to open the property window.
2. In this window, the Delivery Channels that have been configured are listed on the left along with a check box against each channel. Click and select the check box corresponding to the channel you wish to select. You can select multiple channels by selecting the respective check box.
3. Click the **Save Channels** button.

See Work - Delivery Channel for a detailed description of the Delivery Channels property window.

AdvanceUI Fields - The property is used to specify the advance UI fields of the Work activity. These fields, which define the Work activity, include the input variables, activity outputs and output variables. In addition, the AdvanceUI template can be specified along with the GroupUI template. The Work activity can be configured to be Role-based or Non-Role-based with the Role-based version allowing the flexibility of exposing different sets of fields according to the role selected. To assign activity outputs, click the 'AdvanceUI Fields' button in the Properties pane for the Work activity. A new window pops up. In this window, specify whether the activity will be Role-based or Non-Role-based using the radio buttons provided, and proceed to specify the UI fields in subsequent windows.

Property Type: Mandatory (This property must be set for the action to be used in the workflow.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to set the AdvanceUI Fields property

1. To specify the AdvanceUI fields for the Work activity to be assigned, click the button for the 'AdvanceUI Fields' property to open the property window.
2. In this window, select the type of work activity – Role Based or General – using the **Role Based View** or **General View** radio buttons. Click the **Next** button to go to the next window to add the activity details.
3. If you select General View, you will open the Action Outputs setting window for the General View.
4. Specify the Advance UI XML configuration file in the **File Name** field and the GroupUI HTML Template file in theGroupUI File Name field at the top of this window.
5. After specifying these values click on one of the radio buttons to view the corresponding parameter window. The Activity Output radio button is selected by default and the corresponding controls are displayed in the lower half of the window.
6. Map the outputs of the activity with the possible outputs and specify whether the output is a task completion maker using these parameters. Click Save Action Outputs to save the settings.
7. Click Template Placeholders to open the corresponding window.
8. In this window, specify the values to assign to the template placeholders by selecting from the value type and value drop downs. Click Save Template Placeholders to save the settings.
9. Click the Output Variables radio buttons to display the corresponding parameters. Specify the output variable and the HTML element mapped to it here. Click Save Output Variables to save the settings.
10. If you select Role Based View in step 2, add a new role using the fields in the Role Definition window that is displayed. Click Define Filter to open the parameters to define the participant filter for the role. This needs to be done only for the first instance when the Work activity is being used. Click Next to go to the next step.
11. Select the role for which the view is being configured from the Role drop down.
12. Configure the parameters as described in steps 4 through 10 for the selected role.

See Work - AdvanceUI Fields for a detailed description of the AdvanceUI Fields property window.

Work Item Fields - This property is used to add the Work Item Fields to be used to display application data as well as value in variables and content in the Work Item List when this action is executed.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Work Item Fields property

1. To select the Work Item Fields, click the button for the 'Work Item Fields' property to open the property window.
2. In this window, the Work Item Fields that have been set are listed in a grid. Click a row to select the corresponding Work Item Field.
3. Right-click the row, and select "SetWorkItemField" from the context menu to set it.

See Work - Work Item Fields for a detailed description of the Work Item Fields property window.

Actors

The properties in the Actors group have been described below. You can use these properties to specify the actors who will work on the activity and related properties.

Assign Actor(s) - This property specifies the actor(s) who has to take up the Work activity. It is tagged with the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Ignore Unavailable Actors property which in turn is tagged with Alternate Actor Allowed. This will be explained in detail later.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any resource(s).)

Steps to set the Assign Actor(s) property

1. To specify the actors, click the button for the 'Assign Actor(s)' property to open the property window.
2. In this window, build an expression to select the actors using the actors properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.

   Add the condition to the expression by clicking Add. This will display the expression to the text area.

   You can remove a condition already added to the expression by clicking Undo.

   You can also clear an existing expression if necessary by clicking Clear. This is useful if you want to clear large expressions in the text area in order to build a new expression.

   After building the expression, click Update to save it.

   See Work - Assign Actor(s) for a detailed description of the Assign Actor(s) property window.

See the *Using XML Variables* section in the About XML Variables topic for details on how to use the XMLVariables to build the expression.

Assign Queue(s) - This property specifies the queue(s) to which the Task activity is to be assigned.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any Queue(s).)

Steps to set the Assign Queue(s) property

1. To specify the queue(s), click the button for the 'Assign Queue(s)' property to open the property window.
2. In this window, select the queue(s) from the Available Queues list box. You can select multiple queues by selecting with the
3. After selecting the required queue(s) click the [>] button to move it to the Selected Queues list box. You can click the [>>] button to select all the queue(s) and move them to the Selected Queues list box.
4. You can remove any queue from the selected list by selecting it and clicking the [<] button. You can remove all the queues by clicking the [<<] button. After adding the required queue(s), click Assign to assign the activity to the selection.
5. Click Close to close the window without saving the selection.

See Work - Assign Queue(s) for a detailed description of the Assign Queue(s) property window.

Min. Ownerships [% or Number] - This property, which can be specified as a number or percentage, is the minimum number of actors who should take ownership of the Work assignment. This property is related to the 'Timeout warning for Ownership' and 'Timeout for Ownership' properties. To illustrate, when 'Timeout for Ownership' is reached, the activity first checks if the ownership count is less than Min.Ownerships (if it is set). If it is, the work item will be removed from all Work Item lists. If the 'Min.Ownerships' has already been satisfied, then the work item will be removed from the Work Item lists of those who have not taken ownership. If the Min.Ownerships value is 0, then the activity checks if the ownership count is less than Max.Ownerships (if it is set). If it is, the work item will be removed from all Work Item lists. Else the work item will be removed from the Work Item lists of those who have not taken ownership. If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

removed from all Work Item lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)
Max. Ownerships [% or Number] - This property, which can be specified as a number or percentage, is the maximum number of actors who should take ownership of the Work request. If 'Timeout for Ownership' is specified and achieved, and the 'Min.Ownerships' is also satisfied, then the activity waits for action from those who have taken ownership. If the Min.Ownerships value is set to 0, then it checks if the ownership count is less than

Max.Ownerships. If it is, the work item will be removed from all Work Item lists. Else the work item will be removed from the Work Item lists of those who have not taken ownership. If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work Item lists. Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Property Type: Optional (This property needs to be set only if necessary.)

Enable Take Ownership? - If this property is set to Yes, then the actor is provided with a Take Ownership option in the Work items list to take ownership of the task. If you do not not required the Take Ownership option in the Work items list, then set this property as No.

Property Type: Optional (This property needs to be set only if necessary.)

Note: By default, this property is set as Yes.

Steps to set the Enable Take Ownership? property

1. Select Yes or No from the Enable Take Ownership? drop-down list.

Re-Assign Allowed? - If this property is set then the actor is provided with a Re-Assign option. Using this the actor can forward the work item to another actor.

Property Type: Optional (This property needs to be set only if necessary.)

Notification Contents - The properties in the Notification Contents group have been described below. You can use these properties to specify the content used to notify actors about the activity.

Subject - This is the subject of the Work assignment. The subject can be customized for each actor.

Property Type: Optional (This property need not be set. If it is not set, the default subject line from the Actions.xml file is displayed.)

Steps to set the Subject property

1. To specify the subject, click the button for the 'Subject' property to open the property window.
2. In this window, enter the subject text directly in the text editor. If you want to customize the subject you can do so using the Field controls to add resource property values. For example, to add a subject that includes the email id of the actor submitting the Task, you can do as follows: Type in the text "Submitted By". Select "Variable" and "Variable.SubmittedBy.Email" from the two drop downs for Field and click Append. This will add the placeholder for the variable after the text entered in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the email id.
3. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
4. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
5. After adding the subject, click Update to save it.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Click Close to close the window without saving the subject.

See Work - Subject for a detailed description of the Subject property window.

Body - This is the actual message of the Work sent to the actors. The message body can be customized for each actor. There is an Enterprise Console Explorer(plug-in) icon in the Body property window to get the Work item details URL. See Enterprise Console Control (EC Plug-in) for more information.

Property Type: Optional (This property need not be set.)

Steps to set the Body property

1. To specify the message body, click the button for the 'Body' property to open the property window.
2. In this window, enter the body text directly in the text editor.
3. If you want to customize the body you can do so using the Field controls to add resource property values. For example, to add a body that includes the name of the actor submitting the Task, you can do as follows: Type in the text "Hi, Please act on the task assigned to you by". Next select "Variable" and "Variable.SubmittedBy.name" from the two drop downs for Field and click Append. This will add the placeholder for the variable after the text entered in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the name.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the body, click Update to save it.
7. Click Close to close the window without saving the body.

See Work - Body for a detailed description of the Body property window.

From Email Address? - This property is used to specify the email address from which notification messages will be sent for the activity.

Property Type: Optional (This property needs to be set only if necessary. If it is not set, the default value in Activities.xml will be used.)

Hide Responses? - If this property is set to 'No', then actors can see the responses of other actors in a group Work scenario. To view these responses i.e., ownerships, Works or rejections, select the 'Take Ownership' button. This will open a new pop up with the responses of other actors who have taken ownership.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Priority - This property is set to indicate the priority of the Work assignment. Enter the values 0-33 to set as Low, 34-66 for Medium, or 67-99 for High. The default value is set to 34.

Property Type: Optional (This property needs to be set only if necessary.)

Send Notification Email - If this property is set to 'Yes', then notification emails will be sent to the actors. If it is set to 'Yes-HTML', then the emails will be sent in HTML format.

Property Type: Optional (This property needs to be set only if necessary.)

Notify if alternate actor is considered - If this property is set to 'Yes', then notification emails will be sent to the actors when an alternate actor is considered.

Property Type: Optional (This property needs to be set only if necessary.)

Redirect URL - This property is used to get the URL of the web page to redirect the client to, in the case of

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

synchronous execution of the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Redirect URL property

1. To specify the Redirect URL, click the button for the 'Redirect URL' property to open the property window.
2. In this window, enter the Redirect URL directly in the text editor.
3. If you want to customize the Redirect URL you can do so using the Field controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for Field, and clicking Append. This will add the placeholder for the variable in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the URL.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Redirect URL, click Update to save it.
7. Click Close to close the window without saving the Redirect URL.

See Work - Redirect URL for a detailed description of the Redirect URL property window.

Custom Document View URL - This property is used to get the URL of the page to which the custom document dlls does a post. The data posted to the page are application, workflow, executionid, executiondetailsid, and workitemid. The posted data can be retrieved as Request.Form["workitemid"].

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Custom Document View URL property

1. To specify the Custom Document View URL, click the button for the 'Custom Document View URL' property to open the property window.
2. In this window enter the Custom Document View URL directly in the text editor.
3. If you want to customize the Custom Document View URL you can do so using the Field controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for Field, and clicking Append. This will add the placeholder for the variable in the text editor. At run-time, this placeholder will be replaced by the corresponding value, i.e. the URL. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
4. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
5. After adding the Custom Document View URL, click Update to save it.
6. Click Close to close the window without saving the Custom Document View URL.

See Work - Custom Document View URL for a detailed description of the Custom Document View URL property window.

Decision Influencers

The properties in the Decision Influencers group have been described below. You can use these properties to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

specify some parameters that influence the outcome of the activity.

Max Actors Required for WorkAssignment Completion [% or Number] - This property is used to check what percentage or number of actors should take an action for Work. Decision on the Work is taken only after the specified number or percentage of resource(s) have taken the action. The activity waits for the actors' action until the 'Timeout for action' is reached.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Min Actors Required for WorkAssignment Completion [% or Number] - This property specifies the minimum number of actors to act on the Work before the 'Timeout Warning for Action'. If the action limit is reached before the 'Timeout Warning for Action' then the warning message is not shown.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Ignore Unavailable Actors? - If this property is set to 'No', when the Work activity is assigned to the actors list a check is first done to see whether the actor is available. If the actor is not available then a check is made for an alternate actor assigned to the unavailable actor. Simultaneously, the value of the property 'Alternate Actor Allowed' is checked. If that is also set, then the Work is assigned to this actor.

Property Type: Optional (This property needs to be set only if necessary.)

Alternate Actor Allowed? - This property is set to assign a particular Work to an alternate actor in the absence or unavailability of any actor.

Property Type: Optional (This property needs to be set only if necessary.)

Escalation & Timeout

The properties in the Escalation & Timeout group have been described below. You can use these properties to specify escalation and timeout settings for the activity.

Set Calendar for Timeout - This property is used to specify the calendar to be used to determine activity timeouts.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to Set Calendar for Timeout

1. To specify the calendar for timeout, click the button for the 'Set Calendar for Timeout' property to open the property window.
2. In this window select the required calendar by clicking on the calendar name in the tree structure.
3. Click Set Calendar to set the selected calendar.
4. You can specify that a resource calendar should be used, if available, by clicking to select the corresponding check box.

See Work - Set Calendar for Timeout for a detailed description of the Set Calendar for Timeout property window.

Minimum available time for Activity - This property is used to check the availability of an actor to whom a task must be assigned while executing the activity. The availability check for the actor is done based on the Calendar settings (Global or Resource Calendar). If the time taken to complete the task is within the value set for the **Minimum available time for Activity** property, then the task will be assigned to that actor. However, if the time taken to complete the task exceeds the value set for the **Minimum available time for Activity** property, then the task will be assigned based on the values set for the **Ignore Unavailable Actors?** and **Alternate Actor Allowed?** properties.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Timeout Warning for Ownerships - This property specifies the time at which a warning is provided to a actors to take ownership of the work item assigned to him/her. If the actors do not take ownership within this time the engine is notified with a warning. Thus if 'Min.Ownerships' is set, and it is not satisfied before the given time then a warning is sent. If Min.Ownerships is 0, it will check for Max.Ownerships. If the ownership count does not reach the Max.Ownerships value within the set time, a warning is sent. If both Min.Ownerships and Max.Ownerships are not set, this property is ignored.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout for Ownerships - This property specifies the time until which actors are allowed to take ownership, i.e. the time until which the Work activity will wait for actors to take ownership. If Min.Ownerships is set and is not reached, the work item is removed from all work item lists. If Min.Ownerships is 0, it will check if ownership count is less than Max.Ownerships. If it is, the work item is removed from all work item lists. Else the work item will be removed from the work item lists of those who have taken ownership. The work item will remain in the work item list of those who have taken ownership and wait for action from them.

Property Type: Optional (This property needs to be set only if necessary.)

Understand how Ownership Properties relate to each other:

- Work activity is assigned to the actors in the Work assignment list by the workflow engine. Work activity has a default property set that at least one actor has to make a Take Ownership of the activity.
- A drop down interface has been provided for the actor to Take Ownership. However, the actor can act on the Work without taking ownership first, in which case ownership will be considered to be done, i.e. if the actor acts upon the Work activity without taking ownership, then it is considered as an ownership with action taken.
- After the actor has taken ownership, a link is provided to the actor, which opens a pop up with the details of the other actors. This is again tagged with the 'Hide Responses?' property. The link will be provided only If 'Hide Responses?' is set as 'No'. The pop up will have details like Actor name, Ownership Status, Action taken, Comments.
- If the 'Min.Ownerships' property is set and the Ownership count has reached the set limit [percentage or number] then the 'Timeout Warning For Ownership' property is not considered.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] and the 'Time Out For Ownership' is also reached then the work item is removed from other approvers' work item lists and the work item waits for action from those who have taken ownership.
- If the 'Min.Ownerships' limit is not reached for the Work action, and the 'Timeout Warning For Ownership' has been exceeded then the actor is notified with a warning by including an email or an information activity, with the Timeout Warning for Ownership link to it.
- If 'Timeout For Ownerships' is set, if the Work activity has not reached the minimum limit (Min.Ownerships) and the 'Time Out For Ownerships' has been reached, then the work item is removed from the work item lists of all the actors.
- If 'Min.Ownerships' = 0 and if 'Max.Ownerships' is set, the activity waits for actors to take ownership or action until the 'Time Out For Ownership' is reached (if this is set).
- If none of the Ownership properties are set then the activity waits until the actor takes ownership or acts upon it.
- Timeout Warning for Activity - This property specifies when a reminder has to be provided for the actors to act on the Work request, if the action count has not been reached. If properties such as Max.Ownerships, Affirmative Action for Work (AAA), and Action Limit (AL) are set, once the property values are reached then a warning is sent to the actor. When the timeout is reached, the following checks are made: If AL is set, when the action count i.e. the no. of Works or rejections (AC) < AL, then a warning is sent. If AL = 0 and AAA is set,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

and when AC < AAA, then a warning is sent to the participant.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Timeout for Activity - This property's value is used to decide until what time the activity should wait for the actors' action. If properties such as Max.Ownerships, AAA, AL are set, once the property values are reached the work item is removed from the work item lists of all the actors. When the timeout is reached, the following checks are made: If AL is set, when the action count i.e. the no. of Works or rejections (AC) < AL, then the timeout for action is performed and the work item is removed from all work item lists. If AC > AL, it checks with the no. of Works or 'decision count' (DC) and AAA. If DC >= AAA (if it is set), the activity is considered approved; otherwise, it is rejected. If AAA = 0 and if DC >= AL, the action is considered completed. If AL = 0 and AC < AAA, then the timeout for action is performed and the work item is removed from all work item lists. If AL = 0 and AC > AAA, it will check with DC. If DC > AAA it is completed. If AL = 0 and AAA = 0, then if DC is at least 1, then it is considered completed.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Multiple TimeOut Warning - This is used to set recurring timeout warnings that will be sent to the actors who have to take action. To set multiple timeout warnings, click the 'Multiple TimeOut Warning' button in the Properties pane for the Work activity. A new window pops up. In this window, set the interval for the recurring warning as well as the output message. You can also set the Recurring Time Interval. This can be used to set a different recurring time out after the first timeout happens. For example, if 'Time Interval' for the property is set as 7 days and 'Recurring Time Interval' is set as 2 days, then the first time out will happen after 7 days and there will be a recurring time out every second day after the seventh day (first) time out.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Steps to set the Multiple Timeout Warning property

1. To set the multiple timeout warning, click the button for the 'Multiple Timeout Warning' property to open the property window.
2. Enter the time interval for the timeout in the Time Interval field.
3. Select "Yes" from the drop down for the Recurring? field if you want the timeout warning to recur. Select "No" otherwise.
4. Enter the output message corresponding to the timeout warning in the Output field.
5. Enter the time interval for the recurring timeout in the Recurring Time Interval field.
6. After entering the content details, click Add Row to add this timeout setting and display a new row for entering another timeout setting.
7. Remove rows if necessary by selecting them and clicking Remove Selected.
8. After specifying the timeout settings, click Update.

See Work - Multiple Timeout Warning for a detailed description of the Multiple TimeOut Warning property window.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity. Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Miscellaneous

Redirection Enabled? - This property enables redirection. When this property is set to Yes, the workflow

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

redirects the user to the URL in the Redirect URL property in the next activity.

Activity outputs: The Work Activity returns the following values -

Action Completed: Displayed when all the actors have taken part.

Action Completed Affirmative Percentage Not Reached: Displayed when all the actors have not taken part.

Not Enough Resources to Take Acknowledge: Displayed when the filter condition did not yield enough actors to whom the request can be sent.

Timeout - Take Acknowledge: Displayed when the specified minimum number of actors did not take ownership of the request in the given time.

Timeout - Action: Displayed when some or all of the specified actors did not perform the required action in the given time.

Timeout Warning - Take Acknowledge: Displayed when a warning is generated that the specified minimum actors have not taken ownership of the request in the given time.

Timeout Warning - Action: Displayed when a warning for timeout is generated because some or all the actors did not perform the required action in the given time.

Note: The following outputs will return only if user assigned tasks to a queue user.

Resource Unavailable for Queue - If a queue is assigned to a task & if there are no resources available in the queue then this output will get triggered.

TimeOut for Queue Task - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting the completion of that task.

Timeout Warning -Queue - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting in sending a warning out put. However, the task will still be sleeping.

Timeout Warning -Queue Participant - This output will get triggered when the task in the queue is picked by user but still not acted on that task within the queue escalation time out settings resulting in triggering a warning output. However, the task will still be sleeping.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Work - Delivery Channel

Purpose

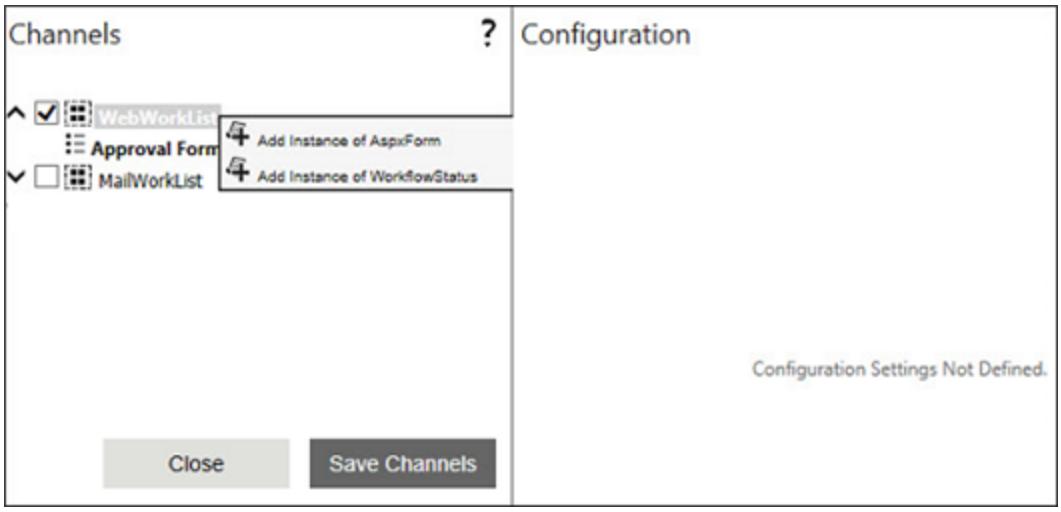
The purpose of this chapter is to describe the feature of WorkflowStatus Form of Delivery Channels property under the category Configuration of any human activity available in AVEVA Work Tasks.

Description: This feature enables us to view the status of the workflow in the work item list in the graphical form.[Optional]

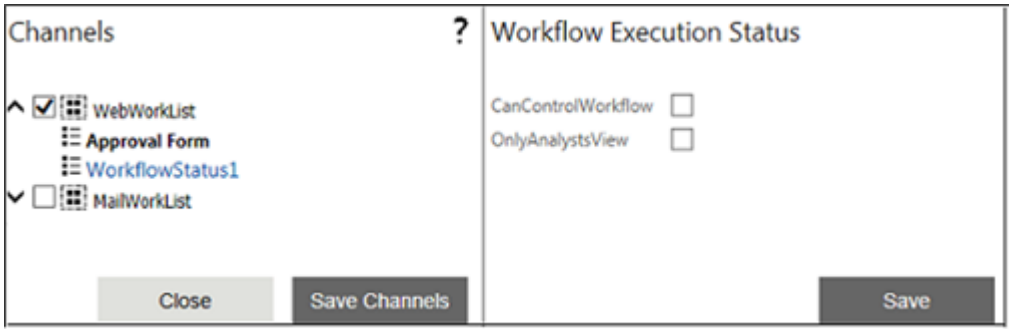
1. Select the Human Activity.
2. Click the **Delivery Channels** Property.

   This will open a window like the one shown below.

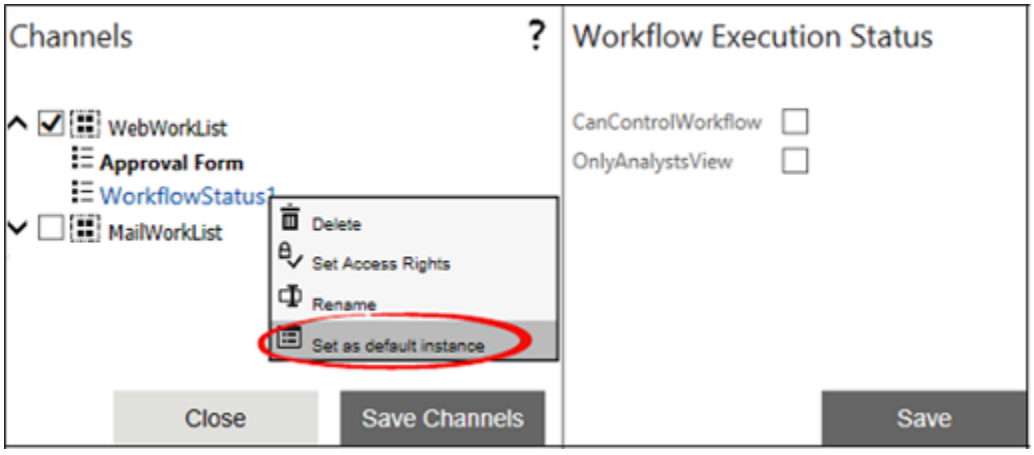
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Right-click **WebWorkList** to display the context menu with the **Add Instance of Workflowstatus** option as shown below.



4. Set the Properties for the Execution View Status Graphical view.
5. Click **Save**.
6. If you want to set the WorkflowStatus form as the default instance, right-click **WorkflowStatus1**, and select the **Set as default instance** option from the context menu.



7. Click the **Save** and **Save Channels** buttons.

Output

Now, when you click any row in the Work items list, you will find this Execution view page being displayed as the detailed view if you set it as a default instance.

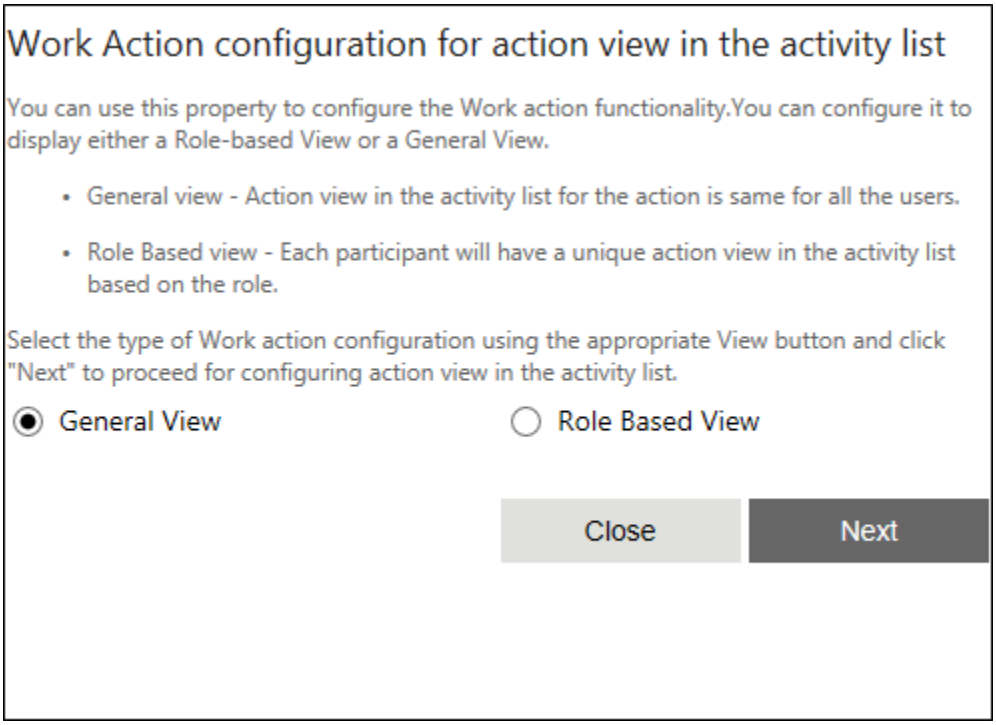
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Otherwise, the workflow status form will appear in the context menu of each row in the Work items list and you can open the execution view by selecting it from the context menu.

Work - AdvanceUI Fields

AdvanceUI Fields Property window

You can launch the user-interface to configure the functionality of the Work activity from this property window. To open this window, click the button for the 'AdvanceUI Fields' property.



You can select the type of work activity view to configure – General or Role Based – using the two radio buttons in this window.

- General View
- Role Based View

You can click **Next** to go to the next window for adding the activity details. This will vary depending on the type of view selected.

Configuration for General View

The window shown below is opened when you click the General View radio button in the previous window. It is used to specify the action outputs for the Work activity with the General View. This window has two fields common to the General View, which are displayed at the top of all the corresponding property windows.

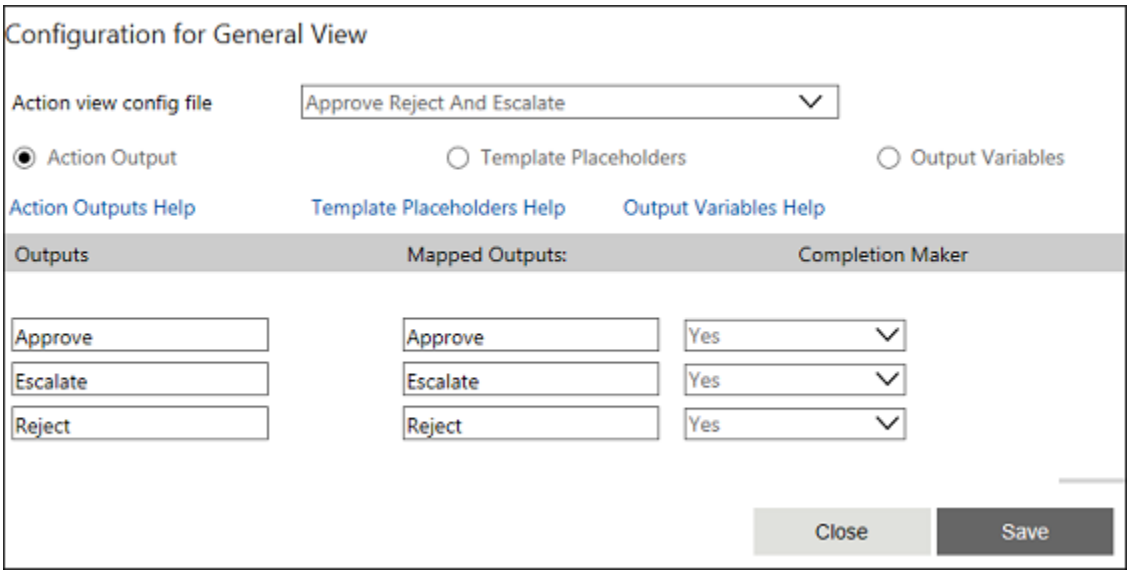
- **Action view config file**

You can specify the Advance UI XML configuration file in this field.

There are three radio buttons – **Action Outputs**, **Template PlaceHolders**, and **Output Variables** – that can be selected to display the respective property setting controls in the lower half of the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The window shown below is displayed when the Activity Output is selected. It can be used to map the outputs of the activity with the possible outputs and specify whether the output is a task completion maker.

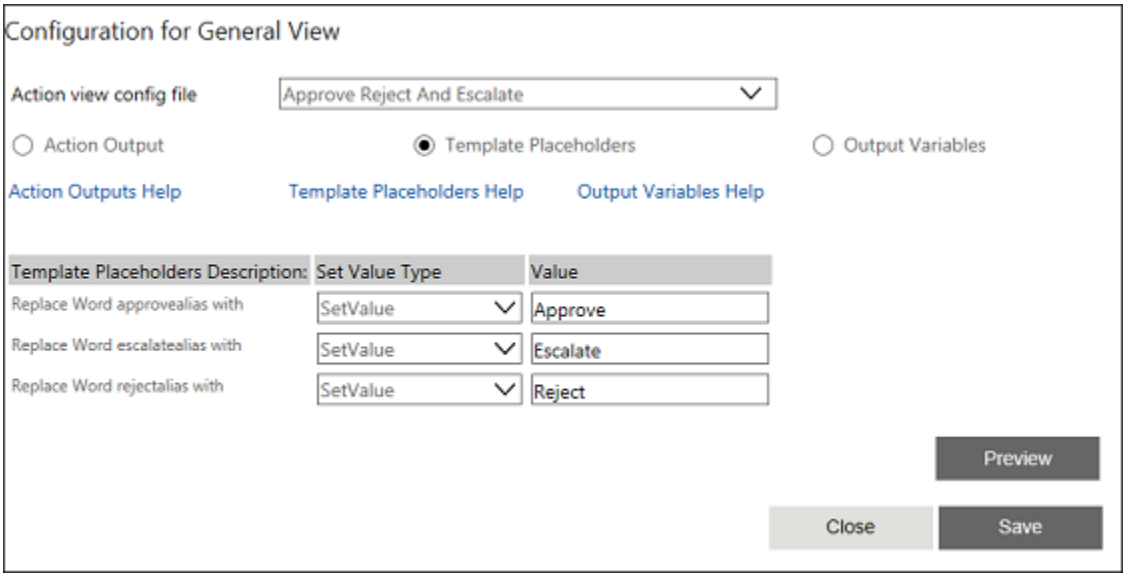


You can set the following parameters here.

- **Outputs** - This shows the output of the activity.
- **Mapped Outputs** - You can specify the outputs to map to here.
- **Completion Maker** - You can select "Yes" from this drop down to indicate that the corresponding output will mark the completion of the activity.

You can click **Save Action Outputs** to save the new settings.

The window shown below is displayed when you select the **Template Placeholders** radio button. You can specify the values to the template placeholders in the Work activity here.



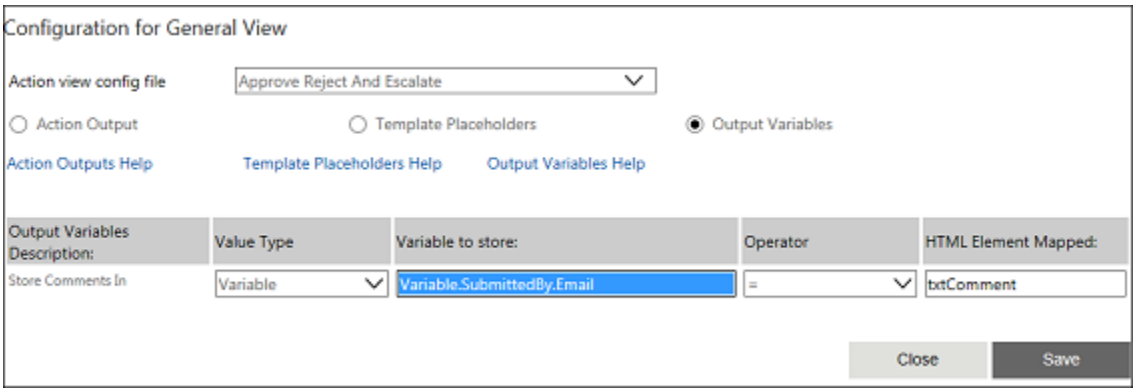
The window lists all the template placeholders configured for the activity under the Template Placeholders Description column. You can specify the value for each placeholder using the following controls.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **The Value type drop down** - You can select the type of value to assign from the options in this drop down. You can set a discrete value, variable or content here.
- **The Value text box/drop down** - You can specify the value here. If you selected "SetValue" in the previous drop down this control will be a text box. It will show a drop down otherwise, where you can select an appropriate variable or content collection.

You can click **Preview** to see a preview of the template setting. You can click Save Template Placeholders to save the settings.

The window shown below is displayed when you select the **Output Variables** radio button. You can specify the output variable and the HTML element mapped to it here



The window lists all the output variables for the activity under the Output Variables Desc column. You can specify the value for each placeholder using the following controls.

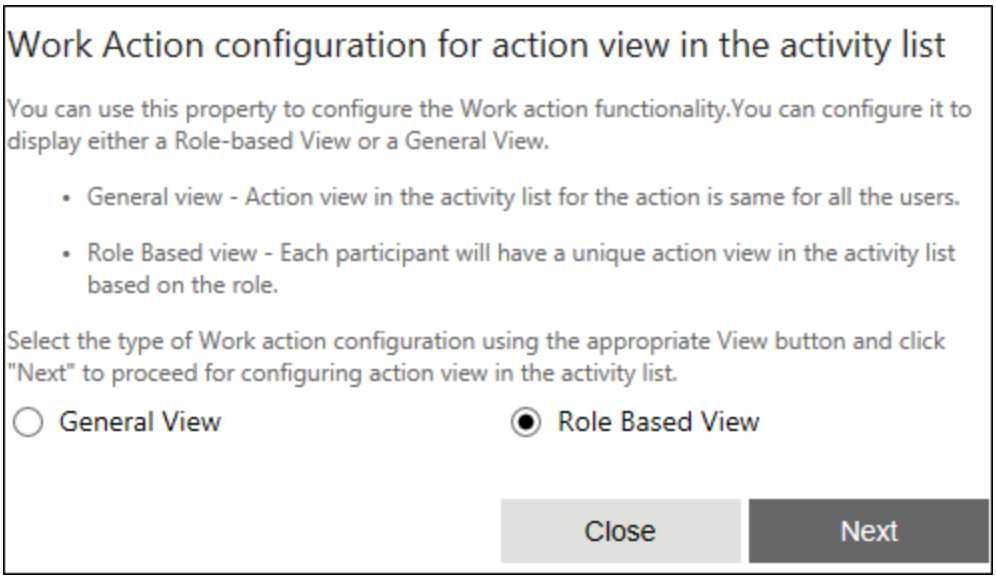
- **Value Type** - You can select the type of value to assign from the options in this drop down. You can set a variable or content here.
- **Variable to store** - You can specify the variable to store the output here. You can select an appropriate variable or content collection from this drop down.
- **Operator** - You can select an operator from this drop down.
- **HTML Element Mapped** - You can specify the HTML Element that is to be mapped to the output variable.

You can click **Save Output Variables** to save the settings.

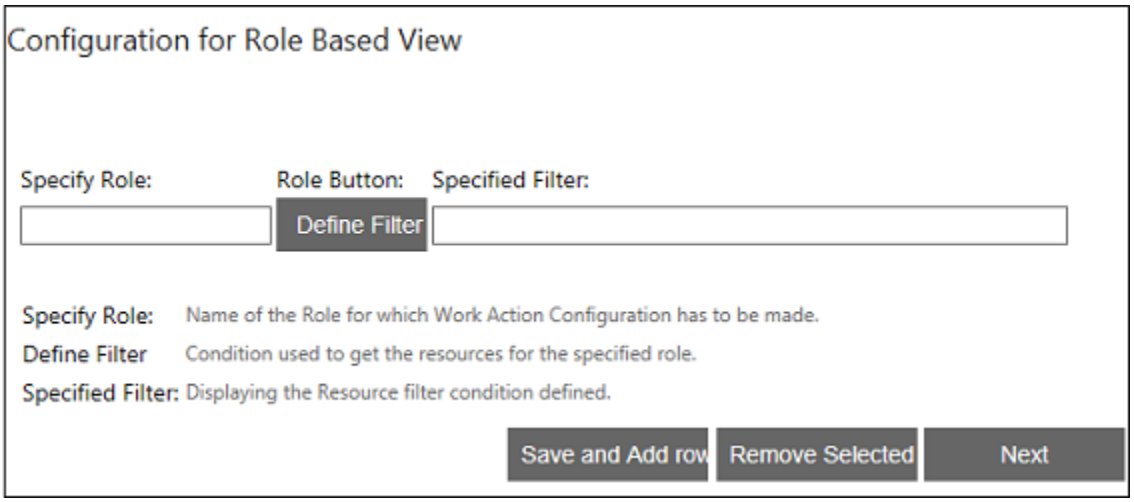
Configuration for Role Based View

The window shown below is opened when you click the **Role Based View** radio button in the main AdvanceUI Fields property window. It is used to define a role for the Work activity. It is displayed when you use the Work activity for the first time. This window has three columns in this window, which you can use to add roles.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



- **Specify Role** - You can specify the name of the new role in this field.
- **Role Button** - This column has the Define Filter button. You can click this button to open the filter definition button.
- **Specified Filter** - The field in this column displays the newly defined filter.

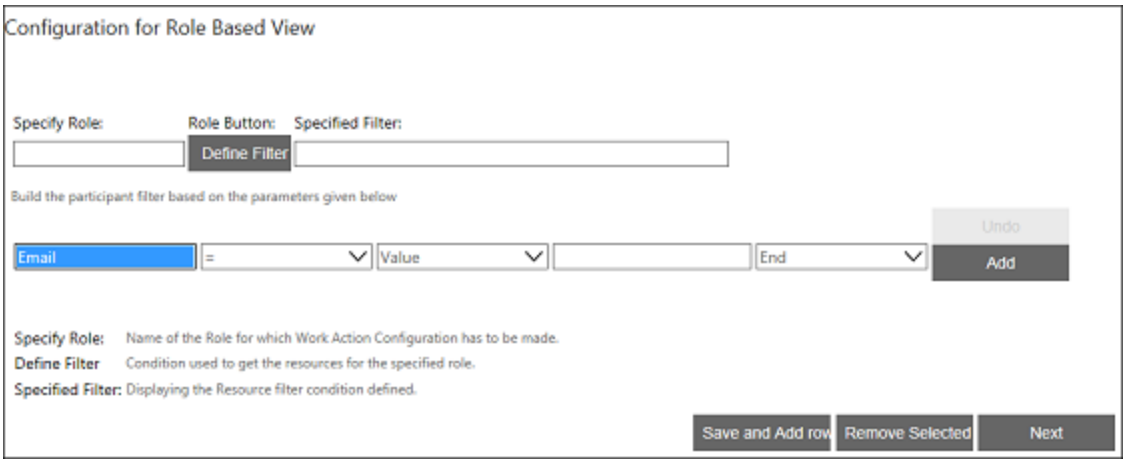


This window has three buttons.

- **Save and Add Row** - You can click this button to save the role defined in the current row and add another row to add a role.
- **Remove Selected** - You can select a row and remove it using this button.
- **Next** - You can click this button after adding the required roles for the Work activity to continue to the main window for the Role Based View.

The window shown below is displayed when you click the **Define Filter** button in the previous window. You can build a role filter condition for the new role here. The following image shows how to build a role filter condition

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a set of drop downs at the top to select the different parameters required to build an expression to define the filter. You can also find related buttons next to the drop downs.

- **The Attribute drop down** - You can select the attribute to be used to filter the participant, e.g., "AdministratorId", in this drop down.
- **The Operator drop down** - You can select the operator for the expression here. E.g., "="
- **The Value type drop down** - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- **The Value text box/drop down** - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down. If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- **The Logical operator drop down** - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the Specified Filter field in the previous window.
- **Undo** - You can click this button to remove a condition you have added to the expression.

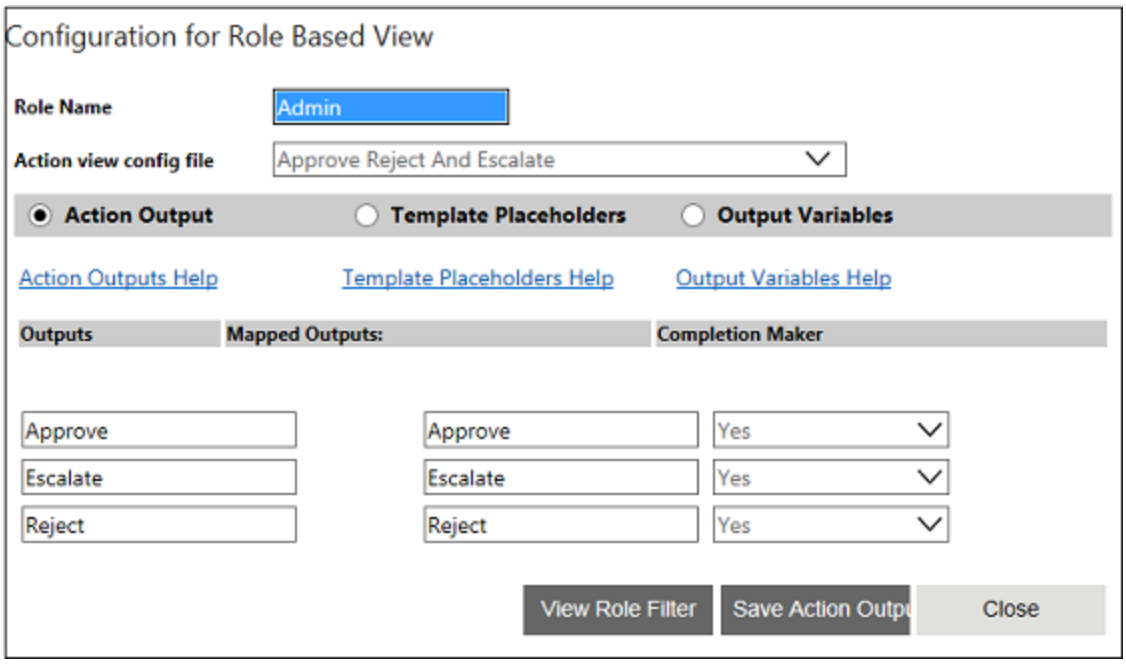
The window shown below is displayed when you click Next in the Role Definition window, if you are using the Work activity for the first time. If you are not using the activity for the first time this window will be opened when you select the Role Based View radio button in the ActivityUI property main window. You can specify the action outputs for the Work activity with the Role Based View. This window has two fields common to the General View, which are displayed at the top of all the corresponding property windows.

- **Role Name** - You can select the role for which the View you are configuring will be accessible from this drop down.
- **Action view config file Name** - You can specify the Advance UI XML configuration file in this field.

There are three radio buttons – **Action Outputs**, **Template Placeholders**, and **Output Variables** – that can be selected to display the respective property setting controls in the lower half of the window.

The window shown below is displayed when the Action Output is selected. It can be used to map the outputs of the activity with the possible outputs and specify whether the output is a task completion maker.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Action Output window for Role Based View

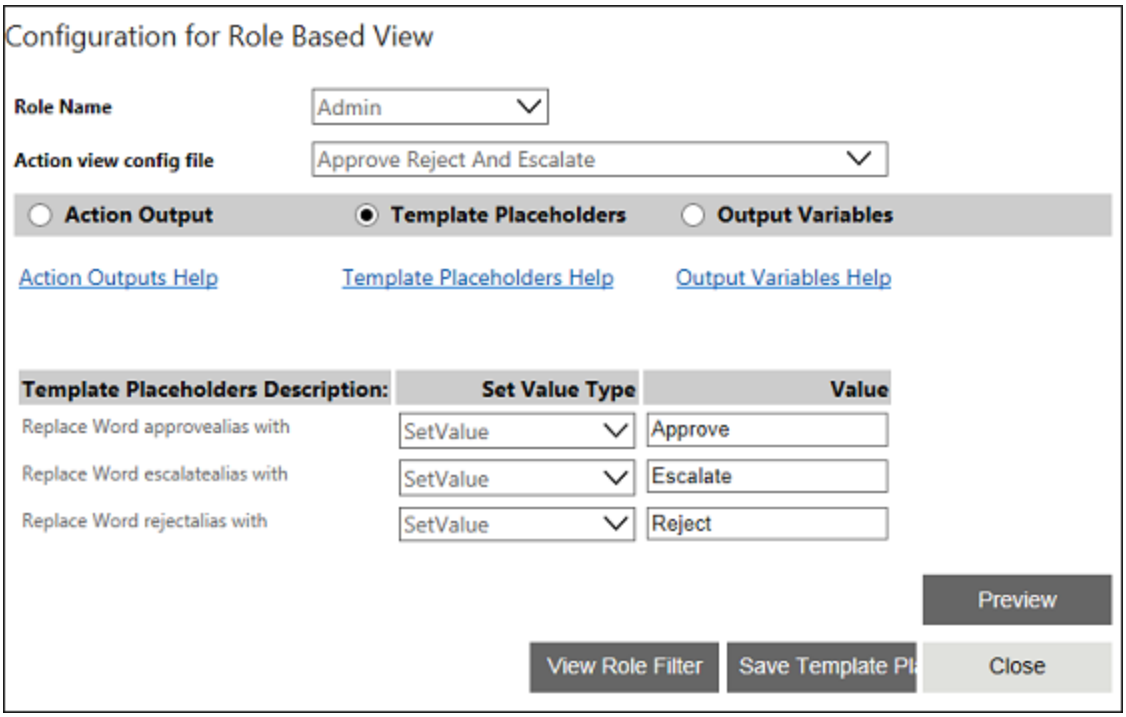
You can set the following parameters here.

- **Outputs** - This shows the output of the activity.
- **Mapped Outputs** - You can specify the outputs to map to here.
- **Completion Maker** - You can select "Yes" from this drop down to indicate that the corresponding output will mark the completion of the activity.

You can click the **Save Action Outputs** to save the new settings.

The window shown below is displayed when you select the Template Placeholders radio button. You can specify the values to the template placeholders in the Work activity here.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



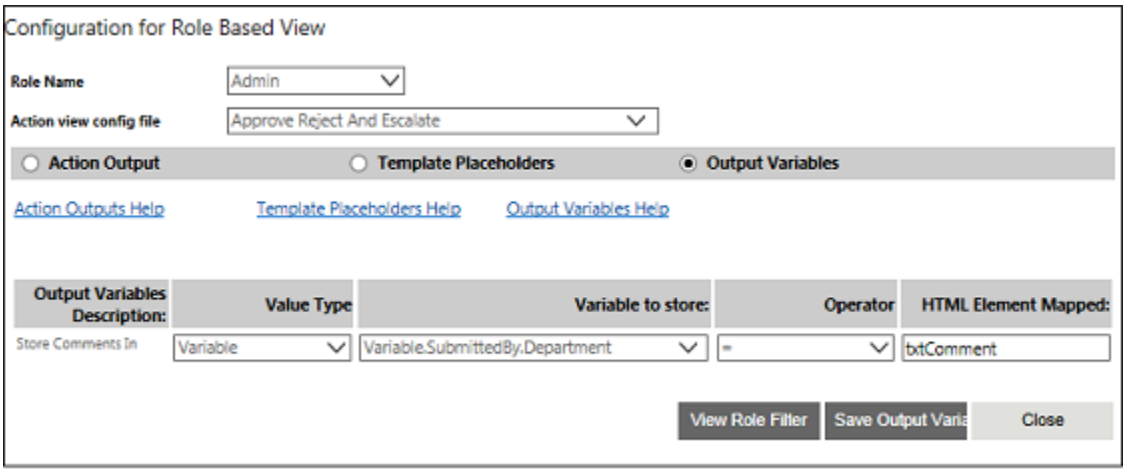
Template Placeholders window for Role Based View

The window lists all the template placeholders configured for the activity under the Template Placeholders Description column. You can specify the value for each placeholder using the following controls.

- **The Value type drop down** - You can select the type of value to assign from the options in this drop down. You can set a discrete value, variable or content here.
- **The Value text box/drop down** - You can specify the value here. If you selected "SetValue" in the previous drop down this control will be a text box. It will show a drop down otherwise, where you can select an appropriate variable or content collection.

You can click Preview to see a preview of the template setting. You can click Save Template Placeholders to save the settings.

The window shown below is displayed when you select the Output Variables radio button. You can specify the output variable and the HTML element mapped to it here.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Output Variables window for Role Based View

The window lists all the output variables for the activity under the Output Variables Desc column. You can specify the value for each placeholder using the following controls.

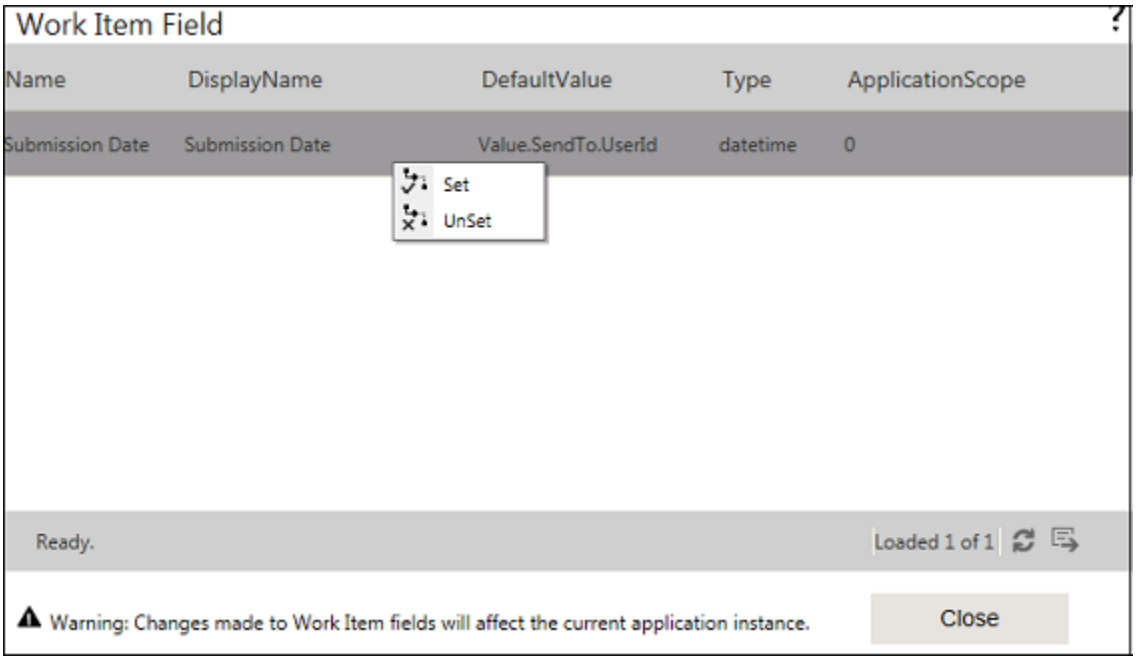
- **Value Type** - You can select the type of value to assign from the options in this drop down. You can set a variable or content here.
- **Variable to store** - You can specify the variable to store the output here. You can select an appropriate variable or content collection from this drop down.
- **Operator** - You can select an operator from this drop down.
- **HTML Element Mapped** - You can specify the HTML Element that is to be mapped to the output variable.

You can click the **Save Output Variables** to save the settings.

Work - Work Item Fields

Work Item Fields Property Window

You can set the Work Item Fields for this activity in this property window. To open this window, click the button for the 'Work Item Fields' property.



The window has a grid listing all the Work Item Fields that have been set for the process.

- Each row in the grid represents a Work Item Field. You can click a Work Item Field to select it.
- The context menu has the Set WorkItemField option, which you can use to set the selected Work Item Field for this activity. It also has the UnSet WorkItemField option to remove the Work Item Field setting.
- **Close** - You can use this button to close the window.

Note: While entering a value in the Work Item field, it is recommended to keep it less than the specified database column size. If there is a need to increase the size of the Work Item field, the user should manually

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

increase the column size in the database.

Work - Assign Actor(s)

Assign Actor(s) Property window

You can set the actors to whom the Work work item is to be sent in this property window. To open this window, click the button for the 'Assign Actor(s)' property.

See Assigning a Task to a Manager for more details.

The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

Different Methods of Assigning an Actor

- **Assigning an Actor by building an expression:**

Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

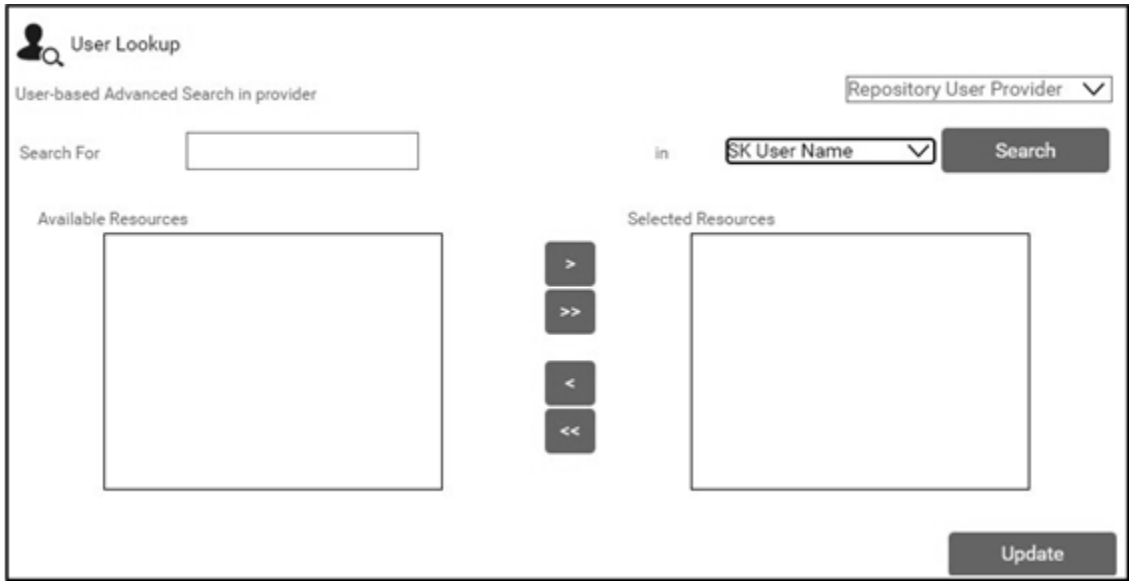
At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

- **Assigning an Actor through User Lookup**

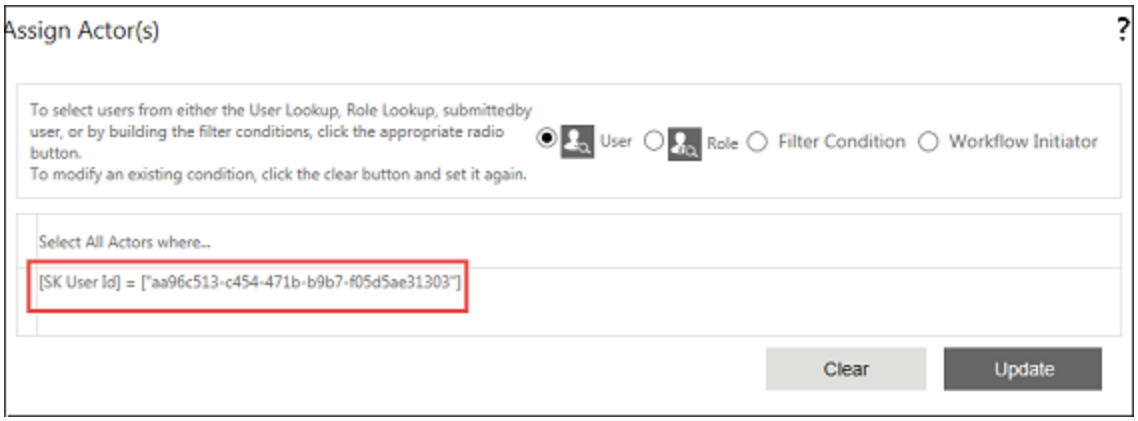
Through the User Lookup you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.
2. The **User Lookup** dialog is displayed, search and select the required actor.



3. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

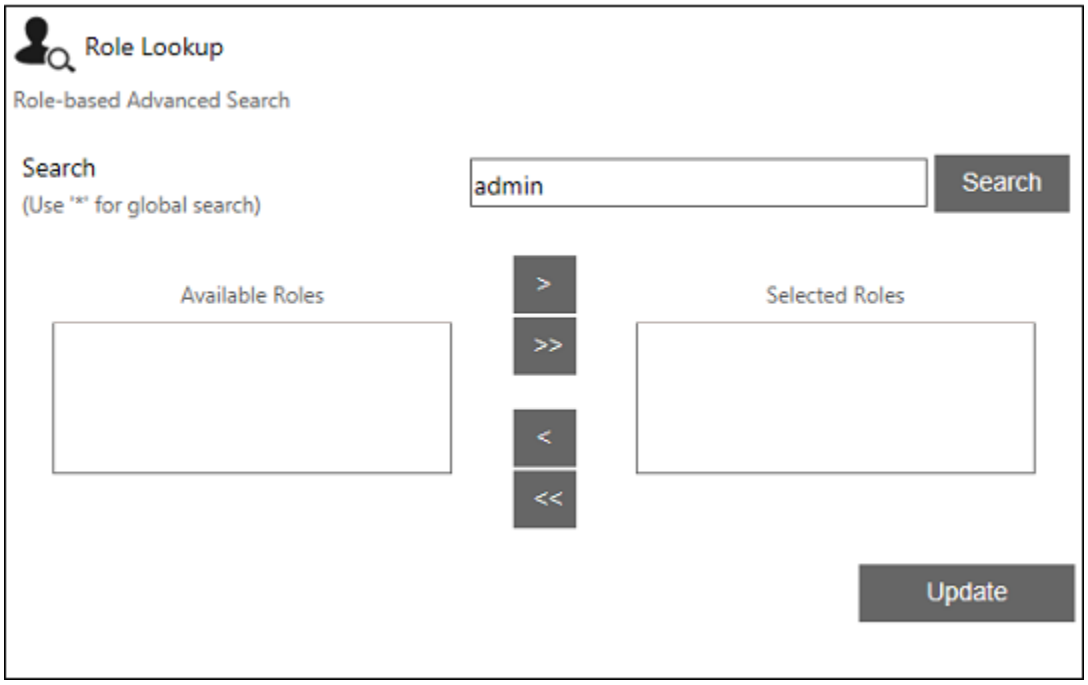


4. Click **Update** in the Assign Actor(s) window to update the selected actor(s) in the Assign Actor property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

- **Assigning an Actor through Role Lookup**

Through the Role Lookup you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

1. Select the **Role Lookup** option.
2. The **Role Lookup** dialog is displayed, search and select for the required role.



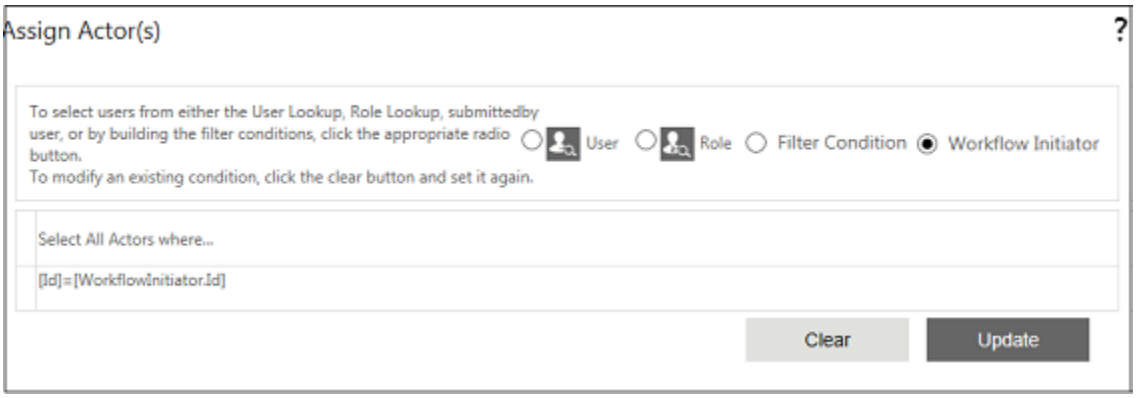
3. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click **Update** in the Assign Actor(s) window to update the **Assign Actor** property.

- **Assigning an Actor through Workflow Initiator**

Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

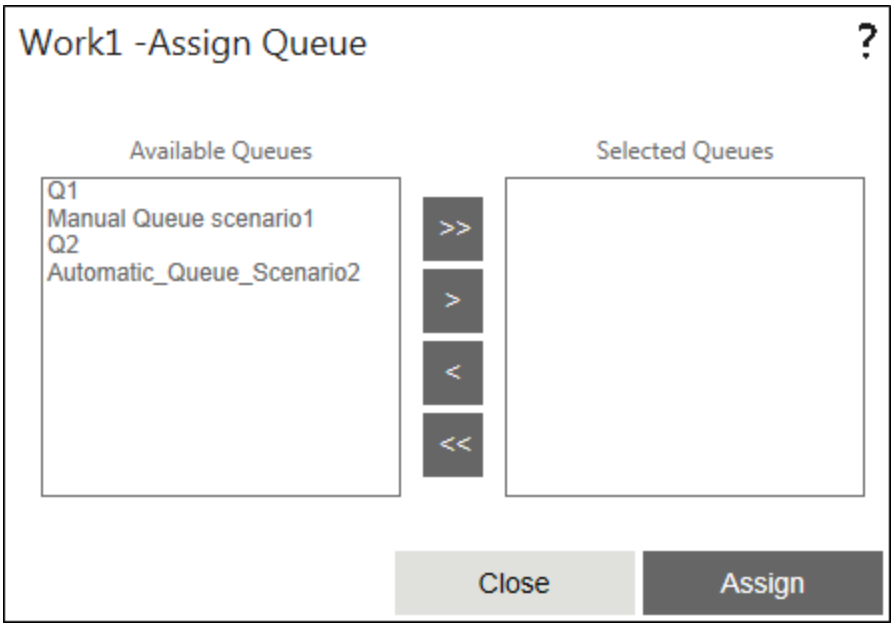


Work - Assign Queue(s)

Assign Queue(s) Property Window

You can select the queues to which the Work work item is to be sent in this property window. To open this window, click the button for the 'Assign Queue(s)' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Assign Queue(s) Properties window for Work

The window has a pair of list boxes to specify the queue(s). The **Available Queues** list shows all the available queues. You can select the required queue(s) from this list box. The **Selected Queues** list shows all the selected queues to which the activity will be assigned.

You can use the following buttons located between the list boxes to move the queues to and from the list boxes.

- [>>] - You can click this button to add all the queues in the Available Queues list to the Selected Queues list.
- [>] - You can click this button to add the selected queue(s) to the Selected Queues list.
- [<] - You can click this button to add the selected queue(s) from the Selected Queues list to the Available Queues list.
- [<<] - You can click this button to add all the queues in the Selected Queues list to the Available Queues list.

At the bottom of the window you will find the following buttons:

- **Assign** - You can click this button to assign the activity to the selected queues.
- **Close** - You can click this button to close this window.

Assigning queue system variable

To assign queues instead of setting the queue through **Assign Queue** property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the Start activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

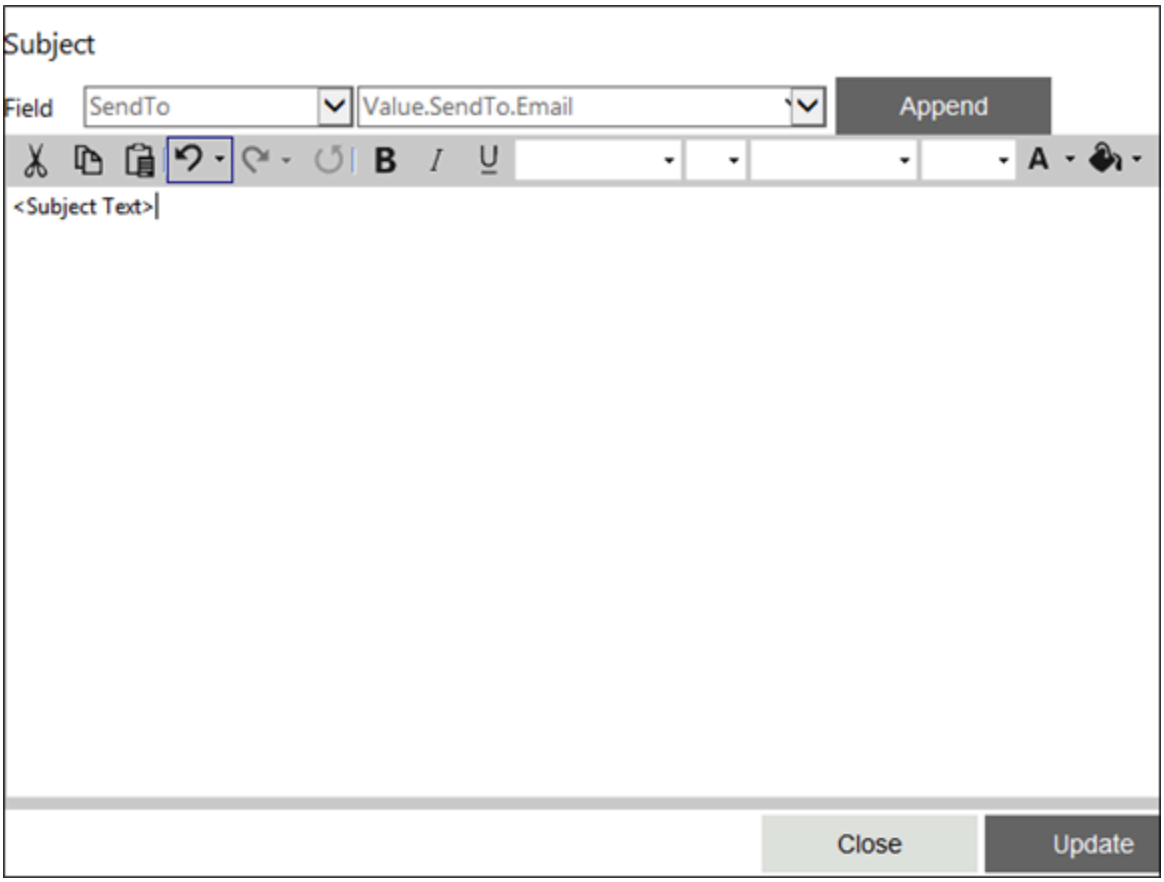
To assign multiple queues, specify the queue-ids with semicolon (;) separator. **Skelta.HWS.Queue.QueueCollection** class can be used to retrieve queue information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Work - Subject

Subject Property Window

You can specify a subject for the Task work item assignment in this property window. To open this window, click the button for the 'Subject' property.



The window has a text editor to enter the subject. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the subject. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the subject in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the subject.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the subject using a combination of discrete value and variable or content. For example, you can specify the subject as "Submitted By <%#Variable.SubmittedBy.Email%>". Here, the string "Submitted By" should be typed directly in the text editor. The second half of the subject is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the **Field** drop down and clicking **Append**.

The text tool bar above the text editor allows you to manage and format the subject. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

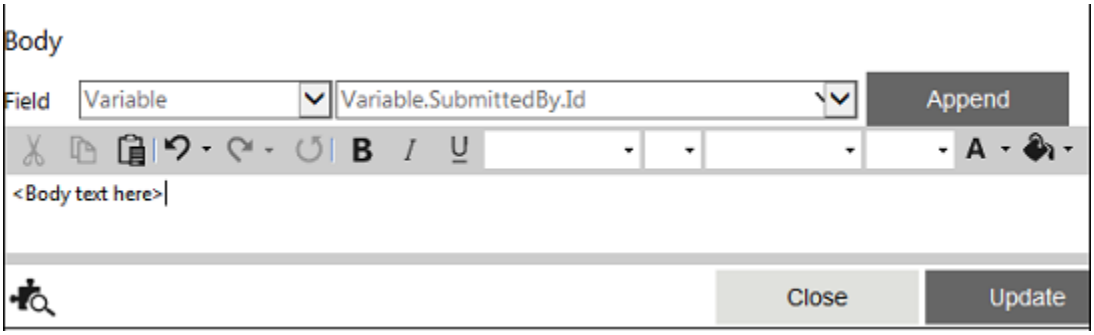
At the bottom of the window, you will find the following buttons:

- **Update** - You can click this button to save the subject.
- **Close** - You can click this button to close this window.

Work - Body

Body Property Window

You can specify a message Body for the Work work item assignment in this property window. To open this window, click the button for the 'Body' property.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The window has a text editor to enter the body text. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the body using a combination of discrete value and variable or content. For example, you can specify the body message as "Hi, Please cast your vote in the poll conducted by <%#Variable.SubmittedBy.name%>". Here, the string "Hi, Please cast your vote in the poll conducted by" must be typed directly in the text editor. The second half of the body is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the Field drop down and clicking Append.

The text tool bar above the text editor allows you to manage and format the body text. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window, you will find the following buttons:

- **Update** - You can click this button to save the body message.

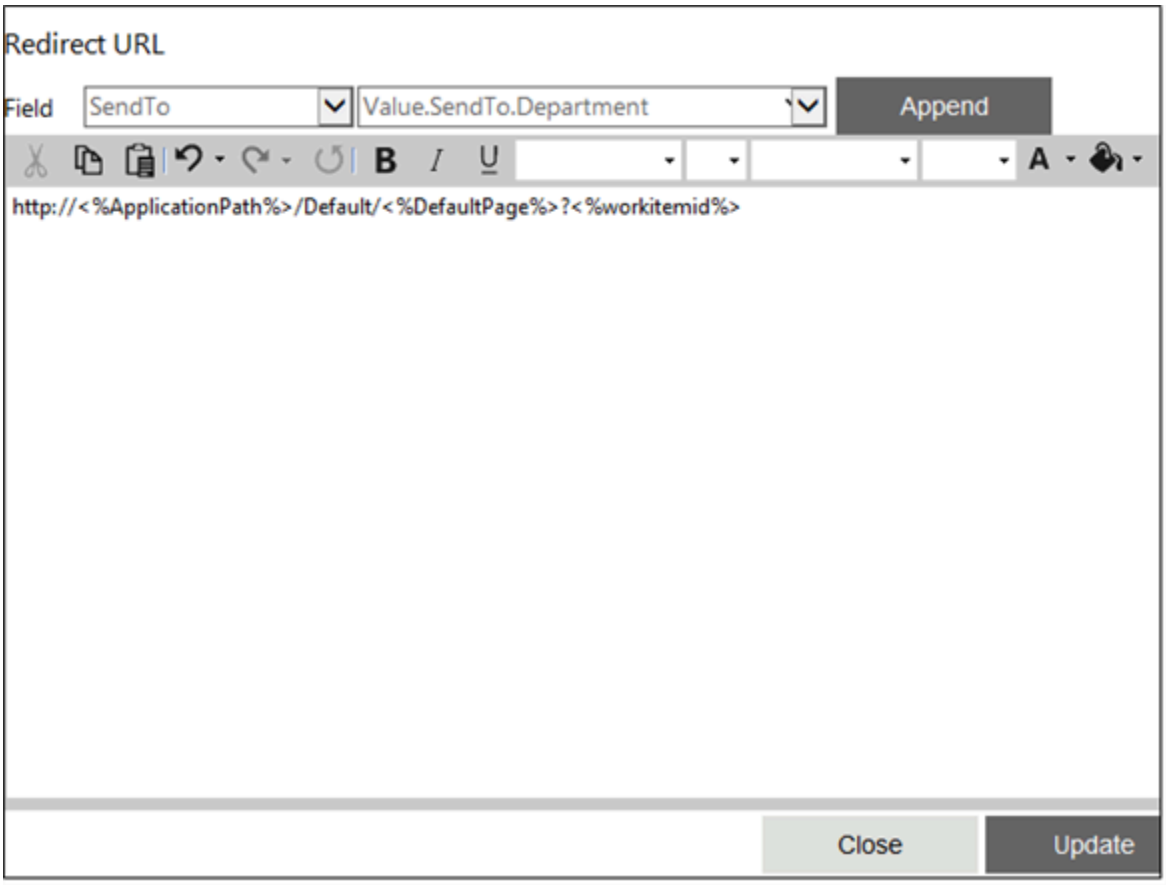
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Close** - You can click this button to close this window.

Work - Redirect URL

Redirect URL Property Window

You can specify a message Redirect URL for the Work activity in this property window. To open this window, click the button for the 'Redirect URL' property.



The window has a text editor to enter the Redirect URL. You can do this either by entering a URL directly in the text editor or by using the Field controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Redirect URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Redirect URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Redirect URL. It has the following

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the Redirect URL.
- **Close** - You can click this button to close this window.

To assign the URL of the web page, click the 'Redirect URL' button in the Properties pane for the Work activity. In the window that pops up, enter the URL. You can either type the URL in the text editor or specify it by selecting the variable or content that points to it and clicking the Append button. Click the Update button to save the URL.

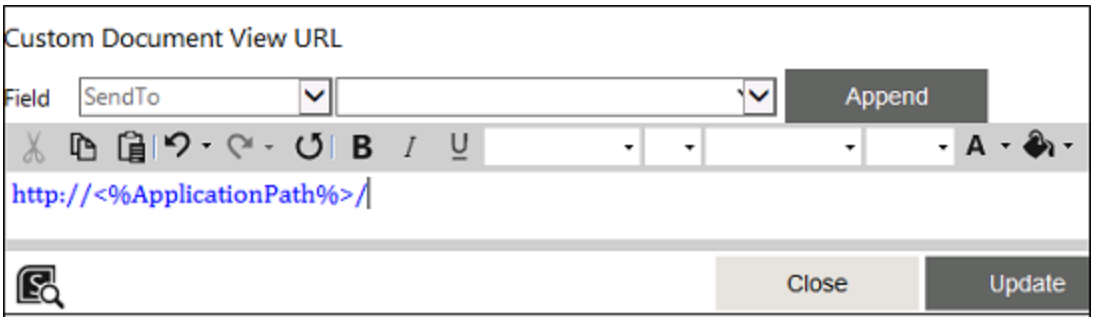
Work - Custom Document View URL

Custom Document View URL Property Window

You can specify a message Custom Document View URL for the Work activity in this property window. To open this window, click the button for the 'Custom Document View URL' property.

For more details, see Approval - Custom Document View URL

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the Custom Document View URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Custom Document View URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Custom Document View URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Custom Document View URL. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

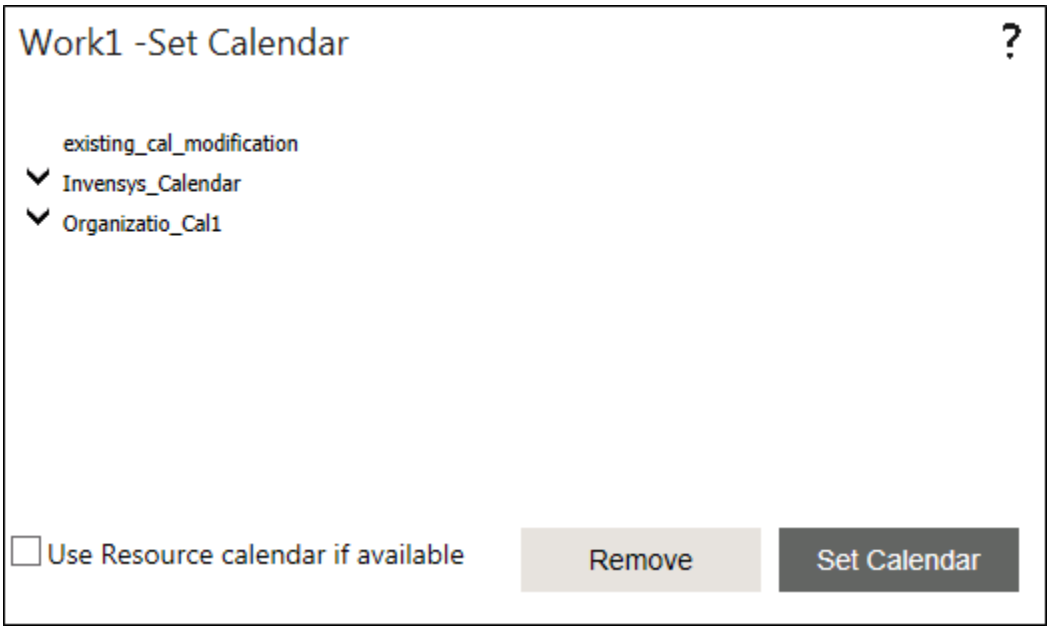
At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the Custom Document View URL.
- **Close** - You can click this button to close this window.

Work - Set Calendar for Timeout

Set Calendar for Timeout Property Window

You can set a Calendar to calculate the timeout for the Work activity in this property window. To open this window, click the button for the 'Set Calendar for Timeout' property.



Set Calendar for Timeout property window

This window lists the available calendars in a tree structure, with lower-level calendars nested within the respective top-level calendar. You can click on a calendar to select it for setting.

There are two buttons in this window.

- **No Calendar** - You can click this button to indicate that there is no calendar set for the activity.
- **Set Calendar** - You can click this button to set the selected calendar.

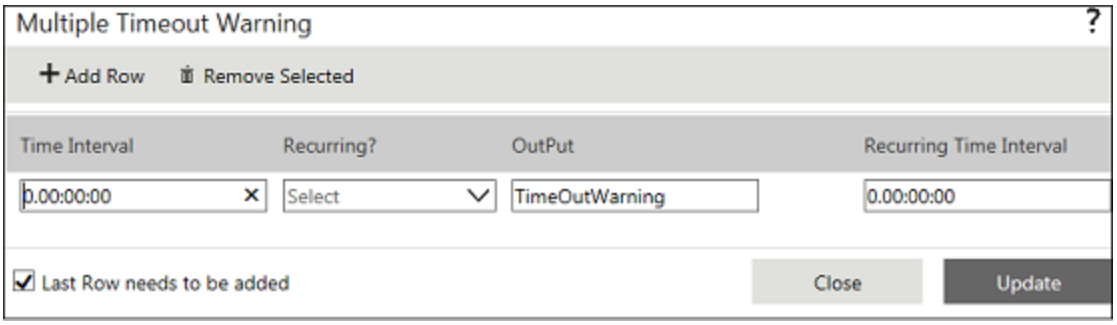
The **User Resource calendar if available** check box can be selected to enable the use of resource calendars if none are displayed in this window for selection.

Work - Multiple Timeout Warning

Multiple TimeOut Warning Property Window

You can set the Multiple Timeout Warning in this property window. To open this window, click the button for the 'Multiple TimeOut Warning' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has two tabs at the top to manage the rows of timeout settings in the grid.

- **Add Row** - You can click this tab to add the newly entered timeout setting and display a new row to the grid where you can add another timeout.
- **Remove Selected** - You can click this tab to remove the timeout setting row selected in the grid.

The grid has the following parameters that you need to specify for the new timeout setting.

- **Time Interval** - You can use this field to enter the time interval in which you want to generate a timeout warning. You must enter the time in format d.hh:mm:ss.
- **Recurring?** - You can use the drop down in this field to indicate whether the timeout is recurring or not. You can select "Yes" to set the timeout warning to recur.
- **Output** - You can enter the output message corresponding to the timeout warning here.
- **Recurring Time Interval** - You can set the time interval in which the timeout warning has to recur here. Enter the time in format d.hh:mm:ss.

There are two buttons and a check box at the bottom of the screen.

- **Update** - You can click this button to save the newly added timeout settings
- **Close** - You can click this to close the property window without saving the newly added timeout settings.
- **Last Row needs to be added** - You need to select this check box to save the last row added.

Invoke Form Activity

Activity Description

The Invoke Form Activity is basically an activity inside workflows, wherein the workflow actually invokes the form. The invoking activity is also linked to an approval process. As an example, consider the Leave Application Form. Once an employee fills out the Leave Application Form and submits it, the associated workflow is triggered and accordingly, the Form is invoked. It is sent to the reporting manager for approval. The manager also has the option to make comments and approve/reject the submitted form.

Collaboration Mode: The Invoke Form Activity is said to be in Collaboration mode when the "Individual actor's response collaboration in Output Instance XML" property is configured to collect responses from multiple users.

Note: For generating possible distinct outputs, use buttons of the following **Button Type**:
- **Submit**
- **Custom** (used for submitting the form)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Application Scenarios

Scenario 1: Opening a different form using Invoke Form activity

When a new employee joins an organization, the HR Person will fill up the employee's details and submits the form. This will go to network team to create mail id and other network requirements.

Scenario 2: Updating Form Parameters with Workflow Variable Values

When an employee submits a Leave Application form, the request will go to the Manager for approval. Manger can view the workflow variable values such as submitted user details like submitted date, user name, and user e-mail. These form parameters are updated with the workflow values while executing.

Workflow Variables for Invoke Form:

Steps to define variables for use in the Invoke Form activity

This section describes how to define variables for use in Invoke Form activities. It gives the syntax for creating the variables and typical examples for different cases.

Variables for capturing WorkItemIds

The Activity IDs for the activity can be retrieved using a variable of type array, with the following syntax:

'^'+CurrentActivity.Name+'WorkItemIds'

For example, ^ManagerFormWorkItemIds

Where 'ManagerForm' is a activity name.

Variables for capturing All Actors' Details

The details of the actor(s) to whom the approval work item gets assigned can be accessed by declaring a variable of type array with the following syntax:

'^'+CurrentActivity.Name+'ActingUsers'

For example, **^ManagerFormActingUsers**.

Where 'ManagerForm' is a activity name.

Variables for capturing Responding Actors' Details

The details of the actor(s) submitting the approval/rejection response can be accessed by declaring a variable of type resource with the following syntax:

'^'+HWSActivity.ActivityName+'RespondingUser'

For example, ^ManagerFormRespondingUser.

Where 'ManagerForm' is a activity name.

This variable stores the responded user information. If it is more than one user, it will retrieve the last user details.

Note: For single user, you can declare this variable as Resource type. If task is assigned to more than one user and to get all responded users information, then declare this variable as array.

Variables for assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

For example, if the activity name is 'ManagerForm' then declare the variable as ^ManagerForm_QueueVariable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

in the Start activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

To assign multiple queues, specify the queue-ids with semicolon (;) separator. Skelta.HWS.Queue.QueueCollection class can be used to retrieve queue information.

Variable for Queue Acting Users

You can declare an array type variable to get updated information about the name of the actor to whom an activity gets assigned from a queue.

In the Workflow Variables, define the variable in the following format:

^+CurrentActivityName+'QueueActingUsers'

Example: **^HRFormQueueActingUsers**

HRForm is the activity name.

For example the value of the user who acts on the HRForm activity will be stored in the variable. The Approval1 task will be assigned to the same user.

Variable for storing timeout warning value

Suppose if you want to store the timeout warning value for Invoke Form activity in your workflow. The naming convention for the system variable to be declared is

^<ActivityName>ResponseByWarningTimeout

The value returned by the variable will be in UTC format. For multiple user scenarios the highest timeout value calculated for a user will be available in the variable.

Note: Multiple timeout warnings cannot be tracked using this feature.

Activity Properties

The Invoke Form activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Invoke Form activity.
The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Display Name - This property specifies the name to be displayed for the activity in the Web Work Item List.

Property Type: Optional (This property need not be set if the display name is already configured in the Actions.xml. The name entered here overrides any display name that is configured for this activity in the Actions.xml file.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Delivery Channels - This property is used to specify the delivery channels to deliver this activity to the assigned actor(s). Multiple channels can be specified giving the actor(s) different options to act on the work item.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Delivery Channels property

1. To select the Delivery Channels, click the button for the 'Delivery Channels' property to open the property window.
2. In this window, the Delivery Channels that have been configured are listed on the left along with a check-box against each channel. Click and select the check-box corresponding to the channel you wish to select. You can select multiple channels by selecting the respective check-box.
3. Click Save Channels.

See Invoke Form - Delivery Channels for a detailed description of the Delivery Channels property window.

Work Item Fields - This property is used to set the Work Item Fields to be used to display application data as well as value in variables and content in the Work items list when this activity is executed.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Work Item Fields property

To select the Work Item Fields, click the button for the 'Work Item Fields' property to open the property window.

In this window, the Work Item Fields that have been set are listed in a grid. Click on a row to select the corresponding Work Item Field.

Right-click the row, and select the "SetWorkItemField" from the context menu to set it.

See Invoke Form - Work Item Fields for a detailed description of the Work Item Fields property window.

Actors

The properties in the Actors group have been described below. You can use these properties to specify the actors who will work on the activity and related properties.

Assign Actor(s) - This property is used to specify the actor(s) who has to take up the Invoke Form activity. It is tagged with the Ignore Unavailable Actors property which in turn is tagged with the Alternate Actor Allowed property. This will be explained in detail later.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any actor(s).)

Steps to set the Assign Actor(s) property

1. To specify the actors, click on the button for the 'Assign Actor(s)' property to open the property window.
2. In this window build an expression to select the actors using the resource properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, and XMLVariables), and the value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking the Add button. This will display the expression to the text

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

area.

4. You can remove a condition already added to the expression by clicking the Undo button.
5. You can also clear an existing expression if necessary by clicking the Clear button. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click the Update button to save it.

See Invoke Form - Assign Actor(s) for a detailed description of the Assign Actor(s) property window.

See the Using XML Variables section in the About the XML Variable User Interface topic for details on how to use the XMLVariables to build the expression.

Assign Queue(s) - This property specifies the queue(s) to which the Invoke Form activity is to be assigned.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any Queue(s).)

Steps to set the Assign Queue(s) property

1. To specify the queue(s), click on the button for the 'Assign Queue(s)' property to open the property window.
2. In this window select the queue(s) from the Available Queues list box. You can select multiple queues by selecting with the <Ctrl> key pressed.
3. After selecting the required queue(s) click on the [>] button to move it to the Selected Queues list box. You can click on the [>>] button to select all the queue(s) and move them to the Selected Queues list box.
4. You can remove any queue from the selected list by selecting it and clicking on the [<] button. You can remove all the queues by clicking on the [<<] button.
5. After adding the required queue(s), click the Assign button to assign the activity to the selection
6. Click the Close button to close the window without saving the selection.

See Invoke Form - Assign Queue(s) for a detailed description of the Assign Queue(s) property window.

Min. Ownerships [% or Number] - This property, which can be specified as a number or percentage, is the minimum number of actors who should take ownership of the Approval request. This property is related to the 'Timeout warning for Ownership' and 'Timeout for Ownership' properties. To illustrate, when 'Timeout for Ownership' is reached, the activity first checks if the ownership count is less than Min.Ownerships (if it is set). If it is, the work item will be removed from all Work Item lists. If the 'Min.Ownerships' has already been satisfied, then the work item will be removed from the Work Item lists of those who have not taken ownership. If the Min.Ownerships value is 0, then the activity checks if the ownership count is less than Max.Ownerships (if it is set). If it is, the work item will be removed from all Work Item lists. Else the work item will be removed from the Work Item lists of those who have not taken ownership. If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work Item lists.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Max. Ownerships [% or Number] - This property (referred to as Max.Ack), which can be specified as a number or percentage, is the maximum number of actors who should take ownership of the Approval request. If 'Timeout for Ownership' is specified and achieved, and the 'Min.Ownerships' is also satisfied, then the activity waits for action from those who have taken ownership. If the Min.Ownerships value is set to 0, then it checks if the ownership count is less than Max.Ownerships. If it is, the work item will be removed from all Work Item lists. Else the work item will be removed from the Work Item lists of those who have not taken ownership. If both Min.Ownerships and Max.Ownerships are set to 0, then there should be at least one actor taking ownership within the set time, otherwise the work item will be removed from all Work Item lists.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Enable Take Ownership? - If this property is set to Yes, then the actor is provided with a Take Ownership option in the Work items list to take ownership of the task. If you are not required Take Ownership option in the Work items list, then set this property as No.

Note: By default, this property is set as **No**.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Enable Take Ownership? property

- Select **Yes** or **No** option from the Enable Take Ownership? drop-down list.

Re-Assign Allowed? - If this property is set then the actor is provided with a Re-Assign option. Using this the approver can re-assign the work item to another resource.

Property Type: Optional (This property needs to be set only if necessary.)

Ignore Unavailable Actors? - This property provides the following options - Yes, Yes - holidays excluded and No. If this property is set to 'No', when the Approval activity is assigned to the actor's Work items list, a check is first done to see whether the actor is available. If the actor is not available then a check is made for an alternate actor assigned to the unavailable actor. Simultaneously the value of the property 'Alternate Actor Allowed' is checked. If that is also set, then the Approval work item is assigned to this actor. If this property is set to Yes - holidays excluded, the workflow engine will ignore the fact that actors are not available if the day is an organization holiday and the output Not Enough Resources will not be generated.

Property Type: Optional (This property needs to be set only if necessary.)

Alternate Actor Allowed? - This property is set to assign a particular task to an alternate actor in the absence or unavailability of any actor.

Property Type: Optional (This property needs to be set only if necessary.)

Form Information

The properties in the Form Information group have been described below. You can use these properties to specify the form details and update the form parameters.

Form Name - This property is used to select form details, by selecting the form name and version.

Or

Use **Expression Editor** to build the expression to get the form name. This is useful if you do not know the values of the properties at the time of designing the form.

Syntax for **Expression Editor** is

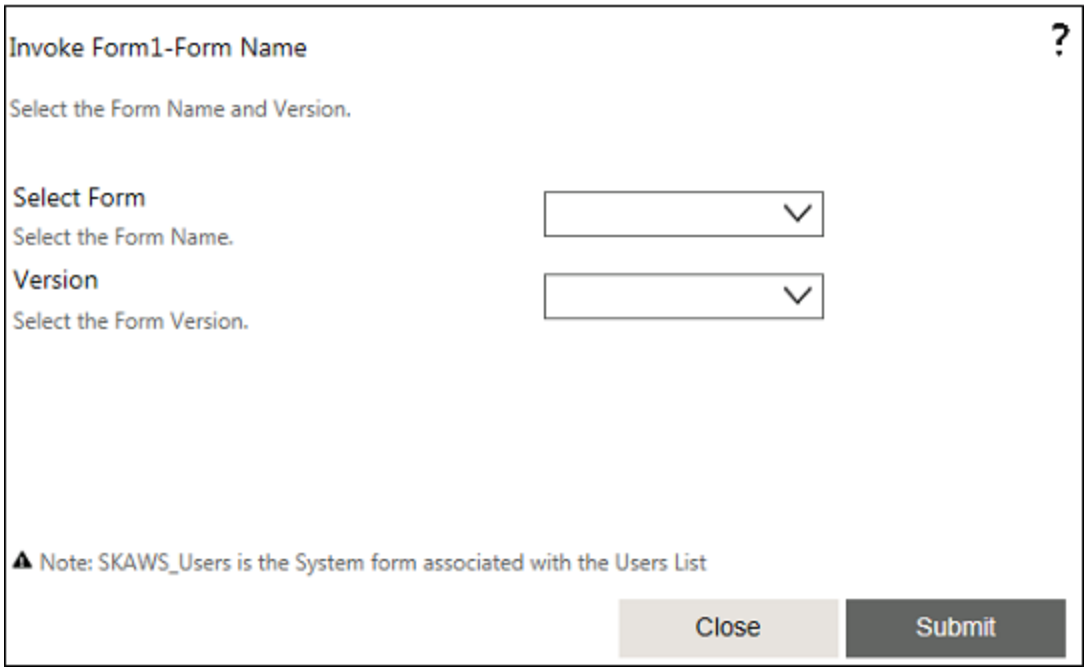
"{FormGuId,Form version}Form Name" where {FormGuId,Form version} is optional

Property Type: Mandatory

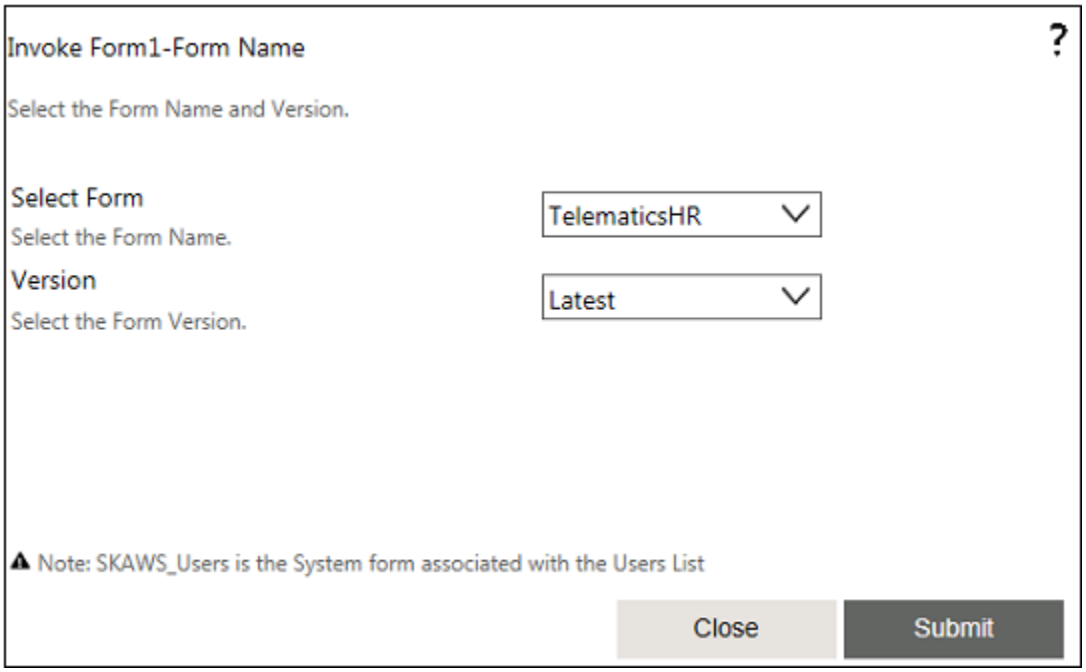
Steps to set the Form Name property

1. Click the edit icon corresponding to Form Name. The following screen is displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



2. Select a form from the Form Name drop-down list.
3. Select a form version from the Version drop-down list. By default, the latest version is displayed.



4. Click Submit, and then Close.

See Invoke Form - Form Name for a detailed description of the Form Name property window.

Input Instance XML - This property is used to display the data in the Forms which is being used in the Invoke Form activity. When the user views this Form in the work item, the data available in the selected Input Instance XML variable will be displayed, provided the form has the same schema of the selected XML Variable.

or

Use Expression Editor to build the expression to get the XML Variable name. This is useful if you do not know the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

values of the properties at the time of designing the form.

Property Type: Optional (This property needs to be set only if necessary.)

For instance:

1. To show a blank form as part of the Invoke Form activity, do not select an Input Instance XML from the Input Instance XML drop-down list. The form is then displayed with all the fields as blank.
2. If the user wants to display an existing list item data, the Input Instance XML variable selection should be of List XML type.

Update form parameters - This property is used to update the form parameter details.

Property Type: Optional (This property needs to be set only if necessary.)

Note: It is important to first select a form. For more information on selecting a form, click Steps to set the Form Name property. Once you select a form, perform the following steps to update the form parameters.

Steps to set the Update form parameters property

A parameter represents a value that is passed to a variable. The parameter value is defined when the variable is declared in the procedure. A procedure can be defined with no parameters, one parameter, or more than one. The part of the procedure definition that specifies the parameters is called the parameter list.

Here, parameters represent the form variables where the user can define while designing the form. These parameters can be used in Forms scripts, in user look up controls to build the filter expression, and as a parameter for the lookup query.

Note: Form parameters values are not stored either in form definition or in database. These values are rendered when the form is rendered.

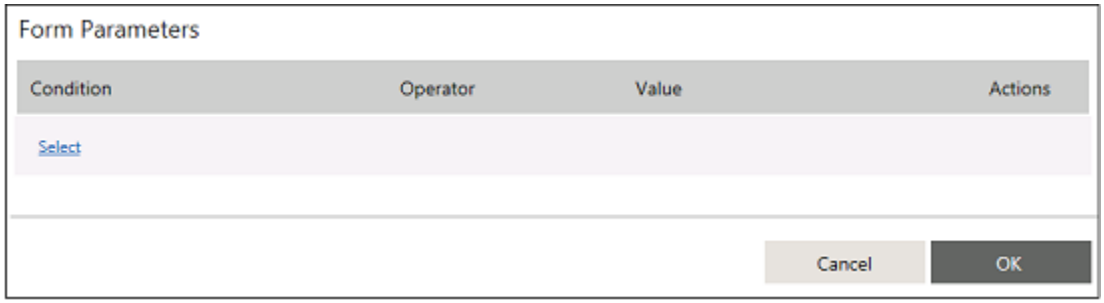
How to define parameters: To work with parameters, the form should have parameters defined. To define the form parameter, on the corresponding Forms Designer screen, click Edit properties of the form on the top. Click the Advanced tab. You can define the parameter in the Form Properties section. You can define the parameter clicking the Add Row.

How Form parameters can be used?

Form parameter can be used in Form scripts, in user look up controls to build the filter expression, and as a parameter for the lookup query. Below is the sample scenario for how parameters can be used in form scripts.

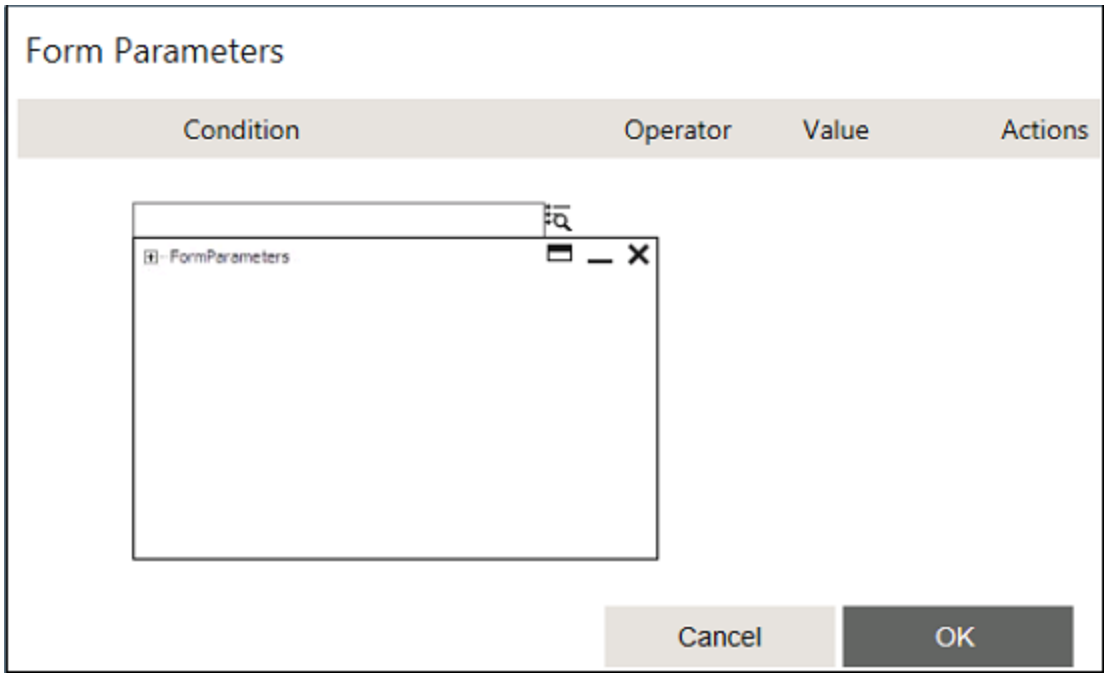
Suppose in our form we have two controls Employee id & Manager Name. Based on the Employee id entered, we can populate the Manager name. To achieve this, we need to write the script (C#) to update the Manager name value from Form parameters. This parameter value can be updated from workflow.

1. Click the edit icon corresponding to Update form parameters. The following screen is displayed.

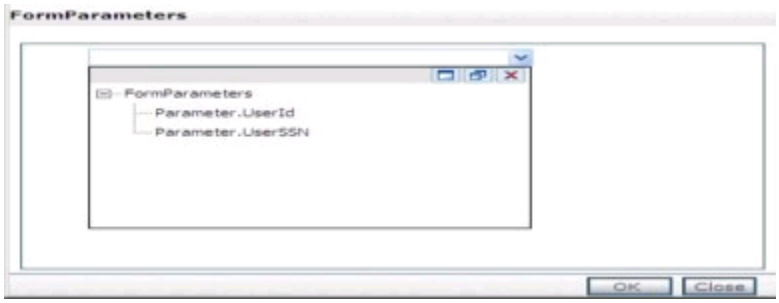


2. To select a form parameter from the drop-down list, click the drop-down. The following screen is displayed.

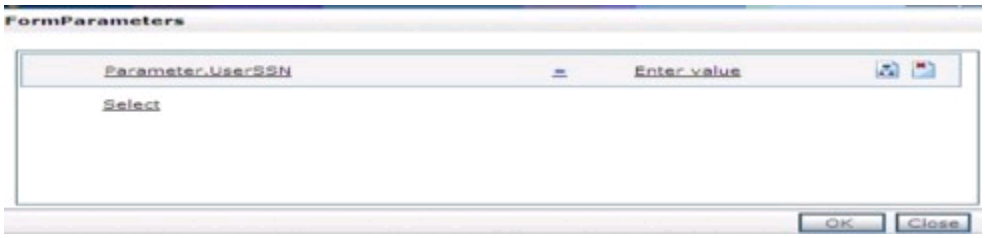
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Expand the tree. The associated parameters of the form are displayed.



4. Select the desired parameter by clicking the name. The following screen is displayed.



5. Click Enter value to display a text box. Enter the new value for the parameter in the text box.
6. Select the desired new parameter to be updated by clicking it.
7. Click OK. The new parameter is updated.

Save as Draft?

Property Type: Mandatory

Steps to set the "Save as Draft" property:

1. Click the Expression Editor to build an expression.
2. Select an option (Yes or No) from the drop-down list. Selecting Yes will allow you to save it as a draft, while

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

selecting No will not save it as a draft.

Note: This property is not available for Invoke Form activity in AVEVA Work Tasks Pro app.

Close the form on click of Save As Draft? - If this property is set as Yes, the edited form will automatically get closed on click of 'Save as Draft?' link.

Property Type: Optional (This property needs to be set only if necessary.)

Form to be opened on work item multi - select

This property enables you to specify the form which needs to be opened when multiple Invoke form work items are selected for processing. This also enables you to select multiple work items related to Invoke Form Activity and act on them from the Work items list or Inbox. The form selected opens on selection of a Work Item or multiple Work Items. On submission, the content gets updated for all the workflow instances.

You need to make sure that the form used for multi-select has the same XML node structure as used for Output Instance XML property. This ensures that the inputs provided in the form get updated to the respective workflow instances output instance XML nodes. Upon this, the update to output instance XML depends on Update output instance XML only for completion maker outputs configured for Possible Distinct Outputs property. Moreover, the behavior is overridden with copying only specific node, when the Invoke form activity is to be used in response collaboration mode .

Note: In the multi-select scenario, the inputs given to the form from the workflow using Input Instance XML property will not be populated when the form is opened.

Property Type: Optional (This property needs to be set only if necessary.)

Scenario

Consider a scenario where a manager receives many leave applications for approval in his Inbox. Instead of opening individual work items and approving or rejecting the same, the manager would like to multi select the work items and approve or reject them together. Also, rejection reasons and approval comments need to be captured for the work items.

Steps to set the Multi-select work item property:

1. Click the edit button corresponding to the Form to be opened on work item multi-select property. The **Invoke Form Multi-Select Form** opens.
2. From the **Select Form** drop-down list, select the form in which you want the multi-select property to be enabled.
3. By default, the Latest version of the form gets selected in the **Version** field.
4. Click **Submit**.

The selected form opens on selection of a work item or multiple work items through the check boxes. The check boxes are available for selection after enabling the multi – select functionality in the Inbox. On submission, the content gets updated for all the workflow instances. For more information, see the example *Selecting and Acting on Multiple Work Items Using the Enable Multi Select option* in the *Forms and Lists User Guide*.

Form to be opened to view actor responses

Property Type: Optional (This property needs to be set only if necessary.)

This property allows you to view the forms submitted to the workflow. Using this property helps you to track errors committed by users while submitting data to the workflow. If nothing is set to this property, then the Form configured for the activity will be used by default for showing the response. If your form has lot of controls and need not have to show the entire details of submission, you can override the default behavior by specifying a form for this property.

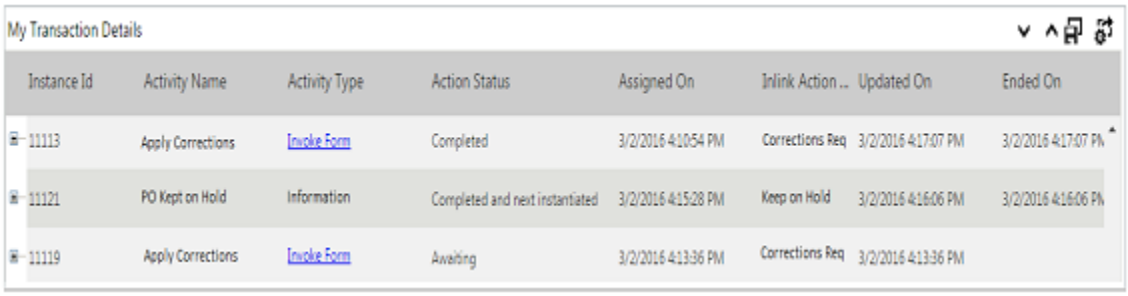
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scenario

Scenario: The Telematics Company is a BPO service provider and uses AVEVA Work Tasks workflow in their service for data entry, verification, data correction, approval etc. During data entry, a user has entered some wrong data, but when questioned denies the entry. The verifier requires substantial proof for the data entry user. Verification inputs provided by the verifier need to be viewed later, to check if the data correction done by the data entry user is as per the input provided. In case of verification by multiple users, each user's inputs need to be viewed separately. Provision to see all the forms submitted by the user is provided to the verifier.

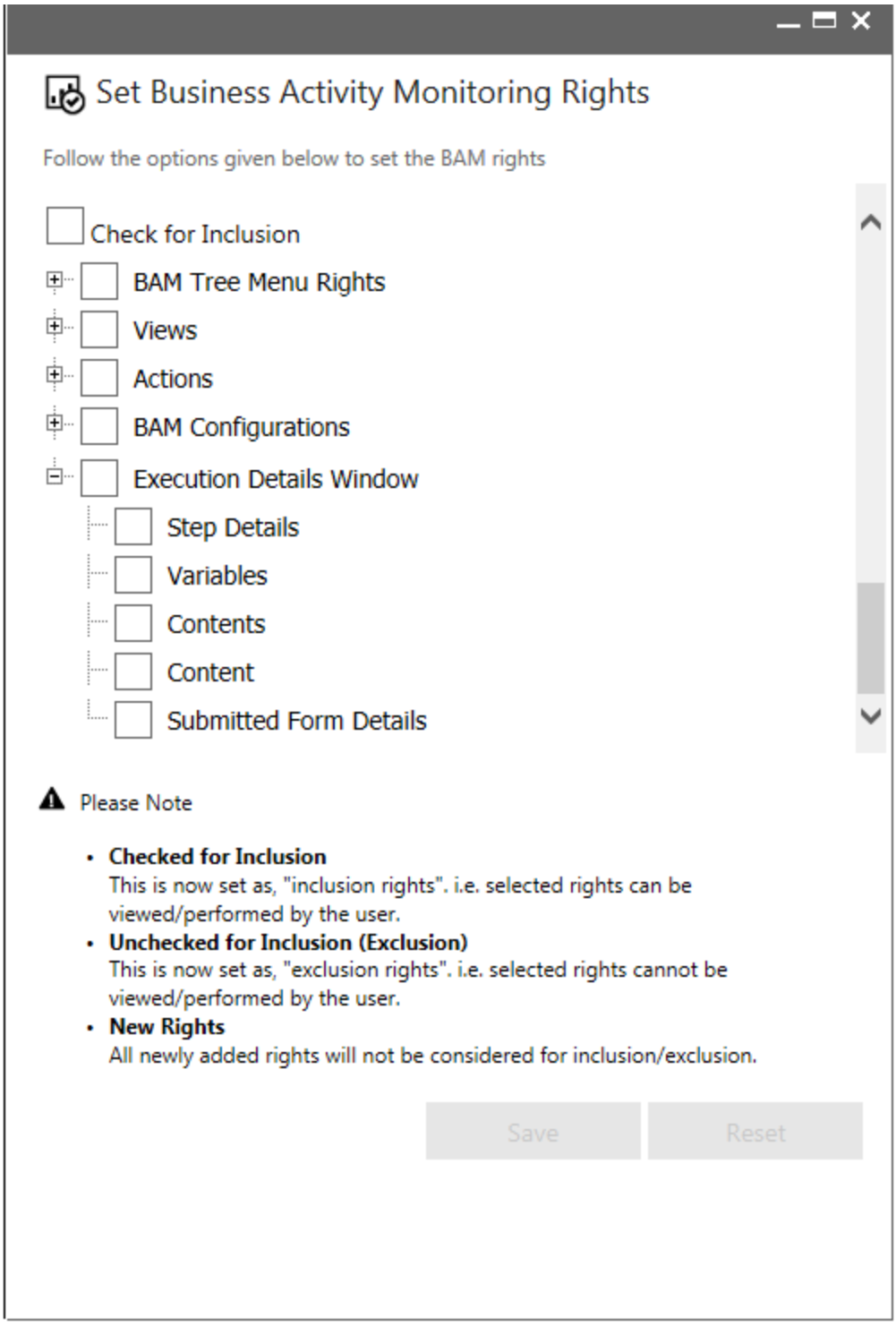
The user can view the submitted form data by clicking the Submitted Form Details tab in the activity details of Invoke Form activity under Process Execution view.

The user can also view the data from My Activities link.



Users can enable security rights to the submitted form details tab in the Set Business Activity Monitoring Rights property while generating BAM reports.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Notification Contents

The properties in the Notification Contents group have been described below. You can use these properties to specify the content used to notify actors about the activity.

Subject - This is the subject of the Approval request. The subject can be customized for each actor.

Property Type: Optional (This property need not be set. If it is not set, the default subject line from the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Actions.xml file is displayed.)

Steps to set the Subject property

1. To specify the subject, click on the button for the 'Subject' property to open the property window.
2. In this window enter the subject text directly in the text editor.
3. If you want to customize the subject you can do so using the Field controls to add resource property values. For example, to add a subject that includes the email id of the actor submitting the leave application, you can do as follows: First type in the text "Submitted By". Next select "Variable" and "Variable.SubmittedBy.Email" from the two drop downs for Field and click on the Append button. This will add the placeholder for the variable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the email id.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the subject, click the Update button to save it
7. Click the Close button to close the window without saving the subject.

See Invoke Form - Subject for a detailed description of the Subject property window.

Body - This is the actual message body of the Approval request sent to the actor(s). The message can be customized for each actor.

Property Type: Optional

Steps to set the Body property

1. To specify the message body, click on the button for the 'Body' property to open the property window.
2. In this window enter the body text directly in the text editor.
3. If you want to customize the body you can do so using the Field controls to add resource property values. For example, to add a body that includes the name of the actor submitting the leave application, you can do as follows: First type in the text "Hi, Please approve the leave application submitted by". Next select "Variable" and "Variable.SubmittedBy.name" from the two drop downs for Field and click on the Append button. This will add the placeholder for the variable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the name.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. If you want to send the form URL, click Enterprise Console Explorer. In the explorer window, select the either Local URL, Internet or Intranet URL. Click Select of the explorer window.
7. After adding the body, click the Update button to save it
8. Click the Close button to close the window without saving the body.

See Invoke Form - Body for a detailed description of the Body property window.

Template Path - This property is for specifying standard or customized mail template path. The standard template 'InvokeFormEmail.html' is available in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

[AVEVA Work Tasks Installed Path]\WorkflowElements\Default\en-US\Templates\FormEmailTemplate\ folder.

Example:

<%#WorkflowManagementInstallPath%>\WorkflowElements\Default\en-US\Templates\FormEmailTemplate\InvokeFormEmail.html

Property Type: Optional

Steps to set the Template Path property

Template Path: Locate the default template 'InvokeFormEmail.html' from the path

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Templates\FormEmailTemplate\InvokeFormEmail.html and type the same path in the Template Path property.

or

If user created a template then specify the complete path in the Template Path property.

Example: User should copy the template in Template folder which is in C drive then set the path as C:\Templates\InvokeFormEmail.html

See Invoke Form - Template Path for more information.

Close the form on click of finish? - This property is used to specify whether the form page can be closed or not when the user submits the form through email. If the value is set as No, the form page will not be closed even after the user submits the form. If the value is set as Yes, the form page will get closed after the user submits the form.

Property Type: Optional (This property needs to be set only if necessary. If it is not set, the default value is No.)

From Email Address? - This property is used to specify the email address from which notification messages will be sent for the activity.

Property Type: Optional (This property needs to be set only if necessary. If it is not set, the default value in Actions.xml will be used.)

Show Custom Document View Window - If this property is set to 'Yes', then the Custom Document View Window will be shown to actors in their Work Item Lists. Here, default value is 'No'.

For more details, see Custom Document View URL.

Property Type: Optional (This property needs to be set only if necessary.)

Priority - This property is set to indicate the priority of the approval request. Enter the values 0-33 to set as Low, 34-66 for Medium, or 67-99 for High. The default value is set to 34.

Property Type: Optional (This property needs to be set only if necessary.)

Send Notification Email - If this property is set to 'Yes', then notification emails will be sent to the actors. If it is set to 'Yes-HTML', then the emails will be sent in HTML format.

Property Type: Optional (This property needs to be set only if necessary.)

Show Email details in the Form - If this property is set to 'Yes', then the subject and body of the email will be displayed in the Work Item Invoke Form.

Property Type: Optional (This property needs to be set only if necessary.) The default value is **Yes**.

Notify if alternate actor is considered - If this property is set to 'Yes', then notification emails will be sent to the actors when an alternate actor is considered.

Property Type: Optional (This property needs to be set only if necessary.)

Redirect URL - This property is used to get the URL of the web page to redirect the client to, in the case of synchronous execution of the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Refer topic **Synchronous Execution of Forms** in the **Lists and Forms Guide** for more details.

Property Type: Optional (This property needs to be set only if necessary.)

Note: The redirection feature may not work on the Application Request Routing (ARR) server.Steps to set the Redirect URL property

1. To specify the Redirect URL, click on the button for the 'Redirect URL' property to open the property window.
2. In this window enter the Redirect URL directly in the text editor.
3. If you want to customize the Redirect URL you can do so using the Field controls to add resource property values. For example, you can add a variable or content that will hold the URL. You can do so by selecting "Variable" from the first drop down and the appropriate option from the second drop down for Field, and clicking on the Append button. This will add the placeholder for the variable in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the URL.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the Redirect URL, click the Update button to save it
7. Click the Close button to close the window without saving the Redirect URL.

See Invoke Form - Redirect URL for a detailed description of the Redirect URL property window.

Custom Document View URL - This property is used to get the URL of the page to which the custom document dlls does a post. The data posted to the page are application, workflow, executionid, executiondetailsid, and workitemid. The posted data can be retrieved as Request.Form["workitemid"].

Property Type: Optional (This property needs to be set only if necessary.)

See Invoke Form - Custom Document View URL for a detailed description of the Custom Document View URL property window.

Decision Influencers

The properties in the Decision Influencers group have been described below. You can use these properties to specify some parameters that influence the outcome of the activity.

Max Actors Required For Invoke Form Completion [% or Number] - This property is used to check what percentage or number of actors should take action for Invoke Form. Decision on the Invoke Form is taken only after the specified number or percentage of actor(s) have taken the action. The activity waits for the actors' action until the 'Timeout for activity' is reached.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Min Actors Required For Invoke Form Completion [% or Number] - This property specifies the minimum number of actors to act on the Invoke Form before the 'Timeout Warning for Activity'. If the activity limit is reached before the 'Timeout Warning for Activity' then the warning message is not shown.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Possible Distinct Outputs - This property enables you

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- To specify multiple conditions using form button and form control values and in effect set different outputs for the Invoke Form activity.
- Each output can be marked as completion maker or not. If no output is marked as completion maker, the task remains in the Inbox.
- To set whether the output instance XML is to be updated with the input data of the form for all the outputs or only for the outputs which are of type completion makers.

Property Type: Optional (This property needs to be set only if multiple outputs are required from the Invoke Form activity)

Escalation & Timeout

The properties in the Escalation & Timeout group have been described below. You can use these properties to specify escalation and timeout settings for the activity.

Set Calendar for Timeout - This property is used to specify the calendar to be used to determine activity timeouts.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to Set Calendar for Timeout

1. To specify the calendar for timeout, click on the button for the 'Set Calendar for Timeout' property to open the property window.
2. In this window select the required calendar by clicking on the calendar name in the tree structure.
3. Click the Set Calendar button to set the selected calendar.
4. You can specify that a resource calendar should be used, if available, by clicking to check the corresponding check box.

See Invoke Form - Set Calendar for Timeout for a detailed description of the Set Calendar for Timeout property window.

Minimum available time for Activity - This property is used to check the availability of an actor to whom a task must be assigned while executing the activity. The availability check for the actor is done based on the Calendar settings (Global or Resource Calendar). If the time taken to complete the task is within the value set for the **Minimum available time for Activity** property, then the task will be assigned to that actor. However, if the time taken to complete the task exceeds the value set for the **Minimum available time for Activity** property, then the task will be assigned based on the values set for the **Ignore Unavailable Actors?** and **Alternate Actor Allowed?** properties.

Property Type: Optional (This property needs to be set only if necessary.)

Timeout for Ownership - This property specifies the time until which actors are allowed to take ownership, i.e. the time until which the Approval activity will wait for actors to take ownership. If Min.Ownerships is set and is not reached, the work item is removed from all Work Item lists. If Min.Ownerships is 0, it will check if ownership count is less than Max.Ownerships. If it is, the work item is removed from all Work Item lists. Else the work item will be removed from the Work Item lists of those who have not taken ownership. The work item will remain in the Work Item list of those who have taken ownership and wait for action from them.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Timeout Warning for Ownership - This property specifies the time at which a warning is provided to an actor to take ownership of the work item assigned to him/her. If the actor does not take ownership within this time the engine is notified with a warning. Thus if 'Min.Ownerships' is set, and it is not satisfied before the given time then a warning is sent. If Min.Ownerships is 0, it will check for Max.Ownerships. If the ownership count does not reach the Max.Ownerships value within the set time, a warning is sent. If both Min.Ownerships and Max.Ownerships are not set, this property is ignored.

Property Type: Optional (This property needs to be set only if necessary.)

Understand how Ownership Properties relate to each other:

- Approval activity is assigned to the approvers in the approval list by the workflow engine. Approval activity has a default property set that at least one person has to make Take Ownership of the Approval activity.
- A drop down interface has been provided for the actor to Take Ownership. However, the actor can act on the Approval without taking ownership first, in which case ownership will be considered to be done, i.e. if the actor acts upon the Approval activity without acknowledging, then it is considered as an ownership with action taken.
- After the actor has taken ownership, a link is provided to the actor, which opens a pop up with the details of the other approvers. This is again tagged with the 'Hide Responses?' property. The link will be provided only If 'Hide Responses?' is set as 'No'. The pop up will have details like Approver name, Ownership Status, Action taken, Comments.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] then the 'Timeout Warning For Ownership' property is not considered.
- If the 'Min.Ownerships' property is set and the ownership count has reached the set limit [percentage or number] and the 'Time Out For Ownership' is also reached then the work item is removed from other approvers' Work Item list and the work item waits for action from those who have taken ownership.
- If the 'Min.Ownerships' limit is not reached for the Approval activity, and the 'Timeout Warning For Ownership' has been exceeded then the actor is notified with a warning by including an email or an information activity, with the Timeout Warning for Ownership link to it.
- If 'Timeout For Ownership' is set, if the Approval activity has not reached the minimum limit (Min.Ownerships) and the 'Time Out For Ownership' has been reached, then the work item is removed from the Work Item lists of all the approvers.
- If 'Min.Ownerships' = 0 and if 'Max.Ownerships' is set, the activity waits for actors to take ownership or action until the 'Time Out For Ownership' is reached (if this is set).
- If none of the ownership properties are set then the work item waits until the actor takes ownership or acts upon it.

Timeout Warning for Activity - This property specifies the time at which a warning is provided to an actor to take ownership of the work item assigned to him/her. If the actor does not take ownership within this time, the engine is notified with a warning. Thus if 'Min.Ownerships' is set, and it is not satisfied before the given time then a warning is sent. If Min.Ownerships is 0, it will check for Max.Ownerships. If the ownership count does not reach the Max.Ownerships value within the set time, a warning is sent. If both Min.Ownerships and Max.Ownerships are not set, this property is ignored.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Timeout for Activity - This property's value is used to decide until what time the work item should wait for user action. If properties such as Max.Ownerships, AAA, AL are set, once the property values are reached the work item is removed from the Work Item lists of all the approvers. When the timeout is reached, the following checks are made: If AL is set, when the activity count i.e. the no. of approvals or rejections (AC) < AL, then the timeout for activity is performed and the activity is removed from all Work Item lists. If AC > AL, it checks with the no. of approvals or 'decision count' (DC) and AAA. If DC >= AAA (if it is set), the work item is considered approved; otherwise, it is rejected. If AAA = 0 and if DC >= AL, the work item is considered approved; otherwise, it is rejected. If AL = 0 and AC < AAA, then the timeout for activity is performed and the work item is removed from all Work Item lists. If AL = 0 and AC > AAA, it will check with DC. If DC > AAA it is approved; otherwise, it is rejected. If AL = 0 and AAA = 0, then if DC is at least 1, then it is considered approved; otherwise, it is rejected.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Multiple TimeOut Warning - This is used to set recurring timeout warnings that will be sent to the actors who have to take action. To set multiple timeout warnings, click the 'Multiple TimeOut Warning' button in the Properties pane for the Approval activity. A new window pops up. In this window, set the interval for the recurring warning as well as the output message. You can also set the Recurring Time Interval. This can be used to set a different recurring time out after the first timeout happens. For example, if 'Time Interval' for the property is set as 7 days and 'Recurring Time Interval' is set as 2 days, then the first time out will happen after 7 days and there will be a recurring time out every second day after the seventh day (first) time out.

Property Type: Optional (This property needs to be set only if necessary. It is useful in setting escalations.)

Steps to set the Multiple TimeOut Warning property

1. To set the multiple timeout warning, click on the button for the 'Multiple TimeOut Warning' property to open the property window.
2. Enter the time interval for the timeout in the Time Interval field
3. Select "Yes" from the drop down for the Recurring? field if you want the timeout warning to recur. Select "No" otherwise.
4. Enter the output message corresponding to the timeout warning in the Output field.
5. Enter the time interval for the recurring timeout in the Recurring Time Interval field.
6. After entering the content details, click the Add Row button to add this timeout setting and display a new row for entering another timeout setting.
7. Remove rows if necessary by selecting them and clicking the Remove Selected button.
8. After specifying the timeout settings, click the Update button.

See Invoke Form - Multiple Timeout Warning for a detailed description of the Multiple TimeOut Warning property window.

Responses and Outputs

The properties in the Responses and Outputs group have been described below. You can use this property to specify the responses and outputs of the activity.

Output Instance XML - This property is used to get the data from the Forms which the user has edited from the Work item page. Depending on the type of XML variable selected in this property, the XML data is stored, provided the output instance XML has the same schema of the Form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

or

Use **Expression Editor** to build the expression to get the XML Variable name. This is useful if you do not know the values of the properties at the time of designing the form.

Property Type: Mandatory (This property must be set in order to use the data entered by the user in the workflow)

For instance:

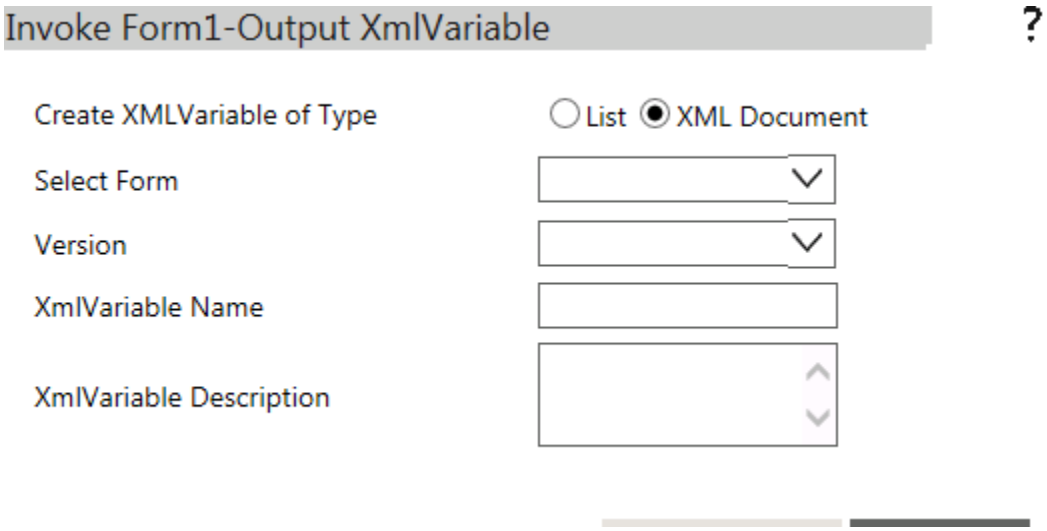
1. If the Output Instance XML is of type XML List , then the XML data received from the Form is stored in the respective List depending on the list item scope defined for the XML Variable.
   a. The list item scope should be New , if a new list item to be added with the Form data in list.
   b. The list item scope should be Current, if the list item that initiated the workflow has to be updated.
   c. The list item scope should be List Query, if a specific list item to be updated in the list.
2. If the Output Instance XML is of type XML Document, then the XML data received is stored in the workflow itself.

Display an existing list item data using Invoke Form and then store the updated Form data in a different list (destination list): An intermediate XML Variable is required for such scenarios. Use an Update Variable activity before the Invoke Form activity, to assign the Source list XML values to the Intermediate XML Variable, Then use this intermediate XML variable as Input Instance XML and destination XML List as Output Instance XML.

Note: Source, destination list, and Intermediate XML variable should have the same schema.

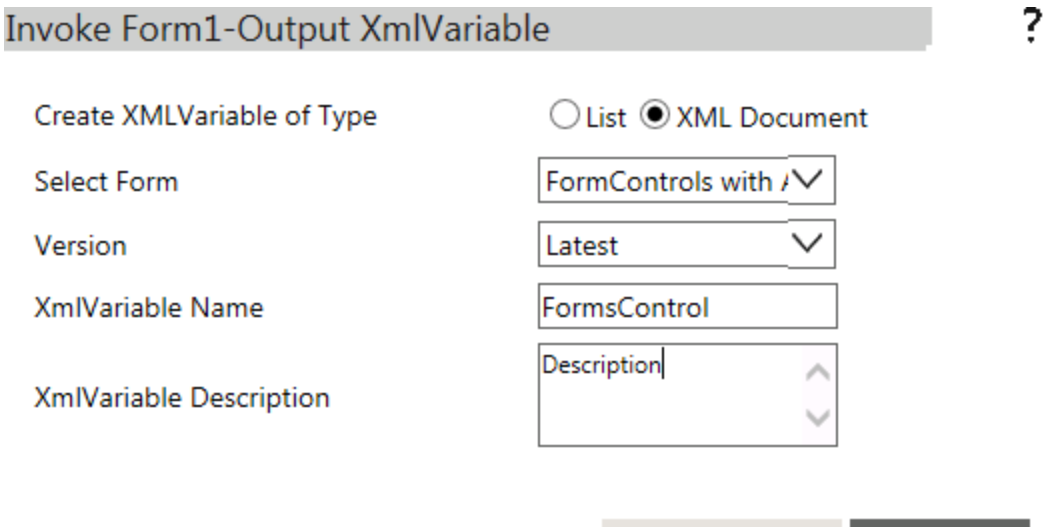
Steps to set the Output Instance XML property

1. Select an existing output XML variable from the Output Instance XML drop-down list, OR
2. Click the edit button corresponding to Create a Output Instance XML variable. The following screen is displayed.

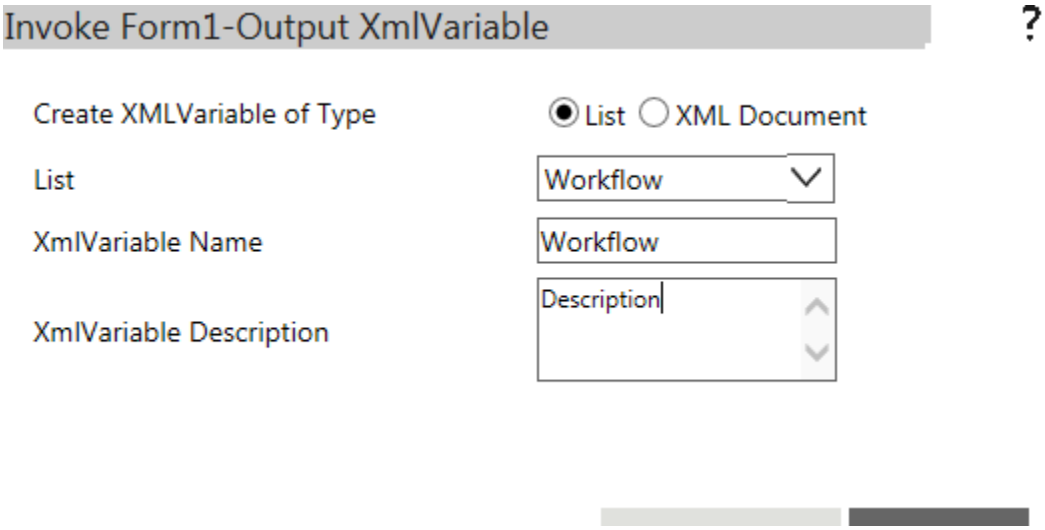


3. Select a form from the Form Name drop-down list.
4. Select a version from the Version drop-down list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Note: By default, the XML Document option is selected and the screen as shown above is displayed. If the List option is selected, the following screen is displayed. The difference, as you can notice, is that the XML Document option allows you to select a form, whereas the List option allows you to select a list.



Note: On this screen, select a list from the List drop-down.

1. Enter a name for the output XML variable in the XMLVariable Name field.
2. Enter a description for the output XML variable in the XMLVariable Description field.
3. Click Submit.
4. The output XML variable is created. The created output XML variable will be available in Start activity.

Individual actor's response collaboration in Output Instance XML

Property Type: Optional (This property needs to be set only if necessary.)

For more information, see Invoke Form - Individual Actor's Response Collaboration in Output Instance XML.

Hide Responses - If this property is set to 'No', then actors can see the responses of other actors in a group approval scenario. To view these responses that is, ownerships, approvals, or rejections, select the 'Take

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Ownership' option. This will open a new pop-up window with the responses of other actors who have taken ownership.

After the actor has taken ownership, a link is provided to the actor, which opens a pop-up window with the details of the other approvers. The link will be provided only if 'Hide Responses' property is set as 'No'. The pop-up window will have details such as Approver name, Ownership Status, Action taken, Comments.

Property Type: Optional (This property needs to be set only if necessary. It is useful in group response scenarios.)

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Miscellaneous

Redirection Enabled? - This property enables redirection. When this property is set to Yes, the workflow redirects the user to the URL in the Redirect URL property in the next activity.

Activity outputs:

The Invoke Form Activity returns the following values -

Action Completed: Displayed when the action is completed when you submit the form.

Not Enough Resources to Acknowledge: Displayed when the filter condition did not yield enough actors to whom the request can be sent.

Timeout - Acknowledge: Displayed when the specified minimum number of actors did not take ownership of the request in the given time.

Timeout - Action: Displayed when some or all of the specified actors did not perform the required activity in the given time.

Timeout Warning - Acknowledgement: Displayed when a warning is generated that the specified minimum actors have not taken ownership of the request in the given time.

Timeout Warning - Action: Displayed when a warning for timeout is generated because some or all the actors did not perform the required activity in the given time.

Note: The following outputs will return only if user assigned tasks to a queue user:

Resource Unavailable for Queue - If a queue is assigned to a task and if there are no resources available in the queue then this output will get triggered.

TimeOut for Queue Task - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting the completion of that task.

Timeout Warning -Queue - This output will get triggered when the task in the queue is not picked by any users within the queue escalation time resulting in sending a warning output. However, the task will still be sleeping.

Timeout Warning -Queue Participant - This output will get triggered when the task in the queue is picked by

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

user but still not acted on that task within the queue escalation time out settings resulting in triggering a warning output. However, the task will still be sleeping.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Note: In Inbox, navigating between items without subject is not supported for the Invoke Form activity.

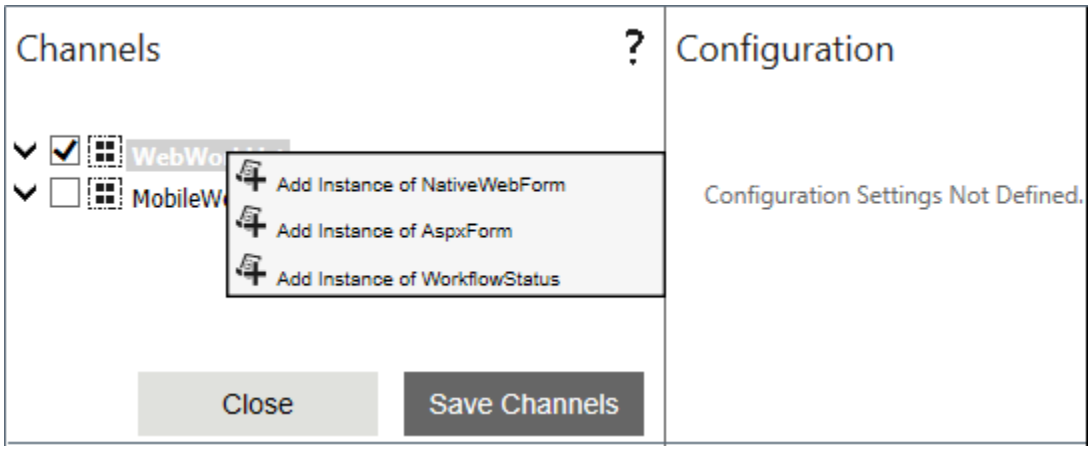
Invoke Form - Delivery Channels

Purpose

The purpose of this chapter is to describe the feature of WorkflowStatus Form of Delivery Channels property under the category Configuration of any human activity available in AVEVA Work Tasks.

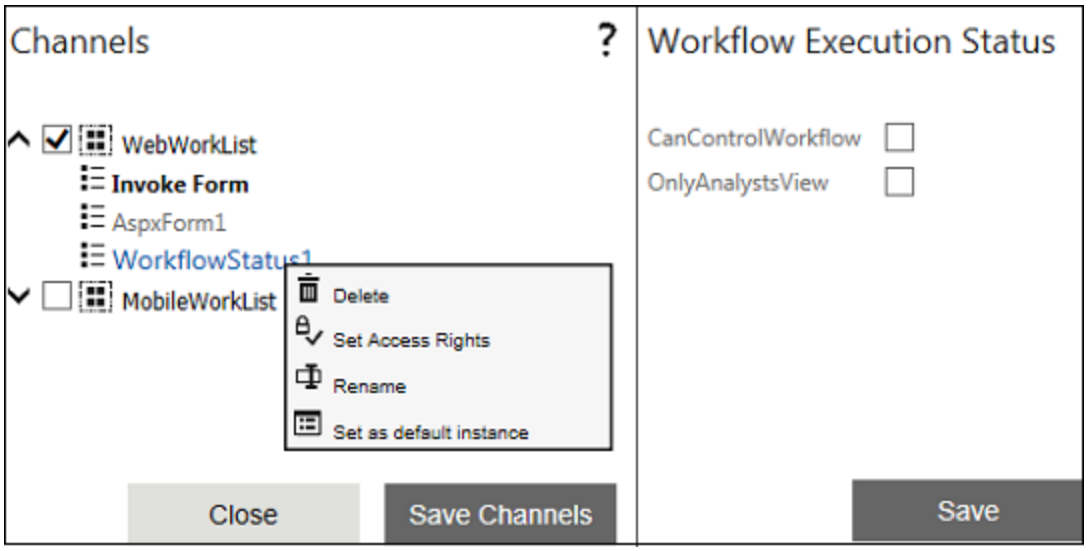
Description: This feature enables us to view the status of the workflow in the work item list in the graphical form. [Optional]

1. Select the Human Activity.
2. Click the **Delivery Channels** property. This will open a window like the one shown below.



3. Right-click on the WebWorkList to display the context menu with the **Add Instance of Workflowstatus** option.
4. Set the Properties for the Execution View Status Graphical view. Click on the **Save** button.
5. If you want to Set the workflowstatus form as the default instance you can do it by right-clicking on workflowstatus1 and selecting **the Set as default instance** option from the context menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



6. Click the **Save and Save Channels** buttons.

Output

Now, when you click on any row in the Work items list, you will find this Execution view page being displayed as the detailed view if you set it as a default instance.

Otherwise workflow status form will appear in context menu of each row in Work items list and you can open the execution view by selecting it from the context menu.

Delivery Channels - MobileWorkList

In order to deliver the work items to the mobile device, the **MobileWorkList** option in the Delivery Channel property needs to be selected. This allows mobile users to view and act upon the tasks from the Inbox that is rendered on their mobile devices.

This topic describes the steps required to configure the **MobileWorkList** option in the **Delivery Channel** property of the Invoke Form activity.

To configure 'MobileWorkList' option in the Delivery Channels Property

1. In the Process Designer, drag and drop the **Invoke Form** activity on to the design area. The activity properties pane of the selected activity appears on the right side of the design area.
2. In the **Activity Properties** pane, expand the **Configuration** node and select the **Delivery Channels** property.
3. In the **Delivery Channels** window, select the **MobileWorkList** check box in the **Channels** section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click **Save Channels**.
5. Set all the other properties and click **Publish and Test Run**.

Invoke Form - Possible Distinct Outputs

Purpose

This property of Invoke form enables you to:

1. To specify multiple conditions using form button and form control values, which in turn will set different outputs for the Invoke Form activity.
2. Each output can be marked as **Completion** maker or not. If no output is marked as completion maker, the task remains in the Inbox.
3. To set whether the output instance XML has to be updated with the input data of the form for all the outputs.

   Or

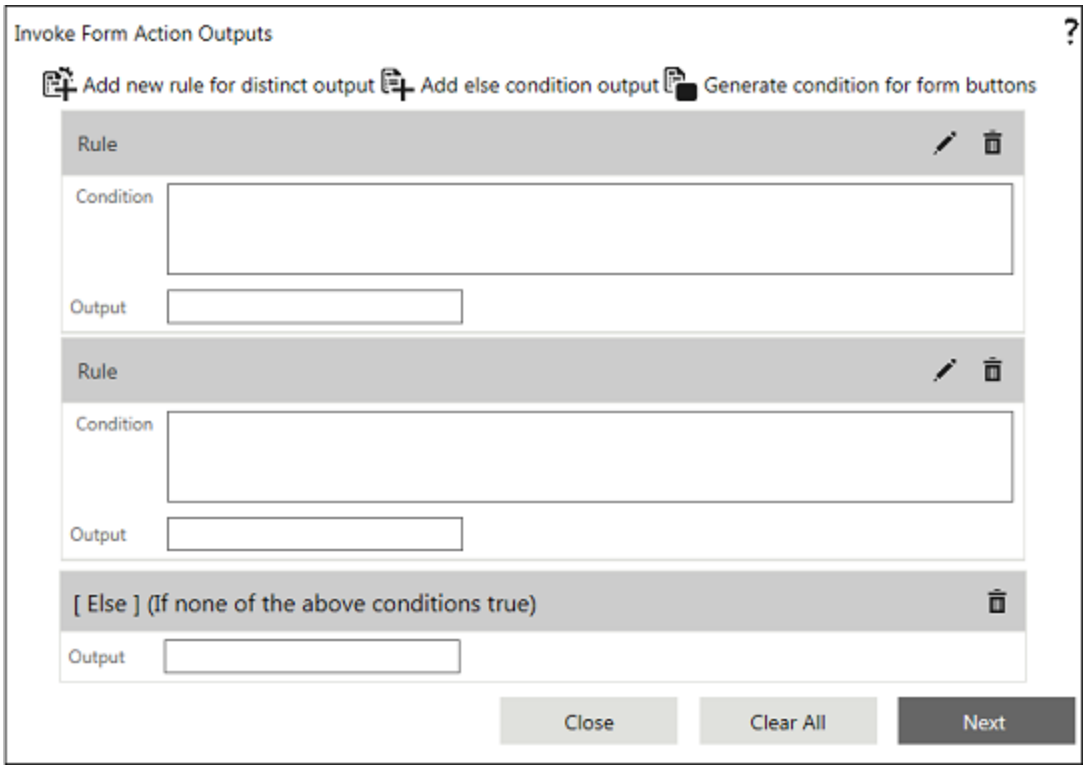
   Only for the outputs which are of type **Completion** makers.

Note: For generating possible distinct outputs, use buttons of the following **Button Type**:
- **Submit**
- **Custom** (used for submitting the form)

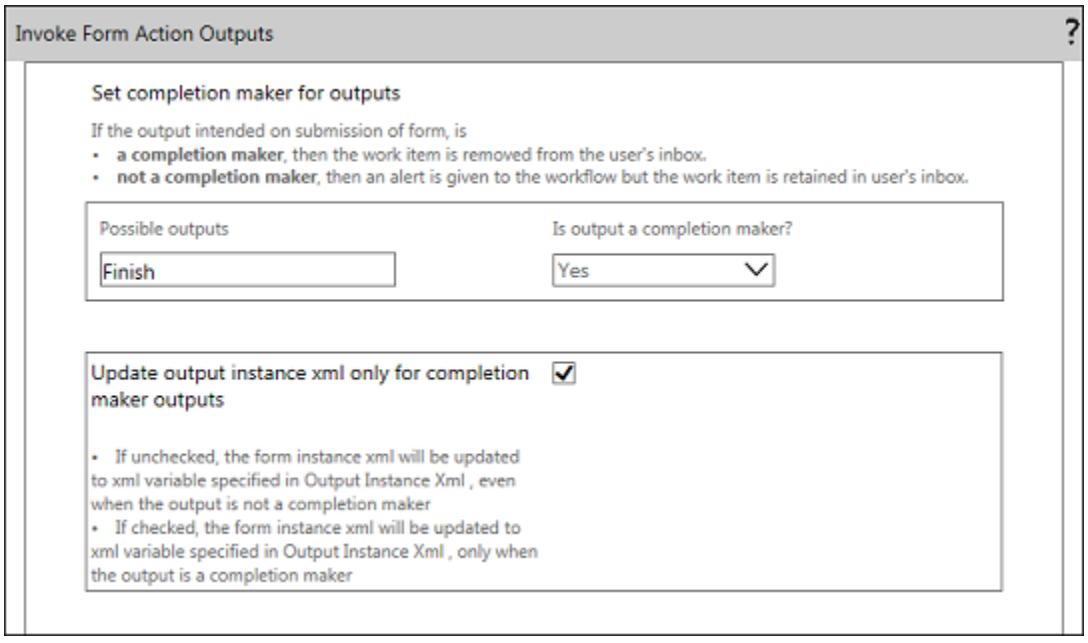
Description

- Add new rule for distinct output in the Invoke Form Action Outputs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

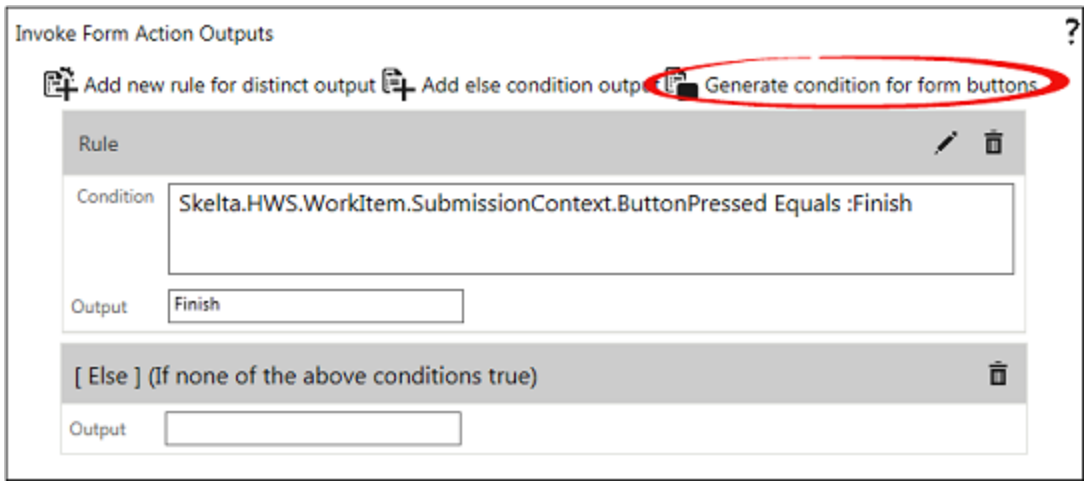


- WorkItemSubmissionContext enables you to select form submission related information. It consists of button pressed, FormsInstanceXML, RespondingUserName and WorkItemId information.

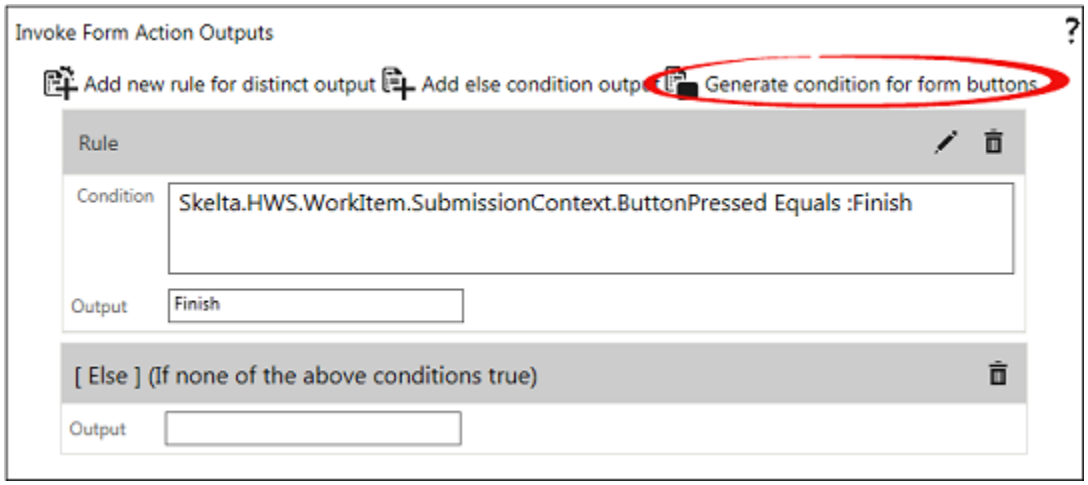


- Specify whether the output is a Completion maker or not. Also we can set whether the output instance XML has to be updated with the input data of the form for all the outputs or only for the outputs which are of type Completion makers.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



- To generate the conditions automatically for the buttons available in the form, you have the option to either use this button to generate the conditions or create the conditions on their own using form fields and button pressed text.



Invoke Form - Possible Distinct Outputs Scenario

Scenario

Telematics company allows the users in Administration Department to initiate **purchase order requisition**. The purchase orders thus requested are send for the managers approval. During the approval process the manager has options to **Approve**, **Reject**, **Apply Corrections,** or **Keep On Hold the request**.

- When the manager **approves the request**, the same is added to the AVEVA Work Tasks list that stores all the approved purchase order for further processing.
- When the manager **rejects the request**, the information is send back to the requester with valid comments for rejection.
- When the manager opts for **Apply Corrections**, the same request is send back to the requester with the comments from manager for applying required corrections before re-submitting.
- When the manager keeps the **request on hold**, the same request is retained in manager's Inbox for later

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

processing along with information being sent to the requester saying the request is kept on hold.

Requirements

1. Two users are in users list with name as Tim and John. For this scenario, John is considered as the manager of Tim.
2. Initial Form for the purchase order requisition.
3. Form to show the requested purchase order to the manager.
4. Form for the requester to make the corrections suggested by the manager.
5. List to store the approved purchase orders.
6. Approval process workflow.

Steps

1. Create Users with name John and Tim
   a. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Users**.
      The Users page appears.
   b. Add a new user with name John.
   c. Add a new user with name Tim, and set the *manager* as John.
2. Provide security rights to the users. Skip this step if security is disabled.
   a. In the **Enterprise Console**, point to the logged in user name, for example, admin, and then click **Repository Settings**.
      The Repository Settings page appears.
   b. In the **Advanced Settings** tab, click **Manage Security Group - Resource Mapping**.
      The **Security Mapping** page appears.
   c. Add a new security mapping for Tim as Contributor (from Map users tab).
   d. Add a new security mapping for John as Contributor (from Map users tab).
      Note: User search is case sensitive
3. Create Form for purchase order requisition
   a. In the **Enterprise Console**, click menu icon, and then click **Forms**.
      The Forms page appears.
   b. Create a new Workflow Initiator form and name it as *Purchase Order Requisition*.
   c. Import the form definition from the **Purchase Order Requisition_Form.xml** using the **Import** menu option in **Forms Designer.** Alternatively, use the following XML for importing:

```
<BaseForm SaveButtonText="@@Finish@@" FormTitle="" EnableActionBar="False"
ProductVersion="1" BuildVersion="50100100" Id="NF1" FormMode="NextGenMode"
XmlNodeBoundTo="SKRootDefinition" Name="Purchase Order Requisition Form"
xmlns="clr-namespace:Skelta.Forms.Core.Controls;assembly:Skelta.Forms.Core"
xmlns:L="clr-namespace:System.Collections.Generic;assembly:mscorlib"
xmlns:C="clr-
namespace:Skelta.Forms.Core.Classes;assembly:Skelta.Forms.Core"><ScriptCode><!
[CDATA[]]></ScriptCode><Controls><TextInput XmlNodeBoundTo="ReferenceNumber"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Name="Reference Number"
Id="T1"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Da
teInput XmlNodeBoundTo="OrderedDate" Name="Ordered Date" Id="D1" /><DateInput
XmlNodeBoundTo="RequiredDate" Name="Required Date" Id="D2" /><RadioButtonGroup
XmlNodeBoundTo="Priority" Name="Priority"
Id="R1"><Options><C:ColoredDictionary Key="High" Value="High" ForeColor=""
BackColor="" /><C:ColoredDictionary Key="Medium" Value="Medium" ForeColor=""
BackColor="" /><C:ColoredDictionary Key="Low" Value="Low" ForeColor=""
BackColor=""
/></Options><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></RadioButt
onGroup><TextInput XmlNodeBoundTo="Status" Name="Status" Id="T2"
Visible="False"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextI
nput><PanelForm NumberOfColumns="1" TabbingSequence="Vertical" FormTitle=""
EnableActionBar="False" ProductVersion="" BuildVersion="" Id="P1"
FormMode="NextGenMode" XmlNodeBoundTo="" Name="Order
details"><Controls><BaseForm ViewMode="Grid" FormTitle=""
EnableActionBar="False" ProductVersion="" BuildVersion="" AddPosition="Bottom"
Id="B1" DefaultRows="1" FormMode="NextGenMode" XmlNodeBoundTo="Items"
Name="Items"><Controls><TextInput XmlNodeBoundTo="ItemNo" Name="Item No"
Id="T3" /><TextInput XmlNodeBoundTo="PartNo" Name="Part No" Id="T4"
/><TextInput XmlNodeBoundTo="Description" Name="Description" Id="T5" /><Number
MaxValue="999999999999999" MinValue="-999999999999999"
XmlNodeBoundTo="UnitPrice" Name="Unit Price"
Id="N1"><DecimalDigits><![CDATA[0]]></DecimalDigits></Number><Number
MaxValue="999999999999999" MinValue="-999999999999999"
XmlNodeBoundTo="Quantity" Name="Quantity"
Id="N2"><DecimalDigits><![CDATA[0]]></DecimalDigits></Number><TextInput
XmlNodeBoundTo="ItemTotal" Name="Item Total" Id="T6"
ReadOnly="True"><OnDataChangedScript><![CDATA[control.findByXmlNode("SubTotal"
).value=control.sumByXmlNode("ItemTotal");]]></OnDataChangedScript><PropertyEx
pressions><Property Name="Value"><Code><![CDATA[var
qty=control.findByXmlNode("Quantity").value;
var price=control.findByXmlNode("UnitPrice").value;
return
qty*price;]]></Code></Property></PropertyExpressions></TextInput></Controls><O
nRowAddedScript><![CDATA[]]></OnRowAddedScript><OnRowDeletedScript><![CDATA[]]
></OnRowDeletedScript></BaseForm><PanelForm NumberOfColumns="1"
TabbingSequence="Vertical" FormTitle="" EnableActionBar="False"
ProductVersion="" BuildVersion="" Id="P2" FormMode="NextGenMode"
XmlNodeBoundTo="" Name="Estimation Details"><Controls><TextInput
XmlNodeBoundTo="SubTotal" Name="Sub Total" Id="T7"
ReadOnly="True"><OnDataChangedScript><![CDATA[var tax=(currentValue*12)/100;
control.findByXmlNode("Tax").value=tax;
var discount=(currentValue*1)/100;
control.findByXmlNode("Discount").value=discount;
control.findByXmlNode("Total").value=parseFloat(currentValue)+parseFloat(tax)-
parseFloat(discount);]]></OnDataChangedScript></TextInput><TextInput
XmlNodeBoundTo="Tax" Name="Tax" Id="T8"
ReadOnly="True"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextI
nput><Number MaxValue="999999999999999" MinValue="-999999999999999"
XmlNodeBoundTo="Discount" ReadOnly="True" Name="Discount"
Id="N3"><DecimalDigits><![CDATA[0]]></DecimalDigits><OnDataChangedScript><![CD
ATA[]]></OnDataChangedScript></Number><TextInput XmlNodeBoundTo="Total"
Name="Total" Id="T9"
ReadOnly="True"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextI
nput></Controls></PanelForm></Controls></PanelForm><PanelForm
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
NumberOfColumns="1" TabbingSequence="Vertical" FormTitle=""
EnableActionBar="False" ProductVersion="" BuildVersion="" Id="P3"
FormMode="NextGenMode" XmlNodeBoundTo="" Name="Approval details"
Visible="False"><Controls><RichTextInput XmlNodeBoundTo="Comments"
Name="Comments" Id="R2" /><UserLookUp FilterExpressions="{x:Null}"
FilterDependencyIds="{x:Null}" IsFilterExpressionSetForProperty="False"
XmlNodeBoundTo="Approver" Name="Approver" Id="U1"
/></Controls></PanelForm></Controls></BaseForm>
```

d. Click **Finish**.

e. Select the form, and then select **Associate** from the ribbon to create association with the workflow.

The Bind window appears.

f. Select the task as **Design a new Workflow**, and then click **Next**.

4. Design the **PurchaseOrderApprovalProcess** workflow.

a. Enter the title of the Workflow as **PurchaseOrderApprovalProcess**.

b. Select **Design the Workflow now**, and then click **Finish**.

c. Using the import option from tools menu, import the workflow from **'PurchaseOrderApprovalProcess.xml**'.

```
<workflow name="" version="2">
<diagramdatacollection>
<diagramdata
key="DesignerInstanceId"><![CDATA[b1d252ef2cdf4715be94978fc9f08a24]]></diagram
data>
</diagramdatacollection>
<action name="Add to Approved Purchase Orders list" type="UpdateVariableValue"
displayname="UpdateVariableValue" description="" looponoutput="">
<properties>
<property name="Set Variable">
<contentupdationfields>
<contentupdationfield name="XmlVariables.ApprovedPos.ApprovedPurchaseOrders"
operator="=" mode="ObjTree"
methodName="ff3e564e36d0411eb79c5abb50a09758"><![CDATA[XmlVariables.SFFormData
.PurchaseOrder]]></contentupdationfield>
<classname><![CDATA[84f0afb6b2dc43079e965718ac011da5]]></classname>
<jstring><![CDATA[[{"RHSMode":"ObjTree",
"LHSColumn":"XmlVariables.ApprovedPos.ApprovedPurchaseOrders",
"RHSColumn":"XmlVariables.SFFormData.PurchaseOrder",
"OPRColumn":"",
"ColumnExpression":{"DisplayExpressionString":"",
"ActualExprtessionString":"",
"MethodParameterList":"",
"HtmlVerexpString":"",
"ReferenceList":"",
"ReferencedFunctionList":"",
"NodeInfo":""}}]]]></jstring>
</contentupdationfields>
</property>
<property name="StopOnExceptions"><![CDATA[Yes]]></property>
<property name="RaiseError">
<raiseerror>
<enableraiseerrors><![CDATA[False]]></enableraiseerrors>
<value><![CDATA[]]></value>
<excludederroroutputs><![CDATA[]]></excludederroroutputs>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<includederroroutputs><![CDATA[]]></includederroroutputs>
<stoponerror><![CDATA[True]]></stoponerror>
<stoponexceptionpropertyoverrided><![CDATA[False]]></stoponexceptionpropertyov
errided>
</raiseerror>
</property>
</properties>
<visual-attributes>
<left>25</left>
<isloop>false</isloop>
<height>65</height>
<top>425</top>
<width>65</width>
</visual-attributes>
</action>
<action name="Apply Corrections" type="Invoke Form" displayname="Invoke Form"
description="" looponoutput="">
<properties>
<property name="Action Display Name"><![CDATA[]]></property>
<property name="Delivery Channels">
<propertychannel selectedchannelscount="1">
<channels>
<channel name="WebWorkList">
<config url="" />
<forms>
<form name="Invoke Form" type="InvokeWebForm">
<actors grantaccess="True" />
<roles grantaccess="True" />
</form>
</forms>
</channel>
<removedchannels list="" />
</channels>
</propertychannel>
</property>
<property name="Work Item Fields">
<Workitemfields />
</property>
<property name="To">
<resource>
<providername><![CDATA[]]></providername>
<globalfilter><![CDATA[]]></globalfilter>
<dispglobalfilter><![CDATA[]]></dispglobalfilter>
<filter><![CDATA[[Title] = [Variable.SubmittedBy.Title]]]></filter>
<dispfilter><![CDATA[[SK User Name] = [Variable.SubmittedBy.SK User
Name]]]></dispfilter>
<multipleproviderfilter><![CDATA[[Title] =
[Variable.SubmittedBy.Title]]]></multipleproviderfilter>
<multipleproviderdispfilter><![CDATA[[SK User Name] = [Variable.SubmittedBy.SK
User Name]]]></multipleproviderdispfilter>
<isuserconfiguredthroughuserlookup><![CDATA[False]]></isuserconfiguredthroughu
serlookup>
<usersconfiguredthroughuserlookup><![CDATA[]]></usersconfiguredthroughuserlook
up>
<isuserconfiguredthroughrolelookup><![CDATA[False]]></isuserconfiguredthroughr
olelookup>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<rolesconfiguredthroughrolelookup><![CDATA[]]></rolesconfiguredthroughrolelook
up>
<isuserconfiguredthroughsubmitter><![CDATA[False]]></isuserconfiguredthroughsu
bmitter>
</resource>
</property>
<property name="Assign Queue">
<queuevalue>
<value><![CDATA[]]></value>
<valueID><![CDATA[]]></valueID>
</queuevalue>
</property>
<property name="Min. Acknowledgements [% or Number]"><![CDATA[1]]></property>
<property name="Max. Acknowledgements [% or
Number]"><![CDATA[100%]]></property>
<property name="Enable TakeOwnership"><![CDATA[No]]></property>
<property name="Forwarding Allowed?"><![CDATA[No]]></property>
<property name="Ignore Unavailable Resources?"><![CDATA[No]]></property>
<property name="Alternate Resource Allowed?"><![CDATA[Yes]]></property>
<property name="FormDetails"><![CDATA[Purchase Order Apply
Corrections?0?2a1f685d-9fdd-422f-8085-04d09d7149ed?Apply
Corrections]]></property>
<property name="InputInstanceXML"><![CDATA[SFFormData]]></property>
<property name="FormParameters">
<contentupdationfields>
<classname><![CDATA[f115bff0df63429aabd8e776df6d14ee]]></classname>
<jstring><![CDATA[]]></jstring>
</contentupdationfields>
</property>
<property name="SaveAsDraft"><![CDATA[Yes]]></property>
<property name="CloseOnSaveAsDraft"><![CDATA[Yes]]></property>
<property name="MaximizeOnLoad"><![CDATA[Yes]]></property>
<property name="MultiSelectFormDetails"><![CDATA[]]></property>
<property name="ResponsesFormDetails"><![CDATA[]]></property>
<property name="Subject">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[Apply Correction for PO #
<%#XmlVariables.SFFormData.PurchaseOrder.referenceNumber%> ]]></value>
<valueHTML><![CDATA[Apply Correction for PO #
&lt;%#XmlVariables.SFFormData.PurchaseOrder.referenceNumber%&gt;
]]></valueHTML>
<classname><![CDATA[871a8cb23b4d4f0db35dbbab60df2676]]></classname>
</workflowmemovalue>
</property>
<property name="Notes">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[170d1e0d937048c0ae62e2d7263f72d1]]></classname>
</workflowmemovalue>
</property>
<property name="WorkitemDetailViewURL"><![CDATA[]]></property>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<property name="CloseOnFormSubmit"><![CDATA[No]]></property>
<property name="From Email Address"><![CDATA[admin@domain.com]]></property>
<property name="Show Custom Document View Window"><![CDATA[0]]></property>
<property name="Priority"><![CDATA[Medium]]></property>
<property name="Send Notification Email"><![CDATA[No]]></property>
<property name="Notify if alternate considered"><![CDATA[No]]></property>
<property name="Redirect URL">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[http://<%ApplicationPath%>/]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[ed8c16c8e74d415bbc19c818acbb131b]]></classname>
</workflowmemovalue>
</property>
<property name="Custom Document View URL">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[http://<%ApplicationPath%>/]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[75e012b867534fa3b091285f164965ed]]></classname>
</workflowmemovalue>
</property>
<property name="OutputInstanceXML"><![CDATA[SFFormData]]></property>
<property
name="AdvancedResponseStorage"><![CDATA[<ResponseStorage><IndRespNode></IndRes
pNode><AllRespsNode></AllRespsNode></ResponseStorage>]]></property>
<property name="Hide Responses?"><![CDATA[0]]></property>
<property name="Affirmative Answer for Approval [% or
Number]"><![CDATA[100%]]></property>
<property name="Action Limit[% or Number]"><![CDATA[100%]]></property>
<property name="Wait For All The Participants To
Act"><![CDATA[No]]></property>
<property name="Decisions">
<Decisions>
<JSDecisonString><![CDATA[]]></JSDecisonString>
<CompletionMaker><![CDATA[]]></CompletionMaker>
<AffirmativeOutput><![CDATA[]]></AffirmativeOutput>
<AffirmativePercentage><![CDATA[]]></AffirmativePercentage>
<AffirmativeNegativeOutput><![CDATA[]]></AffirmativeNegativeOutput>
<classname><![CDATA[7a046c0eceb94d92a318394e34677c7a]]></classname>
</Decisions>
</property>
<property name="Set Calendar"><![CDATA[]]></property>
<property name="Minimum Time for Action"><![CDATA[0]]></property>
<property name="Timeout warning for Acknowledgement"><![CDATA[0]]></property>
<property name="Timeout for Acknowledgement"><![CDATA[0]]></property>
<property name="Timeout warning for Action"><![CDATA[0]]></property>
<property name="Timeout for Action"><![CDATA[0]]></property>
<property name="Multiple TimeOut Warning"><![CDATA[]]></property>
<property name="Enable Redirection?"><![CDATA[No]]></property>
<property name="RaiseError">
<raiseerror>
<enableraiseerrors><![CDATA[False]]></enableraiseerrors>
<value><![CDATA[]]></value>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<excludederroroutputs><![CDATA[]]></excludederroroutputs>
<includederroroutputs><![CDATA[]]></includederroroutputs>
<stoponerror><![CDATA[True]]></stoponerror>
<stoponexceptionpropertyoverrided><![CDATA[False]]></stoponexceptionpropertyov
errided>
</raiseerror>
</property>
</properties>
<visual-attributes>
<left>500</left>
<isloop>false</isloop>
<height>65</height>
<top>75</top>
<width>65</width>
</visual-attributes>
</action>
<action name="Approval of PO" type="Invoke Form" displayname="Invoke Form"
description="" looponoutput="">
<properties>
<property name="Action Display Name"><![CDATA[]]></property>
<property name="Delivery Channels">
<propertychannel selectedchannelscount="1">
<channels>
<channel name="WebWorkList">
<config url="" />
<forms>
<form name="Invoke Form" type="InvokeWebForm">
<actors grantaccess="True" />
<roles grantaccess="True" />
</form>
</forms>
</channel>
<removedchannels list="" />
</channels>
</propertychannel>
</property>
<property name="Work Item Fields">
<Workitemfields />
</property>
<property name="To">
<resource>
<providername><![CDATA[]]></providername>
<globalfilter><![CDATA[]]></globalfilter>
<dispglobalfilter><![CDATA[]]></dispglobalfilter>
<filter><![CDATA[[Id] = [Variable.SubmittedBy.Manager]]]></filter>
<dispfilter><![CDATA[[SK User Id] = [Variable.SubmittedBy.SK
Manager]]]></dispfilter>
<multipleproviderfilter><![CDATA[[Id] =
[Variable.SubmittedBy.Manager]]]></multipleproviderfilter>
<multipleproviderdispfilter><![CDATA[[SK User Id] = [Variable.SubmittedBy.SK
Manager]]]></multipleproviderdispfilter>
<isuserconfiguredthroughuserlookup><![CDATA[False]]></isuserconfiguredthroughu
serlookup>
<usersconfiguredthroughuserlookup><![CDATA[]]></usersconfiguredthroughuserlook
up>
<isuserconfiguredthroughrolelookup><![CDATA[False]]></isuserconfiguredthroughr
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
olelookup>
<rolesconfiguredthroughrolelookup><![CDATA[]]></rolesconfiguredthroughrolelook
up>
<isuserconfiguredthroughsubmitter><![CDATA[False]]></isuserconfiguredthroughsu
bmitter>
</resource>
</property>
<property name="Assign Queue">
<queuevalue>
<value><![CDATA[]]></value>
<valueID><![CDATA[]]></valueID>
</queuevalue>
</property>
<property name="Min. Acknowledgements [% or Number]"><![CDATA[1]]></property>
<property name="Max. Acknowledgements [% or
Number]"><![CDATA[100%]]></property>
<property name="Enable TakeOwnership"><![CDATA[No]]></property>
<property name="Forwarding Allowed?"><![CDATA[No]]></property>
<property name="Ignore Unavailable Resources?"><![CDATA[No]]></property>
<property name="Alternate Resource Allowed?"><![CDATA[Yes]]></property>
<property name="FormDetails"><![CDATA[Purchase Order Approval
Form?0?6b50ad12-5000-4433-a6da-b2a886c332f5?Approval of PO]]></property>
<property name="InputInstanceXML"><![CDATA[SFFormData]]></property>
<property name="FormParameters">
<contentupdationfields>
<classname><![CDATA[2c5c84cc1427420ba2b0b617447af148]]></classname>
<jstring><![CDATA[]]></jstring>
</contentupdationfields>
</property>
<property name="SaveAsDraft"><![CDATA[No]]></property>
<property name="CloseOnSaveAsDraft"><![CDATA[Yes]]></property>
<property name="MaximizeOnLoad"><![CDATA[Yes]]></property>
<property name="MultiSelectFormDetails"><![CDATA[]]></property>
<property name="ResponsesFormDetails"><![CDATA[]]></property>
<property name="Subject">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[Approval Request of PO #
<%#XmlVariables.SFFormData.PurchaseOrder.referenceNumber%> ]]></value>
<valueHTML><![CDATA[Approval Request of PO #
&lt;%#XmlVariables.SFFormData.PurchaseOrder.referenceNumber%&gt;
]]></valueHTML>
<classname><![CDATA[2cc3474df6ab4f9da47fcde5ac5ed9dd]]></classname>
</workflowmemovalue>
</property>
<property name="Notes">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[46637cac79774826b7c3dbb40431b312]]></classname>
</workflowmemovalue>
</property>
<property name="WorkitemDetailViewURL"><![CDATA[]]></property>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<property name="CloseOnFormSubmit"><![CDATA[Yes]]></property>
<property name="From Email Address"><![CDATA[admin@domain.com]]></property>
<property name="Show Custom Document View Window"><![CDATA[0]]></property>
<property name="Priority"><![CDATA[Medium]]></property>
<property name="Send Notification Email"><![CDATA[No]]></property>
<property name="Notify if alternate considered"><![CDATA[No]]></property>
<property name="Redirect URL">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[http://<%ApplicationPath%>/]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[b6237f3653b04cbbaaac26a97177e30d]]></classname>
</workflowmemovalue>
</property>
<property name="Custom Document View URL">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[http://<%ApplicationPath%>/]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[ba62c179ab774fcfb8dbbebd52b9b52a]]></classname>
</workflowmemovalue>
</property>
<property name="OutputInstanceXML"><![CDATA[SFFormData]]></property>
<property
name="AdvancedResponseStorage"><![CDATA[<ResponseStorage><IndRespNode></IndRes
pNode><AllRespsNode></AllRespsNode></ResponseStorage>]]></property>
<property name="Hide Responses?"><![CDATA[0]]></property>
<property name="Affirmative Answer for Approval [% or
Number]"><![CDATA[100%]]></property>
<property name="Action Limit[% or Number]"><![CDATA[100%]]></property>
<property name="Wait For All The Participants To
Act"><![CDATA[No]]></property>
<property name="Decisions">
<Decisions>
<JSDecisonString><![CDATA[[{"ConditionString":"[{\"LHSBrace\":\"\",\n\"LHSColu
mn\":{\"DisplayText\":\"Skelta.HWS.WorkItem.SubmissionContext.ButtonPressed\",
\n\"Actualtext\":\"Skelta.HWS.WorkItem.SubmissionContext.ButtonPressed\",\n\"D
ataType\":\"System.String\",\n\"RefFunction\":\"\",\n\"ParamList\":\"\",\n\"Co
lumnMode\":\"default\",\n\"ColumnExpression\":{\"DisplayExpressionString\":\"\
",\n\"ActualExprtessionString\":\"\",\n\"MethodParameterList\":\"\",\n\"HtmlVe
rexpString\":\"\",\n\"ReferenceList\":\"\",\n\"ReferencedFunctionList\":\"\",\
n\"NodeInfo\":\"\"}},\n\"MHSActualText\":\"EQ\",\n\"MHSDisplayText\":\"Equals\
",\n\"RHSColumn\":{\"DisplayText\":\"Approve\",\n\"Actualtext\":\"\",\n\"DataT
ype\":\"\",\n\"RefFunction\":\"\",\n\"ParamList\":\"\",\n\"ColumnMode\":\"defa
ult\",\n\"ColumnExpression\":{\"DisplayExpressionString\":\"\",\n\"ActualExprt
essionString\":\"\",\n\"MethodParameterList\":\"\",\n\"HtmlVerexpString\":\"\"
,\n\"ReferenceList\":\"\",\n\"ReferencedFunctionList\":\"\",\n\"NodeInfo\":\"\
"},\n\"ActualText\":\"Approve\"},\n\"LHSMode\":\"default\",\n\"RHSMode\":\"def
ault\",\n\"LogicalOpr\":\"\",\n\"RHSValue\":\"Approve\",\n\"RHSBrace\":\"\",\n
\"OprValidate\":\"no\",\n\"IsRHSExists\":true}]",
"OutPut":"Approved",
"HaltExecution":false},
{"ConditionString":"[{\"LHSBrace\":\"\",\n\"LHSColumn\":{\"DisplayText\":\"Ske
lta.HWS.WorkItem.SubmissionContext.ButtonPressed\",\n\"Actualtext\":\"Skelta.H
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
WS.WorkItem.SubmissionContext.ButtonPressed\",\n\"DataType\":\"System.String\"
,\n\"RefFunction\":\"\",\n\"ParamList\":\"\",\n\"ColumnMode\":\"default\",\n\"
ColumnExpression\":{\"DisplayExpressionString\":\"\",\n\"ActualExprtessionStri
ng\":\"\",\n\"MethodParameterList\":\"\",\n\"HtmlVerexpString\":\"\",\n\"Refer
enceList\":\"\",\n\"ReferencedFunctionList\":\"\",\n\"NodeInfo\":\"\"}},\n\"MH
SActualText\":\"EQ\",\n\"MHSDisplayText\":\"Equals\",\n\"RHSColumn\":{\"Displa
yText\":\"Reject\",\n\"Actualtext\":\"\",\n\"DataType\":\"\",\n\"RefFunction\"
:\"\",\n\"ParamList\":\"\",\n\"ColumnMode\":\"default\",\n\"ColumnExpression\"
:{\"DisplayExpressionString\":\"\",\n\"ActualExprtessionString\":\"\",\n\"Meth
odParameterList\":\"\",\n\"HtmlVerexpString\":\"\",\n\"ReferenceList\":\"\",\n
\"ReferencedFunctionList\":\"\",\n\"NodeInfo\":\"\"},\n\"ActualText\":\"Reject
\"},\n\"LHSMode\":\"default\",\n\"RHSMode\":\"default\",\n\"LogicalOpr\":\"\",
\n\"RHSValue\":\"Reject\",\n\"RHSBrace\":\"\",\n\"OprValidate\":\"no\",\n\"IsR
HSExists\":true}]",
"OutPut":"Rejected",
"HaltExecution":false},
{"ConditionString":"[{\"LHSBrace\":\"\",\n\"LHSColumn\":{\"DisplayText\":\"Ske
lta.HWS.WorkItem.SubmissionContext.ButtonPressed\",\n\"Actualtext\":\"Skelta.H
WS.WorkItem.SubmissionContext.ButtonPressed\",\n\"DataType\":\"System.String\"
,\n\"RefFunction\":\"\",\n\"ParamList\":\"\",\n\"ColumnMode\":\"default\",\n\"
ColumnExpression\":{\"DisplayExpressionString\":\"\",\n\"ActualExprtessionStri
ng\":\"\",\n\"MethodParameterList\":\"\",\n\"HtmlVerexpString\":\"\",\n\"Refer
enceList\":\"\",\n\"ReferencedFunctionList\":\"\",\n\"NodeInfo\":\"\"}},\n\"MH
SActualText\":\"EQ\",\n\"MHSDisplayText\":\"Equals\",\n\"RHSColumn\":{\"Displa
yText\":\"Keep On
Hold\",\n\"Actualtext\":\"\",\n\"DataType\":\"\",\n\"RefFunction\":\"\",\n\"Pa
ramList\":\"\",\n\"ColumnMode\":\"default\",\n\"ColumnExpression\":{\"DisplayE
xpressionString\":\"\",\n\"ActualExprtessionString\":\"\",\n\"MethodParameterL
ist\":\"\",\n\"HtmlVerexpString\":\"\",\n\"ReferenceList\":\"\",\n\"Referenced
FunctionList\":\"\",\n\"NodeInfo\":\"\"},\n\"ActualText\":\"Keep On
Hold\"},\n\"LHSMode\":\"default\",\n\"RHSMode\":\"default\",\n\"LogicalOpr\":\
"\",\n\"RHSValue\":\"Keep On
Hold\",\n\"RHSBrace\":\"\",\n\"OprValidate\":\"no\",\n\"IsRHSExists\":true}]",
"OutPut":"Keep On Hold",
"HaltExecution":false},
{"ConditionString":"[{\"LHSBrace\":\"\",\n\"LHSColumn\":{\"DisplayText\":\"Ske
lta.HWS.WorkItem.SubmissionContext.ButtonPressed\",\n\"Actualtext\":\"Skelta.H
WS.WorkItem.SubmissionContext.ButtonPressed\",\n\"DataType\":\"System.String\"
,\n\"RefFunction\":\"\",\n\"ParamList\":\"\",\n\"ColumnMode\":\"default\",\n\"
ColumnExpression\":{\"DisplayExpressionString\":\"\",\n\"ActualExprtessionStri
ng\":\"\",\n\"MethodParameterList\":\"\",\n\"HtmlVerexpString\":\"\",\n\"Refer
enceList\":\"\",\n\"ReferencedFunctionList\":\"\",\n\"NodeInfo\":\"\"}},\n\"MH
SActualText\":\"EQ\",\n\"MHSDisplayText\":\"Equals\",\n\"RHSColumn\":{\"Displa
yText\":\"Apply
Corrections\",\n\"Actualtext\":\"\",\n\"DataType\":\"\",\n\"RefFunction\":\"\"
,\n\"ParamList\":\"\",\n\"ColumnMode\":\"default\",\n\"ColumnExpression\":{\"D
isplayExpressionString\":\"\",\n\"ActualExprtessionString\":\"\",\n\"MethodPar
ameterList\":\"\",\n\"HtmlVerexpString\":\"\",\n\"ReferenceList\":\"\",\n\"Ref
erencedFunctionList\":\"\",\n\"NodeInfo\":\"\"},\n\"ActualText\":\"Apply
Corrections\"},\n\"LHSMode\":\"default\",\n\"RHSMode\":\"default\",\n\"Logical
Opr\":\"\",\n\"RHSValue\":\"Apply
Corrections\",\n\"RHSBrace\":\"\",\n\"OprValidate\":\"no\",\n\"IsRHSExists\":t
rue}]",
"OutPut":"Corrections Required",
"HaltExecution":false},
]]]></JSDecisonString>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<CompletionMaker><![CDATA[<Outputs UpdateOnlyForCM='Yes'><Data
Output='Approved' status='yes'></Data><Data Output='Rejected'
status='yes'></Data><Data Output='Keep On Hold' status='no'></Data><Data
Output='Corrections Required'
status='yes'></Data></Outputs>]]></CompletionMaker>
<AffirmativeOutput><![CDATA[]]></AffirmativeOutput>
<AffirmativePercentage><![CDATA[]]></AffirmativePercentage>
<AffirmativeNegativeOutput><![CDATA[]]></AffirmativeNegativeOutput>
<classname><![CDATA[46470308f9ea447f920ebb1660c42f66]]></classname>
</Decisions>
</property>
<property name="Set Calendar"><![CDATA[]]></property>
<property name="Minimum Time for Action"><![CDATA[0]]></property>
<property name="Timeout warning for Acknowledgement"><![CDATA[0]]></property>
<property name="Timeout for Acknowledgement"><![CDATA[0]]></property>
<property name="Timeout warning for Action"><![CDATA[0]]></property>
<property name="Timeout for Action"><![CDATA[0]]></property>
<property name="Multiple TimeOut Warning"><![CDATA[]]></property>
<property name="Enable Redirection?"><![CDATA[No]]></property>
<property name="RaiseError">
<raiseerror>
<enableraiseerrors><![CDATA[False]]></enableraiseerrors>
<value><![CDATA[]]></value>
<excludederroroutputs><![CDATA[]]></excludederroroutputs>
<includederroroutputs><![CDATA[]]></includederroroutputs>
<stoponerror><![CDATA[True]]></stoponerror>
<stoponexceptionpropertyoverrided><![CDATA[False]]></stoponexceptionpropertyov
errided>
</raiseerror>
</property>
</properties>
<visual-attributes>
<left>275</left>
<isloop>false</isloop>
<height>65</height>
<top>75</top>
<width>65</width>
</visual-attributes>
</action>
<action name="Information to submitted user" type="Information"
displayname="Information" description="" looponoutput="">
<properties>
<property name="Action Display Name"><![CDATA[]]></property>
<property name="Delivery Channels">
<propertychannel selectedchannelscount="1">
<channels>
<channel name="WebWorkList">
<config url="" />
<forms>
<form name="Information Form" type="NativeWebForm">
<actors grantaccess="True" />
<roles grantaccess="True" />
</form>
</forms>
</channel>
<removedchannels list="" />
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</channels>
</propertychannel>
</property>
<property name="Work Item Fields">
<Workitemfields />
</property>
<property name="To">
<resource>
<providername><![CDATA[]]></providername>
<globalfilter><![CDATA[]]></globalfilter>
<dispglobalfilter><![CDATA[]]></dispglobalfilter>
<filter><![CDATA[[Title] = [Variable.SubmittedBy.Title]]]></filter>
<dispfilter><![CDATA[[SK User Name] = [Variable.SubmittedBy.SK User
Name]]]></dispfilter>
<multipleproviderfilter><![CDATA[[Title] =
[Variable.SubmittedBy.Title]]]></multipleproviderfilter>
<multipleproviderdispfilter><![CDATA[[SK User Name] = [Variable.SubmittedBy.SK
User Name]]]></multipleproviderdispfilter>
<isuserconfiguredthroughuserlookup><![CDATA[False]]></isuserconfiguredthroughu
serlookup>
<usersconfiguredthroughuserlookup><![CDATA[]]></usersconfiguredthroughuserlook
up>
<isuserconfiguredthroughrolelookup><![CDATA[False]]></isuserconfiguredthroughr
olelookup>
<rolesconfiguredthroughrolelookup><![CDATA[]]></rolesconfiguredthroughrolelook
up>
<isuserconfiguredthroughsubmitter><![CDATA[False]]></isuserconfiguredthroughsu
bmitter>
</resource>
</property>
<property name="Assign Queue">
<queuevalue>
<value><![CDATA[]]></value>
<valueID><![CDATA[]]></valueID>
</queuevalue>
</property>
<property name="Subject">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[PO #
<%#XmlVariables.SFFormData.PurchaseOrder.referenceNumber%> is
<%#XmlVariables.SFFormData.PurchaseOrder.status%> ]]></value>
<valueHTML><![CDATA[PO #
&lt;%#XmlVariables.SFFormData.PurchaseOrder.referenceNumber%&gt; is
&lt;%#XmlVariables.SFFormData.PurchaseOrder.status%&gt;
]]></valueHTML>
<classname><![CDATA[97414f4e2a974d56b525c96f00c86242]]></classname>
</workflowmemovalue>
</property>
<property name="Notes">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[]]></value>
<valueHTML><![CDATA[]]></valueHTML>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<classname><![CDATA[432da39ffdb04b36a888363e60303c57]]></classname>
</workflowmemovalue>
</property>
<property name="From Email Address"><![CDATA[admin@domain.com]]></property>
<property name="Show Custom Document View Window"><![CDATA[Yes]]></property>
<property name="Send Notification Email"><![CDATA[No]]></property>
<property name="Notify if alternate considered"><![CDATA[No]]></property>
<property name="Redirect URL">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[http://<%ApplicationPath%>/]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[295f69eedfb446d8be537d05c5af22ec]]></classname>
</workflowmemovalue>
</property>
<property name="Custom Document View URL">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[http://<%ApplicationPath%>/]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[11312f4a8fd6414aa35d590489e940d9]]></classname>
</workflowmemovalue>
</property>
<property name="Use html template from property"><![CDATA[No]]></property>
<property name="HTML template">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[c9d37d6adef24d88b705c2c6f407e433]]></classname>
</workflowmemovalue>
</property>
<property name="AttachReport">
<AttachmentInfo>
<SingleStringValue><![CDATA[]]></SingleStringValue>
<TypeOfAttachment><![CDATA[0]]></TypeOfAttachment>
<MailSendingOption><![CDATA[0]]></MailSendingOption>
<ClassName><![CDATA[78392f6821e54f158a67b078c8a45a49]]></ClassName>
</AttachmentInfo>
</property>
<property name="RaiseError">
<raiseerror>
<enableraiseerrors><![CDATA[False]]></enableraiseerrors>
<value><![CDATA[]]></value>
<excludederroroutputs><![CDATA[]]></excludederroroutputs>
<includederroroutputs><![CDATA[]]></includederroroutputs>
<stoponerror><![CDATA[True]]></stoponerror>
<stoponexceptionpropertyoverrided><![CDATA[False]]></stoponexceptionpropertyov
errided>
</raiseerror>
</property>
</properties>
<visual-attributes>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<left>275</left>
<isloop>false</isloop>
<height>65</height>
<top>425</top>
<width>65</width>
</visual-attributes>
</action>
<action name="PO Kept On Hold" type="Information" displayname="Information"
description="" looponoutput="">
<properties>
<property name="Action Display Name"><![CDATA[]]></property>
<property name="Delivery Channels">
<propertychannel selectedchannelscount="1">
<channels>
<channel name="WebWorkList">
<config url="" />
<forms>
<form name="Information Form" type="NativeWebForm">
<actors grantaccess="True" />
<roles grantaccess="True" />
</form>
</forms>
</channel>
<removedchannels list="" />
</channels>
</propertychannel>
</property>
<property name="Work Item Fields">
<Workitemfields />
</property>
<property name="To">
<resource>
<providername><![CDATA[]]></providername>
<globalfilter><![CDATA[]]></globalfilter>
<dispglobalfilter><![CDATA[]]></dispglobalfilter>
<filter><![CDATA[[Title] = [Variable.SubmittedBy.Title]]]></filter>
<dispfilter><![CDATA[[SK User Name] = [Variable.SubmittedBy.SK User
Name]]]></dispfilter>
<multipleproviderfilter><![CDATA[[Title] =
[Variable.SubmittedBy.Title]]]></multipleproviderfilter>
<multipleproviderdispfilter><![CDATA[[SK User Name] = [Variable.SubmittedBy.SK
User Name]]]></multipleproviderdispfilter>
<isuserconfiguredthroughuserlookup><![CDATA[False]]></isuserconfiguredthroughu
serlookup>
<usersconfiguredthroughuserlookup><![CDATA[]]></usersconfiguredthroughuserlook
up>
<isuserconfiguredthroughrolelookup><![CDATA[False]]></isuserconfiguredthroughr
olelookup>
<rolesconfiguredthroughrolelookup><![CDATA[]]></rolesconfiguredthroughrolelook
up>
<isuserconfiguredthroughsubmitter><![CDATA[False]]></isuserconfiguredthroughsu
bmitter>
</resource>
</property>
<property name="Assign Queue">
<queuevalue>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<value><![CDATA[]]></value>
<valueID><![CDATA[]]></valueID>
</queuevalue>
</property>
<property name="Subject">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[PO #
<%#XmlVariables.SFFormData.PurchaseOrder.referenceNumber%> is on hold
]]></value>
<valueHTML><![CDATA[PO #
&lt;%#XmlVariables.SFFormData.PurchaseOrder.referenceNumber%&gt; is on hold
]]></valueHTML>
<classname><![CDATA[e0612552c4314f07b8796364cccf36bf]]></classname>
</workflowmemovalue>
</property>
<property name="Notes">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[be3a8dfbd2384eb4aa04e5afb21776c0]]></classname>
</workflowmemovalue>
</property>
<property name="From Email Address"><![CDATA[admin@domain.com]]></property>
<property name="Show Custom Document View Window"><![CDATA[Yes]]></property>
<property name="Send Notification Email"><![CDATA[No]]></property>
<property name="Notify if alternate considered"><![CDATA[No]]></property>
<property name="Redirect URL">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[http://<%ApplicationPath%>/]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[606bc1696e0b498d8d047ec00f8dc608]]></classname>
</workflowmemovalue>
</property>
<property name="Custom Document View URL">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[http://<%ApplicationPath%>/]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[f0d0ae00b44f45d8a1bb882047a5a484]]></classname>
</workflowmemovalue>
</property>
<property name="Use html template from property"><![CDATA[No]]></property>
<property name="HTML template">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[ff753c37c4954ef88066eed0d8337046]]></classname>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</workflowmemovalue>
</property>
<property name="AttachReport">
<AttachmentInfo>
<SingleStringValue><![CDATA[]]></SingleStringValue>
<TypeOfAttachment><![CDATA[0]]></TypeOfAttachment>
<MailSendingOption><![CDATA[0]]></MailSendingOption>
<ClassName><![CDATA[69bc1f03f4e14f8fb8071cb196cdeaf8]]></ClassName>
</AttachmentInfo>
</property>
<property name="RaiseError">
<raiseerror>
<enableraiseerrors><![CDATA[False]]></enableraiseerrors>
<value><![CDATA[]]></value>
<excludederroroutputs><![CDATA[]]></excludederroroutputs>
<includederroroutputs><![CDATA[]]></includederroroutputs>
<stoponerror><![CDATA[True]]></stoponerror>
<stoponexceptionpropertyoverrided><![CDATA[False]]></stoponexceptionpropertyov
errided>
</raiseerror>
</property>
</properties>
<visual-attributes>
<left>525</left>
<isloop>false</isloop>
<height>65</height>
<top>225</top>
<width>65</width>
</visual-attributes>
</action>
<action name="Set status as Approved" type="UpdateVariableValue"
displayname="UpdateVariableValue" description="" looponoutput="">
<properties>
<property name="Set Variable">
<contentupdationfields>
<contentupdationfield name="XmlVariables.SFFormData.PurchaseOrder.status"
operator="=" mode="Default"
methodName="e17f1e7fe43a493b9337b30ef7223c57"><![CDATA[Approved]]></contentupd
ationfield>
<classname><![CDATA[0651c1c0e3f94057a821bafaa89945e4]]></classname>
<jstring><![CDATA[[{"RHSMode":"Default",
"LHSColumn":"XmlVariables.SFFormData.PurchaseOrder.status",
"RHSColumn":"Approved",
"OPRColumn":"",
"ColumnExpression":{"DisplayExpressionString":"",
"ActualExprtessionString":"",
"MethodParameterList":"",
"HtmlVerexpString":"",
"ReferenceList":"",
"ReferencedFunctionList":"",
"NodeInfo":""}}]]]></jstring>
</contentupdationfields>
</property>
<property name="StopOnExceptions"><![CDATA[Yes]]></property>
<property name="RaiseError">
<raiseerror>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<enableraiseerrors><![CDATA[False]]></enableraiseerrors>
<value><![CDATA[]]></value>
<excludederroroutputs><![CDATA[]]></excludederroroutputs>
<includederroroutputs><![CDATA[]]></includederroroutputs>
<stoponerror><![CDATA[True]]></stoponerror>
<stoponexceptionpropertyoverrided><![CDATA[False]]></stoponexceptionpropertyov
errided>
</raiseerror>
</property>
</properties>
<visual-attributes>
<left>125</left>
<isloop>false</isloop>
<height>65</height>
<top>275</top>
<width>65</width>
</visual-attributes>
</action>
<action name="Set status as Rejected" type="UpdateVariableValue"
displayname="UpdateVariableValue" description="" looponoutput="">
<properties>
<property name="Set Variable">
<contentupdationfields>
<contentupdationfield name="XmlVariables.SFFormData.PurchaseOrder.status"
operator="=" mode="Default"
methodName="94044eb680d64a68af4f1232d2974293"><![CDATA[Rejected]]></contentupd
ationfield>
<classname><![CDATA[a58c3f8e7be3458bb2c0d6ef11f42f0c]]></classname>
<jstring><![CDATA[[{"RHSMode":"Default",
"LHSColumn":"XmlVariables.SFFormData.PurchaseOrder.status",
"RHSColumn":"Rejected",
"OPRColumn":"",
"ColumnExpression":{"DisplayExpressionString":"",
"ActualExprtessionString":"",
"MethodParameterList":"",
"HtmlVerexpString":"",
"ReferenceList":"",
"ReferencedFunctionList":"",
"NodeInfo":""}}]]]></jstring>
</contentupdationfields>
</property>
<property name="StopOnExceptions"><![CDATA[Yes]]></property>
<property name="RaiseError">
<raiseerror>
<enableraiseerrors><![CDATA[False]]></enableraiseerrors>
<value><![CDATA[]]></value>
<excludederroroutputs><![CDATA[]]></excludederroroutputs>
<includederroroutputs><![CDATA[]]></includederroroutputs>
<stoponerror><![CDATA[True]]></stoponerror>
<stoponexceptionpropertyoverrided><![CDATA[False]]></stoponexceptionpropertyov
errided>
</raiseerror>
</property>
</properties>
<visual-attributes>
<left>275</left>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<isloop>false</isloop>
<height>65</height>
<top>275</top>
<width>65</width>
</visual-attributes>
</action>
<action name="Start" type="Start" displayname="" description=""
looponoutput="">
<properties>
<property name="Variables"><![CDATA[]]></property>
<property name="Contents"><![CDATA[]]></property>
<property name="XmlVariables">
<XmlVariables>
<XmlVariable>
<Name><![CDATA[SFFormData]]></Name>
<Description><![CDATA[Generated by Skelta Forms]]></Description>
<StorageType><![CDATA[XmlDocument]]></StorageType>
<Scope><![CDATA[global]]></Scope>
<Schema><![CDATA[<?xml version="1.0" encoding="utf-16"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="SKRootDefinition">
<xs:complexType>
<xs:sequence>
<xs:element name="referenceNumber" type="xs:string" />
<xs:element name="dateOrdered" type="xs:date" />
<xs:element name="dateRequired" type="xs:date" />
<xs:element name="priority" type="xs:string" />
<xs:element name="status" type="xs:string" />
<xs:element maxOccurs="unbounded" name="Items">
<xs:complexType>
<xs:sequence>
<xs:element name="itemNo" type="xs:string" />
<xs:element name="partNo" type="xs:string" />
<xs:element name="description" type="xs:string" />
<xs:element name="unitPrice" type="xs:double" />
<xs:element name="quantity" type="xs:double" />
<xs:element name="itemTotal" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="subtotal" type="xs:string" />
<xs:element name="taxes" type="xs:string" />
<xs:element name="discount" type="xs:double" />
<xs:element name="total" type="xs:string" />
<xs:element maxOccurs="unbounded" name="approvalDetails">
<xs:complexType>
<xs:sequence>
<xs:element name="comments" type="xs:string" />
<xs:element name="approver" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>]]></Schema>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<Xsl><![CDATA[]]></Xsl>
<Storage>
<InternalStorage><![CDATA[variables]]></InternalStorage>
</Storage>
</XmlVariable>
<XmlVariable>
<Name><![CDATA[SKEventData]]></Name>
<Description><![CDATA[]]></Description>
<StorageType><![CDATA[XmlDocument]]></StorageType>
<Scope><![CDATA[global]]></Scope>
<Schema><![CDATA[<?xml version="1.0" encoding="utf-16"?>
<xsd:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/
XMLSchema" attributeFormDefault="unqualified" elementFormDefault="qualified">
<xs:element name="SKEventData">
<xs:complexType>
<xs:sequence>
<xs:element name="EventProperties" />
<xs:element name="EventType" />
<xs:element name="EventProvider" />
<xs:element name="EventDateTimeUTC" type="xs:string" />
<xs:element name="EventDateTime" type="xs:string" />
<xs:element name="UserId" />
<xs:element name="UserName" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xsd:schema>]]></Schema>
<Xsl><![CDATA[]]></Xsl>
<Storage>
<InternalStorage><![CDATA[variables]]></InternalStorage>
</Storage>
</XmlVariable>
<XmlVariable>
<Name><![CDATA[ApprovedPos]]></Name>
<Description><![CDATA[]]></Description>
<StorageType><![CDATA[XmlSkeltaList]]></StorageType>
<Scope><![CDATA[global]]></Scope>
<Schema><![CDATA[<?xml version="1.0" encoding="utf-16"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="ApprovedPurchaseOrders">
<xs:complexType>
<xs:sequence>
<xs:element name="referenceNumber" type="xs:string" />
<xs:element name="dateOrdered" type="xs:dateTime" />
<xs:element name="dateRequired" type="xs:dateTime" />
<xs:element name="priority" type="xs:string" />
<xs:element name="status" type="xs:string" />
<xs:element maxOccurs="unbounded" name="items">
<xs:complexType>
<xs:sequence>
<xs:element maxOccurs="unbounded" name="item">
<xs:complexType>
<xs:sequence>
<xs:element name="itemNo" type="xs:string" />
<xs:element name="partNo" type="xs:string" />
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
      <xs:element name="description" type="xs:string" />
      <xs:element name="unitPrice" type="xs:decimal" />
      <xs:element name="quantity" type="xs:double" />
      <xs:element name="itemTotal" type="xs:decimal" />
      </xs:sequence>
      </xs:complexType>
      </xs:element>
      </xs:sequence>
      </xs:complexType>
      </xs:element>
      <xs:element name="subtotal" type="xs:decimal" />
      <xs:element name="taxes" type="xs:decimal" />
      <xs:element name="discount" type="xs:decimal" />
      <xs:element name="total" type="xs:decimal" />
      <xs:element maxOccurs="unbounded" name="approvalDetails">
      <xs:complexType>
      <xs:sequence>
      <xs:element name="comments" type="xs:string" />
      <xs:element name="approver" type="xs:string" />
      </xs:sequence>
      </xs:complexType>
      </xs:element>
      <xs:element name="System_x003A_Internal">
      <xs:complexType>
      <xs:sequence>
      <xs:element name="SaveAs">
      <xs:complexType>
      <xs:sequence />
      <xs:attribute name="Published" type="xs:boolean" />
      <xs:attribute name="NextVersionAsDraft" type="xs:boolean" />
      <xs:attribute name="NextVersionAsPublished" type="xs:boolean" />
      <xs:attribute name="NextMajorVersionAsDraft" type="xs:boolean" />
      <xs:attribute name="NextMajorVersionAsPublished" type="xs:boolean" />
      <xs:attribute name="NextMinorVersionAsDraft" type="xs:boolean" />
      <xs:attribute name="NextMinorVersionAsPublished" type="xs:boolean" />
      <xs:attribute name="SpecificVersionAsDraft" type="xs:string" />
      <xs:attribute name="SpecificVersionAsPublished" type="xs:string" />
      <xs:attribute name="AsFolder" type="xs:boolean" />
      </xs:complexType>
      </xs:element>
      </xs:sequence>
      </xs:complexType>
      </xs:element>
      </xs:sequence>
      </xs:complexType>
      </xs:element>
      </xs:schema>]]></Schema>
      <Xsl><![CDATA[]]></Xsl>
      <Storage>
      <ListName><![CDATA[Approved Purchase Orders]]></ListName>
      <ListId><![CDATA[06706398-376e-412a-a792-0ebe1a4a64a2]]></ListId>
      <FolderName><![CDATA[]]></FolderName>
      <FolderId><![CDATA[00000000-0000-0000-0000-000000000000]]></FolderId>
      <ListItemScope><![CDATA[New]]></ListItemScope>
      <ListItemId><![CDATA[00000000-0000-0000-0000-000000000000]]></ListItemId>
      <XmlTemplate><![CDATA[<?xml
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
version="1.0"?><ApprovedPurchaseOrders><referenceNumber /><dateOrdered
/><dateRequired /><priority /><status /><items><item><itemNo /><partNo
/><description /><unitPrice /><quantity /><itemTotal
/></item></items><subtotal /><taxes /><discount /><total
/><approvalDetails><comments /><approver
/></approvalDetails><System_x003A_Internal><SaveAs><Published
/><NextVersionAsDraft /><NextVersionAsPublished /><NextMajorVersionAsDraft
/><NextMajorVersionAsPublished /><NextMinorVersionAsDraft
/><NextMinorVersionAsPublished /><SpecificVersionAsDraft
/><SpecificVersionAsPublished /><AsFolder
/></SaveAs></System_x003A_Internal></ApprovedPurchaseOrders>]]></XmlTemplate>
<ListQuery><![CDATA[]]></ListQuery>
<ListItemVersion><![CDATA[]]></ListItemVersion>
</Storage>
</XmlVariable>
</XmlVariables>
</property>
<property name="ExecutionView"><![CDATA[]]></property>
<property name="SOAObjects">
<SOAObjects />
</property>
<property name="Initialize Variable(s)">
<contentupdationfields>
<classname><![CDATA[e81fc9ff018b4531afdf65ae2ded948e]]></classname>
<jstring><![CDATA[]]></jstring>
</contentupdationfields>
</property>
<property name="Raise Exception if Completion Maker Output is not
linked?"><![CDATA[Default [Backward Compatible]]]></property>
<property name="ExceptionTemplate">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[]]></value>
<valueHTML><![CDATA[]]></valueHTML>
</workflowmemovalue>
</property>
</properties>
<visual-attributes>
<left>100</left>
<isloop>false</isloop>
<height>65</height>
<top>75</top>
<width>65</width>
</visual-attributes>
</action>
<link from="Start" to="Approval of PO" condition="" description=""
name="Link0">
<visual-attributes>
<sourceport>r</sourceport>
<sourceportside>dynamic</sourceportside>
<sourceportindex>0</sourceportindex>
<sourceportfinalside>r</sourceportfinalside>
<sourceportfinalindex>0</sourceportfinalindex>
<sourceportfinalx>165</sourceportfinalx>
<sourceportfinaly>108</sourceportfinaly>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<targetport>l</targetport>
<targetportside>dynamic</targetportside>
<targetportindex>0</targetportindex>
<targetportfinalside>l</targetportfinalside>
<targetportfinalindex>0</targetportfinalindex>
<targetportfinalx>275</targetportfinalx>
<targetportfinaly>108</targetportfinaly>
<serializedpath>Link0;false;2;220;108;Link0,right,0,187,108,165,108,false,fals
e,NaN,NaN,!,,Link0_0,;Link0,,0,187,108,187,108,false,false,NaN,NaN,!,,Link0_1,
;Link0,right,0,253,108,187,108,false,true,220,108,!,,Link0_2,;Link0,,0,253,108
,253,108,false,false,NaN,NaN,!,,Link0_3,;Link0,right,0,275,108,253,108,true,fa
lse,NaN,NaN,!,,Link0_4,</serializedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
<link from="Approval of PO" to="Set status as Rejected" condition="Rejected"
description="" name="Link1">
<visual-attributes>
<sourceport>b</sourceport>
<sourceportside>b</sourceportside>
<sourceportindex>0</sourceportindex>
<sourceportfinalside>b</sourceportfinalside>
<sourceportfinalindex>0</sourceportfinalindex>
<sourceportfinalx>308</sourceportfinalx>
<sourceportfinaly>140</sourceportfinaly>
<targetport>t</targetport>
<targetportside>t</targetportside>
<targetportindex>0</targetportindex>
<targetportfinalside>t</targetportfinalside>
<targetportfinalindex>0</targetportfinalindex>
<targetportfinalx>308</targetportfinalx>
<targetportfinaly>275</targetportfinaly>
<serializedpath>Link1;false;2;308;207;Link1,down,0,308,167,308,140,false,false
,NaN,NaN,!,,Link1_0,;Link1,,0,308,167,308,167,false,false,NaN,NaN,!,,Link1_1,;
Link1,down,0,308,248,308,167,false,true,308,207,!,,Link1_2,;Link1,,0,308,248,3
08,248,false,false,NaN,NaN,!,,Link1_3,;Link1,down,0,308,275,308,248,true,false
,NaN,NaN,!,,Link1_4,</serializedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
<link from="Approval of PO" to="Set status as Approved" condition="Approved"
description="" name="Link2">
<visual-attributes>
<sourceport>bl</sourceport>
<sourceportside>bl</sourceportside>
<sourceportindex>1</sourceportindex>
<sourceportfinalside>bl</sourceportfinalside>
<sourceportfinalindex>1</sourceportfinalindex>
<sourceportfinalx>288</sourceportfinalx>
<sourceportfinaly>140</sourceportfinaly>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<targetport>t</targetport>
<targetportside>dynamic</targetportside>
<targetportindex>0</targetportindex>
<targetportfinalside>t</targetportfinalside>
<targetportfinalindex>0</targetportfinalindex>
<targetportfinalx>158</targetportfinalx>
<targetportfinaly>275</targetportfinaly>
<serializedpath>Link2;true;3;180;164;Link2,down,0,288,144,288,140,false,false,
288,140,!,,Link2_0,;Link2,,0,288,144,288,144,false,false,NaN,NaN,!,,Link2_1,;L
ink2,down,0,288,164,288,144,false,false,0,0,!,,Link2_2,;Link2,left,0,159,164,2
88,164,false,true,223,164,!,,Link2_3,;Link2,down,0,159,260,159,164,false,true,
159,212,!,,Link2_4,;Link2,left,0,158,260,159,260,false,false,0,0,!,,Link2_5,;L
ink2,down,0,158,272,158,260,false,false,0,0,!,,Link2_6,;Link2,,0,158,272,158,2
72,false,false,NaN,NaN,!,,Link2_7,;Link2,down,0,158,275,158,272,true,false,158
,275,!,,Link2_8,</serializedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
<link from="Set status as Rejected" to="Information to submitted user"
condition="Updated" description="" name="Link3">
<visual-attributes>
<sourceport>b</sourceport>
<sourceportside>dynamic</sourceportside>
<sourceportindex>0</sourceportindex>
<sourceportfinalside>b</sourceportfinalside>
<sourceportfinalindex>0</sourceportfinalindex>
<sourceportfinalx>308</sourceportfinalx>
<sourceportfinaly>340</sourceportfinaly>
<targetport>t</targetport>
<targetportside>t</targetportside>
<targetportindex>0</targetportindex>
<targetportfinalside>t</targetportfinalside>
<targetportfinalindex>0</targetportfinalindex>
<targetportfinalx>308</targetportfinalx>
<targetportfinaly>425</targetportfinaly>
<serializedpath>Link3;false;2;308;382;Link3,down,0,308,357,308,340,false,false
,NaN,NaN,!,,Link3_0,;Link3,,0,308,357,308,357,false,false,NaN,NaN,!,,Link3_1,;
Link3,down,0,308,408,308,357,false,true,308,382,!,,Link3_2,;Link3,,0,308,408,3
08,408,false,false,NaN,NaN,!,,Link3_3,;Link3,down,0,308,425,308,408,true,false
,NaN,NaN,!,,Link3_4,</serializedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
<link from="Set status as Approved" to="Information to submitted user"
condition="Updated" description="" name="Link4">
<visual-attributes>
<sourceport>b</sourceport>
<sourceportside>dynamic</sourceportside>
<sourceportindex>0</sourceportindex>
<sourceportfinalside>b</sourceportfinalside>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<sourceportfinalindex>0</sourceportfinalindex>
<sourceportfinalx>158</sourceportfinalx>
<sourceportfinaly>340</sourceportfinaly>
<targetport>l</targetport>
<targetportside>l</targetportside>
<targetportindex>0</targetportindex>
<targetportfinalside>l</targetportfinalside>
<targetportfinalindex>0</targetportfinalindex>
<targetportfinalx>275</targetportfinalx>
<targetportfinaly>458</targetportfinaly>
<serializedpath>Link4;true;2;158;457;Link4,down,0,158,363,158,340,false,false,
NaN,NaN,!,,Link4_0,;Link4,,0,158,363,158,363,false,false,NaN,NaN,!,,Link4_1,;L
ink4,down,0,158,458,158,363,false,true,158,410,!,,Link4_2,;Link4,right,0,252,4
58,158,458,false,true,205,458,!,,Link4_3,;Link4,,0,252,458,252,458,false,false
,NaN,NaN,!,,Link4_4,;Link4,right,0,275,458,252,458,true,false,NaN,NaN,!,,Link4
_5,</serializedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
<link from="Approval of PO" to="PO Kept On Hold" condition="Keep On Hold"
description="" name="Link5">
<visual-attributes>
<sourceport>rb</sourceport>
<sourceportside>rb</sourceportside>
<sourceportindex>1</sourceportindex>
<sourceportfinalside>rb</sourceportfinalside>
<sourceportfinalindex>1</sourceportfinalindex>
<sourceportfinalx>340</sourceportfinalx>
<sourceportfinaly>128</sourceportfinaly>
<targetport>tl</targetport>
<targetportside>tl</targetportside>
<targetportindex>1</targetportindex>
<targetportfinalside>tl</targetportfinalside>
<targetportfinalindex>1</targetportfinalindex>
<targetportfinalx>538</targetportfinalx>
<targetportfinaly>225</targetportfinaly>
<serializedpath>Link5;false;4;405;210;Link5,right,0,342,128,340,128,false,fals
e,NaN,NaN,!,,Link5_0,;Link5,,0,342,128,342,128,false,false,NaN,NaN,!,,Link5_1,
;Link5,right,0,350,128,342,128,false,false,0,0,!,,Link5_2,;Link5,down,0,350,21
0,350,128,false,true,350,169,!,,Link5_3,;Link5,right,0,538,210,350,210,false,t
rue,444,210,!,,Link5_4,;Link5,down,0,538,222,538,210,false,false,0,0,!,,Link5_
5,;Link5,,0,538,222,538,222,false,false,NaN,NaN,!,,Link5_6,;Link5,down,0,538,2
25,538,222,true,false,NaN,NaN,!,,Link5_7,</serializedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
<link from="Approval of PO" to="Apply Corrections" condition="Corrections
Required" description="" name="Link6">
<visual-attributes>
<sourceport>r</sourceport>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

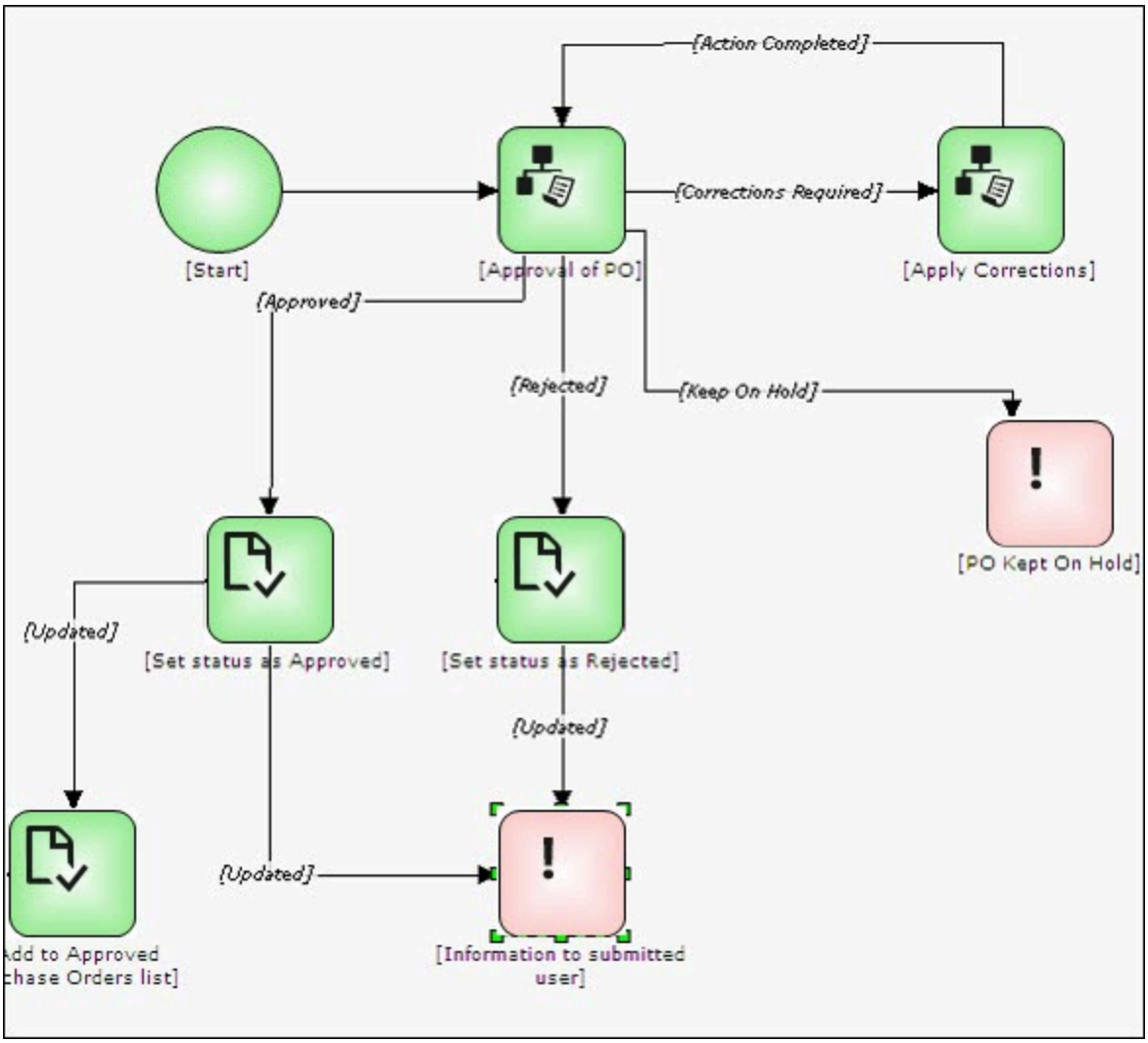
```
<sourceportside>dynamic</sourceportside>
<sourceportindex>0</sourceportindex>
<sourceportfinalside>r</sourceportfinalside>
<sourceportfinalindex>0</sourceportfinalindex>
<sourceportfinalx>340</sourceportfinalx>
<sourceportfinaly>108</sourceportfinaly>
<targetport>l</targetport>
<targetportside>dynamic</targetportside>
<targetportindex>0</targetportindex>
<targetportfinalside>l</targetportfinalside>
<targetportfinalindex>0</targetportfinalindex>
<targetportfinalx>500</targetportfinalx>
<targetportfinaly>108</targetportfinaly>
<serializedpath>Link6;false;2;420;108;Link6,right,0,372,108,340,108,false,fals
e,NaN,NaN,!,,Link6_0,;Link6,,0,372,108,372,108,false,false,NaN,NaN,!,,Link6_1,
;Link6,right,0,468,108,372,108,false,true,420,108,!,,Link6_2,;Link6,,0,468,108
,468,108,false,false,NaN,NaN,!,,Link6_3,;Link6,right,0,500,108,468,108,true,fa
lse,NaN,NaN,!,,Link6_4,</serializedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
<link from="Apply Corrections" to="Approval of PO" condition="Action
Completed" description="" name="Link7">
<visual-attributes>
<sourceport>t</sourceport>
<sourceportside>t</sourceportside>
<sourceportindex>0</sourceportindex>
<sourceportfinalside>t</sourceportfinalside>
<sourceportfinalindex>0</sourceportfinalindex>
<sourceportfinalx>533</sourceportfinalx>
<sourceportfinaly>75</sourceportfinaly>
<targetport>t</targetport>
<targetportside>t</targetportside>
<targetportindex>0</targetportindex>
<targetportfinalside>t</targetportfinalside>
<targetportfinalindex>0</targetportfinalindex>
<targetportfinalx>308</targetportfinalx>
<targetportfinaly>75</targetportfinaly>
<serializedpath>Link7;true;0;421;32;Link7,up,0,533,67,533,75,false,false,NaN,N
aN,!,,Link7_0,;Link7,,0,533,67,533,67,false,false,NaN,NaN,!,,Link7_1,;Link7,up
,0,533,32,533,67,false,true,533,49,!,,Link7_2,;Link7,left,0,360,32,533,32,fals
e,true,446,32,!,,Link7_3,;Link7,down,0,360,33,360,32,false,false,0,0,!,,Link7_
4,;Link7,left,0,308,33,360,33,false,true,334,33,!,,Link7_5,;Link7,down,0,308,6
7,308,33,false,true,308,50,!,,Link7_6,;Link7,,0,308,67,308,67,false,false,NaN,
NaN,!,,Link7_7,;Link7,down,0,308,75,308,67,true,false,NaN,NaN,!,,Link7_8,</ser
ializedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
<link from="Set status as Approved" to="Add to Approved Purchase Orders list"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
      condition="Updated" description="" name="Link8">
      <visual-attributes>
      <sourceport>l</sourceport>
      <sourceportside>l</sourceportside>
      <sourceportindex>0</sourceportindex>
      <sourceportfinalside>l</sourceportfinalside>
      <sourceportfinalindex>0</sourceportfinalindex>
      <sourceportfinalx>125</sourceportfinalx>
      <sourceportfinaly>308</sourceportfinaly>
      <targetport>t</targetport>
      <targetportside>t</targetportside>
      <targetportindex>0</targetportindex>
      <targetportfinalside>t</targetportfinalside>
      <targetportfinalindex>0</targetportfinalindex>
      <targetportfinalx>58</targetportfinalx>
      <targetportfinaly>425</targetportfinaly>
      <serializedpath>Link8;true;3;58;333;Link8,left,0,112,308,125,308,false,false,N
      aN,NaN,!,,Link8_0,;Link8,,0,112,308,112,308,false,false,NaN,NaN,!,,Link8_1,;Li
      nk8,left,0,58,308,112,308,false,true,85,308,!,,Link8_2,;Link8,down,0,58,402,58
      ,308,false,true,58,355,!,,Link8_3,;Link8,,0,58,402,58,402,false,false,NaN,NaN,
      !,,Link8_4,;Link8,down,0,58,425,58,402,true,false,NaN,NaN,!,,Link8_5,</seriali
      zedpath>
      <thickness>1</thickness>
      <color>#000000</color>
      <linetype />
      <islocked>false</islocked>
      </visual-attributes>
      </link>
      </workflow>
```

The Workflow looks as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



d. Publish the Workflow.

5. Create **Purchase Order Approval Form** for the approver.

   a. In the **Enterprise Console**, click ☰ menu icon, and then click **Forms**.

      The Forms page appears.

   b. Create a new Form named **Purchase Order Approval Form** which will be used for the approvals.

   c. Import the form definition from '**Purchase Order Approval_Form.xml**' using the **Import** menu option in **Forms Designer**, or use the following xml for importing and publish the form.

```
<BaseForm SaveButtonText="@@Finish@@" FormTitle="Purchase Order Approval Form"
ShowSubmitButton="False" ProductVersion="1" BuildVersion="40107100" Id="NF1"
FormMode="NextGenMode" XmlNodeBoundTo="PurchaseOrder" Name="Purchase Order
Requisition Form" Description="Purchase Order Requisition" xmlns="clr-
namespace:Skelta.Forms.Core.Controls;assembly:Skelta.Forms.Core" xmlns:L="clr-
namespace:System.Collections.Generic;assembly:mscorlib" xmlns:C="clr-
namespace:Skelta.Forms.Core.Classes;assembly:Skelta.Forms.Core"><ScriptCode><!
[CDATA[]]></ScriptCode><Controls><TextInput XmlNodeBoundTo="referenceNumber"
Name="Reference Number"
Id="T1"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Da
teInput XmlNodeBoundTo="dateOrdered" Name="Ordered Date"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Id="D1"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></DateInput><Da
teInput XmlNodeBoundTo="dateRequired" Name="Required Date"
Id="D2"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></DateInput><Dr
opDownChoice DefaultText="High" XmlNodeBoundTo="priority" Enabled="False"
Name="Priority" Id="D3"><Options><C:ColoredDictionary Key="High" Value="High"
ForeColor="#d33333" BackColor="" /><C:ColoredDictionary Key="Medium"
Value="Medium" ForeColor="#3427a9" BackColor="" /><C:ColoredDictionary
Key="Low" Value="Low" ForeColor="" BackColor=""
/></Options><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></DropDownC
hoice><TextInput XmlNodeBoundTo="status" Name="Status" Id="T2"
Visible="False"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextI
nput><PanelForm NumberOfColumns="1" FormTitle="" ProductVersion=""
BuildVersion="" Id="P1" FormMode="NextGenMode" XmlNodeBoundTo="items"
Name="Order details"><Controls><BaseForm FormTitle="" ProductVersion=""
BuildVersion="" Id="B1" FormMode="NextGenMode" XmlNodeBoundTo="item"
Name="Items" ViewMode="Grid" AddPosition="Bottom"
DefaultRows="1"><Controls><TextInput XmlNodeBoundTo="itemNo" Name="Item No"
Id="T5"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Te
xtInput XmlNodeBoundTo="partNo" Name="Part No"
Id="T4"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Te
xtInput XmlNodeBoundTo="description" Name="Description"
Id="T3"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Nu
mber XmlNodeBoundTo="unitPrice" Name="Unit Price"
Id="N1"><DecimalDigits><![CDATA[0]]></DecimalDigits><OnDataChangedScript><![CD
ATA[]]></OnDataChangedScript></Number><Number XmlNodeBoundTo="quantity"
Name="Quantity"
Id="N2"><DecimalDigits><![CDATA[0]]></DecimalDigits><OnDataChangedScript><![CD
ATA[]]></OnDataChangedScript></Number><Number XmlNodeBoundTo="itemTotal"
Name="Item Total"
Id="T6"><DecimalDigits><![CDATA[0]]></DecimalDigits><OnDataChangedScript><![CD
ATA[]]></OnDataChangedScript><PropertyExpressions><Property
Name="Value"><Code><![CDATA[qty = control.findById(/*Quantity*/"N2").value;
price = control.findById(/*Unit Price*/"N1").value;
return
parseInt(qty)*parseInt(price);]]></Code></Property></PropertyExpressions></Num
ber></Controls><OnRowAddedScript><![CDATA[]]></OnRowAddedScript><OnRowDeletedS
cript><![CDATA[]]></OnRowDeletedScript></BaseForm></Controls></PanelForm><Pane
lForm NumberOfColumns="1" FormTitle="" ProductVersion="" BuildVersion=""
Id="P2" FormMode="NextGenMode" XmlNodeBoundTo="" Name="Estimation
Details"><Controls><Number EnableValueAutoUpdate="True"
XmlNodeBoundTo="subtotal" Name="Sub Total"
Id="T7"><DecimalDigits><![CDATA[0]]></DecimalDigits><OnDataChangedScript><![CD
ATA[Taxcontrol = control.findById("T8");
taxVal = (currentValue*12)/100;
Taxcontrol.value = taxVal;
DiscountControl = control.findById("N3");
discountVal = (currentValue*1)/100;
DiscountControl.value=discountVal;
TotalControl = control.findById("T9");
TotalControl.value = parseInt(currentValue)+parseInt(taxVal)-
parseInt(discountVal);]]></OnDataChangedScript><PropertyExpressions><Property
Name="Value"><Code><![CDATA[return
parseFloat(control.parentForm.sumById("T6"));]]></Code></Property></PropertyEx
pressions></Number><TextInput XmlNodeBoundTo="taxes" Name="Tax"
Id="T8"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Nu
mber EnableValueAutoUpdate="True" XmlNodeBoundTo="discount" Name="Discount"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Id="N3"><DecimalDigits><![CDATA[0]]></DecimalDigits><OnDataChangedScript><![CD
ATA[]]></OnDataChangedScript></Number><TextInput XmlNodeBoundTo="total"
Name="Total"
Id="T9"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput></C
ontrols></PanelForm><PanelForm NumberOfColumns="1" FormTitle=""
ProductVersion="" BuildVersion="" Id="P3" FormMode="JCSMode"
XmlNodeBoundTo="approvalDetails" Name="Approval details"><Controls><TextInput
IsMandatory="True" XmlNodeBoundTo="comments" Name="Comments"
Id="T10"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><U
serLookUp FilterExpressions="{x:Null}" FilterDependencyIds="{x:Null}"
IsFilterExpressionSetForProperty="False" XmlNodeBoundTo="approver"
Enabled="False" Name="Approver"
Id="U1"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript><PropertyExpres
sions><Property Name="Value"><Code><![CDATA[return
SFU.getVirtualActorId();]]></Code></Property></PropertyExpressions></UserLookU
p></Controls><ScriptCode><![CDATA[]]></ScriptCode></PanelForm><FormButton
ShowAsFormLevelButton="True" ButtonText="Apply Corrections"
ButtonAlignment="Left" XmlNodeBoundTo="" LabelPosition="Left"
Id="F1"><ClickScript><![CDATA[if(control.findById("T10").value === "")
{
SFU.showAlert("You need to enter comments");
}
else
{
control.submitForm(event);
}]]></ClickScript></FormButton><FormButton ShowAsFormLevelButton="True"
ButtonText="Keep On Hold" XmlNodeBoundTo=""
Id="F2"><ClickScript><![CDATA[]]></ClickScript></FormButton><FormButton
ShowAsFormLevelButton="True" ButtonText="Reject" XmlNodeBoundTo=""
Id="F3"><ClickScript><![CDATA[if(control.findById("T10").value === "")
{
SFU.showAlert("You need to enter reason for rejection in comments");
}
else
{
control.submitForm(event);
}]]></ClickScript></FormButton><FormButton ShowAsFormLevelButton="True"
ButtonText="Approve" XmlNodeBoundTo=""
Id="F4"><ClickScript><![CDATA[]]></ClickScript></FormButton></Controls></BaseF
orm>
```

6. Create **Purchase Order Apply Corrections Form** for the requestors to make corrections based on the comments provided by approver.

   a. In the **Enterprise Console,** click ☰ menu icon, and then click **Forms**.

      The Forms page appears.

   b. Create a new Form named **Purchase Order Apply Corrections**, which will be used for the requester for making corrections suggested by approver.

   c. Import the Form definition from '**Purchase Order Apply Corrections_Form.xml**' using the Import menu option in Forms Designer or use the following xml file for importing.

```
<BaseForm SaveButtonText="@@Finish@@" FormTitle="" ProductVersion="1"
BuildVersion="40107100" Id="NF1" FormMode="NextGenMode"
XmlNodeBoundTo="PurchaseOrder" Name="Purchase Order Apply Corrections"
Description="Purchase Order Apply Corrections" xmlns="clr-
namespace:Skelta.Forms.Core.Controls;assembly:Skelta.Forms.Core" xmlns:L="clr-
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
namespace:System.Collections.Generic;assembly:mscorlib" xmlns:C="clr-
namespace:Skelta.Forms.Core.Classes;assembly:Skelta.Forms.Core"><ScriptCode><!
[CDATA[]]></ScriptCode><Controls><TextInput XmlNodeBoundTo="referenceNumber"
ReadOnly="True" Name="Reference Number"
Id="T1"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Da
teInput XmlNodeBoundTo="dateOrdered" Name="Ordered Date"
Id="D1"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></DateInput><Da
teInput XmlNodeBoundTo="dateRequired" Name="Required Date"
Id="D2"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></DateInput><Ra
dioButtonGroup XmlNodeBoundTo="priority" Name="Priority"
Id="R1"><Options><C:ColoredDictionary Key="High" Value="High"
ForeColor="#d61717" BackColor="" /><C:ColoredDictionary Key="Medium"
Value="Medium" ForeColor="#2d649f" BackColor="" /><C:ColoredDictionary
Key="Low" Value="Low" ForeColor="" BackColor=""
/></Options><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></RadioButt
onGroup><TextInput XmlNodeBoundTo="status" ReadOnly="True" Name="Status"
Id="T2"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Pa
nelForm NumberOfColumns="1" FormTitle="" ProductVersion="" BuildVersion=""
Id="P1" FormMode="NextGenMode" XmlNodeBoundTo="items" Name="Order
details"><Controls><BaseForm ViewMode="Grid" FormTitle="" ProductVersion=""
BuildVersion="" AddPosition="Bottom" Id="B1" DefaultRows="1"
FormMode="NextGenMode" XmlNodeBoundTo="item" Name="Items"><Controls><TextInput
XmlNodeBoundTo="itemNo" Name="Item No"
Id="T3"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Te
xtInput XmlNodeBoundTo="partNo" Name="Part No"
Id="T4"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Te
xtInput XmlNodeBoundTo="description" Name="Description"
Id="T5"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Nu
mber MaxValue="999999999999999" MinValue="-999999999999999"
XmlNodeBoundTo="unitPrice" Name="Unit Price"
Id="N1"><DecimalDigits><![CDATA[0]]></DecimalDigits><OnDataChangedScript><![CD
ATA[]]></OnDataChangedScript></Number><Number MaxValue="999999999999999"
MinValue="-999999999999999" XmlNodeBoundTo="quantity" Name="Quantity"
Id="N2"><DecimalDigits><![CDATA[0]]></DecimalDigits><OnDataChangedScript><![CD
ATA[]]></OnDataChangedScript></Number><Number XmlNodeBoundTo="itemTotal"
Name="Item Total"
Id="T6"><DecimalDigits><![CDATA[0]]></DecimalDigits><OnDataChangedScript><![CD
ATA[]]></OnDataChangedScript><PropertyExpressions><Property
Name="Value"><Code><![CDATA[qty = control.findById(/*Quantity*/"N2").value;
price = control.findById(/*Unit Price*/"N1").value;
return
parseInt(qty)*parseInt(price);]]></Code></Property></PropertyExpressions></Num
ber></Controls><OnRowAddedScript><![CDATA[]]></OnRowAddedScript><OnRowDeletedS
cript><![CDATA[]]></OnRowDeletedScript></BaseForm></Controls></PanelForm><Pane
lForm NumberOfColumns="1" FormTitle="" ProductVersion="" BuildVersion=""
Id="P2" FormMode="NextGenMode" XmlNodeBoundTo="" Name="Estimation Details
"><Controls><Number XmlNodeBoundTo="subtotal" Name="Sub Total"
Id="T7"><DecimalDigits><![CDATA[0]]></DecimalDigits><OnDataChangedScript><![CD
ATA[Taxcontrol = control.findById("T8");
taxVal = (currentValue*12)/100;
Taxcontrol.value = taxVal;
DiscountControl = control.findById("N3");
discountVal = (currentValue*1)/100;
DiscountControl.value=discountVal;
TotalControl = control.findById("T9");
TotalControl.value = parseInt(currentValue)+parseInt(taxVal)-
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
parseInt(discountVal);]]></OnDataChangedScript><PropertyExpressions><Property
Name="Value"><Code><![CDATA[return
parseFloat(control.parentForm.sumById("T6"));]]></Code></Property></PropertyEx
pressions></Number><TextInput XmlNodeBoundTo="taxes" Name="Tax"
Id="T8"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Nu
mber MaxValue="999999999999999" MinValue="-999999999999999"
XmlNodeBoundTo="discount" Name="Discount"
Id="N3"><DecimalDigits><![CDATA[0]]></DecimalDigits><OnDataChangedScript><![CD
ATA[]]></OnDataChangedScript></Number><TextInput XmlNodeBoundTo="total"
Name="Total"
Id="T9"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Pa
nelForm NumberOfColumns="1" FormTitle="" ProductVersion="" BuildVersion=""
Id="P3" FormMode="NextGenMode" XmlNodeBoundTo="approvalDetails" Name="Approval
Details"><Controls><TextInput XmlNodeBoundTo="comments" Name="Comments"
Id="T10"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><U
serLookUp FilterExpressions="{x:Null}" FilterDependencyIds="{x:Null}"
IsFilterExpressionSetForProperty="False" XmlNodeBoundTo="approver"
Name="Reviewer"
Id="U1"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></UserLookUp></
Controls></PanelForm></Controls></PanelForm></Controls></BaseForm>
```

d. Publish the form.

7. Create **Approved Purchase Orders** list to store the approved purchase orders.

   a. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**.

      The Manage List page appears.

   b. Create a new List named **Approved Purchase Orders**.

   c. Click **Save & Continue**.

      The **Select Table** page appears.

   d. Select the **Generic Table** option.

   e. Click **Next**.

      The List Designer appears.

   f. Import the list definition from **Approved Purchase Orders_List.xml** using **Import** menu option in List Designer or use the following XML:

```
<ListMainForm IsTableVersioned="False" DefaultField="Reference Number" UniqueField=""
EnableRecordIds="True" LockEnabled="False" OnEditCreateNewVersion="False"
OnEditCreateNewVersionforVG="False" UniqueTitle="False" AllowChildItems="True"
OldTableName="" TableName="SKEListItemDataNV"
TableId="6e370043-9f0d-43e6-ab45-5bab735c7a40" FormTitle="" ProductVersion=""
BuildVersion="" Id="_sys_approved purchase orders_list_skeltadynamiclist"
FormMode="JCSMode" XmlNodeBoundTo="ApprovedPurchaseOrders" Name="Approved Purchase
Orders" Description="Approved Purchase Orders" xmlns="clr-
namespace:Skelta.Forms.Core.Controls;assembly:Skelta.Forms.Core" xmlns:R="clr-
namespace:System;assembly:mscorlib" xmlns:x="http://uri.skelta.com/default/
serializer"><Controls><ListTextDataItem DisplayInGridAs="{x:Null}"
BoundToTableField="Title" GridDisplayWidth="100" DisplayInLookup="True"
XmlNodeBoundTo="referenceNumber" Name="Reference Number"
Id="_sys_ListForms_Titled05a335c720345699ba2f0b5c154c513" /><ListDateDataItem
DisplayInGridAs="{x:Null}" BoundToTableField="datetime1" XmlNodeBoundTo="dateOrdered"
Name="Date Ordered" Id="L1"><MaxDate><![CDATA[12/31/2099
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
23:59:59]]></MaxDate><MinDate><![CDATA[01/01/1901
00:00:00]]></MinDate><DateFormat><![CDATA[]]></DateFormat><Culture><![CDATA[]]></Culture><In
DisplayInGridAs="{x:Null}" BoundToTableField="datetime2"
XmlNodeBoundTo="dateRequired" Name="Date Required" Id="L2"><MaxDate><![CDATA[12/31/
2099 23:59:59]]></MaxDate><MinDate><![CDATA[01/01/1901
00:00:00]]></MinDate><DateFormat><![CDATA[]]></DateFormat><Culture><![CDATA[]]></Culture><In
DisplayInGridAs="{x:Null}" BoundToTableField="nvarchar1" XmlNodeBoundTo="priority"
Name="Priority" Id="L3" /><ListTextDataItem DisplayInGridAs="{x:Null}"
BoundToTableField="nvarchar2" XmlNodeBoundTo="status" Name="Status" Id="L4"
/><ListTableForm OldTableName="" IsTableVersioned="True"
TableName="SKEListItemDataNV" TableId="d6c4c0cf-0781-4172-b96d-705e8a842553"
FormTitle="" ProductVersion="" BuildVersion=""
Id="_sys_ApprovedPurchaseOrders_9c80ff1f2eca4427823aa01f471f4953" FormMode="JCSMode"
XmlNodeBoundTo="items/item" Name="Items"><Controls><ListTextDataItem
DisplayInGridAs="{x:Null}" BoundToTableField="nvarchar1" XmlNodeBoundTo="itemNo"
Name="Item No" Id="L6" /><ListTextDataItem DisplayInGridAs="{x:Null}"
BoundToTableField="nvarchar2" XmlNodeBoundTo="partNo" Name="Part No" Id="L7"
/><ListTextDataItem DisplayInGridAs="{x:Null}" BoundToTableField="nvarchar3"
XmlNodeBoundTo="description" Name="Description" Id="L8" /><ListDecimalDataItem
DisplayInGridAs="{x:Null}" BoundToTableField="decimal1" XmlNodeBoundTo="unitPrice"
Name="Unit Price" Id="L9" /><ListIntDataItem DisplayInGridAs="{x:Null}"
BoundToTableField="int1" XmlNodeBoundTo="quantity" Name="Quantity" Id="L10"
/><ListDecimalDataItem DisplayInGridAs="{x:Null}" BoundToTableField="decimal2"
XmlNodeBoundTo="itemTotal" Name="Item Total" Id="L11"
/></Controls></ListTableForm><ListDecimalDataItem DisplayInGridAs="{x:Null}"
BoundToTableField="decimal1" XmlNodeBoundTo="subtotal" Name="Sub Total" Id="L12"
/><ListDecimalDataItem DisplayInGridAs="{x:Null}" BoundToTableField="decimal2"
XmlNodeBoundTo="taxes" Name="Taxes" Id="L13" /><ListDecimalDataItem
DisplayInGridAs="{x:Null}" BoundToTableField="decimal3" XmlNodeBoundTo="discount"
Name="Discount" Id="L14" /><ListDecimalDataItem DisplayInGridAs="{x:Null}"
BoundToTableField="decimal4" XmlNodeBoundTo="total" Name="Total" Id="L15"
/><ListTableForm OldTableName="" IsTableVersioned="True"
TableName="SKEListItemDataNV" TableId="f2bb005d-9f99-42f1-b0b6-2a071c6bd15f"
FormTitle="" ProductVersion="" BuildVersion=""
Id="_sys_ApprovedPurchaseOrders_c3054ad789a349e680fa6da0372f12b7" FormMode="JCSMode"
XmlNodeBoundTo="approvalDetails" Name="Approval Details"><Controls><ListTextDataItem
DisplayInGridAs="{x:Null}" BoundToTableField="nvarchar1" XmlNodeBoundTo="comments"
Name="Comments" Id="L17" /><ListUserLookUp ShowDefaultUser="True" ForeignKeyQuery=""
MappedParentControl="" MappedParentControlName="" DisplayInGridAs="approver100"
BoundToTableField="uniqueidentifier1" XmlNodeBoundTo="approver" Name="Approver"
Id="L18"><BoundToTableFieldType><x:type typename="System.Guid"
/></BoundToTableFieldType></ListUserLookUp></Controls></ListTableForm></Controls></ListMainF
```

g. On submitting, a default Form for the List will be generated.

h. Say Yes to edit the generated (default) Form .

To change the look and feel of this Default Form, import the **Approved Purchase Orders_Form.xml** from Import menu option in Forms Designer or use this XML:

```
<BaseForm SaveButtonText="@@Finish@@" FormTitle="" ProductVersion="1"
BuildVersion="40107100" Id="NF1" FormMode="NextGenMode"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
XmlNodeBoundTo="ApprovedPurchaseOrders" Name="Approved Purchase Orders"
xmlns="clr-namespace:Skelta.Forms.Core.Controls;assembly:Skelta.Forms.Core"
xmlns:R="clr-namespace:System;assembly:mscorlib"
xmlns:x="http://uri.skelta.com/default/
serializer"><ScriptCode><![CDATA[]]></ScriptCode><Controls><TextInput
MaxLength="250" DisplayInLookup="True" XmlNodeBoundTo="referenceNumber"
Name="Reference Number"
Id="_sys_ListForms_Titled05a335c720345699ba2f0b5c154c513" /><DateTimeInput
Interval="01:00:00" BoundToTableField="datetime1" GridDisplayWidth="20"
DisplayInLookup="True" XmlNodeBoundTo="dateOrdered" Name="Date Ordered"
Id="L1"><BoundToTableFieldType><x:type typename="System.DateTime"
/></BoundToTableFieldType></DateTimeInput><DateTimeInput Interval="01:00:00"
BoundToTableField="datetime2" GridDisplayWidth="20" DisplayInLookup="True"
XmlNodeBoundTo="dateRequired" Name="Date Required"
Id="L2"><BoundToTableFieldType><x:type typename="System.DateTime"
/></BoundToTableFieldType></DateTimeInput><TextInput MaxLength="255"
DisplayInLookup="True" XmlNodeBoundTo="priority" Name="Priority" Id="L3"
/><TextInput MaxLength="255" DisplayInLookup="True" XmlNodeBoundTo="status"
Name="Status" Id="L4" /><PanelForm NumberOfColumns="1" FormTitle=""
ProductVersion="" BuildVersion="" Id="P1" FormMode="NextGenMode"
XmlNodeBoundTo="items" Name="Order Details"><Controls><BaseForm FormTitle=""
ProductVersion="" BuildVersion=""
Id="_sys_ApprovedPurchaseOrders_9c80ff1f2eca4427823aa01f471f4953"
FormMode="JCSMode" XmlNodeBoundTo="item" Name="Items" ViewMode="Grid"
IsTableVersioned="True" AddPosition="Bottom"><Controls><TextInput
MaxLength="255" DisplayInLookup="True" XmlNodeBoundTo="itemNo" Name="Item No"
Id="L6" /><TextInput MaxLength="255" DisplayInLookup="True"
XmlNodeBoundTo="partNo" Name="Part No" Id="L7" /><TextInput MaxLength="255"
DisplayInLookup="True" XmlNodeBoundTo="description" Name="Description" Id="L8"
/><Number MaxValue="9999999999999998" MinValue="-9999999999999998"
BoundToTableField="decimal1" DisplayInLookup="True" XmlNodeBoundTo="unitPrice"
Name="Unit Price"
Id="L9"><DecimalDigits><![CDATA[2]]></DecimalDigits><BoundToTableFieldType><x:
type typename="System.Decimal" /></BoundToTableFieldType></Number><Number
MaxValue="2147483647" MinValue="-2147483648" BoundToTableField="int1"
DisplayInLookup="True" XmlNodeBoundTo="quantity" Name="Quantity"
Id="L10"><DecimalDigits><![CDATA[0]]></DecimalDigits><BoundToTableFieldType><x
:type typename="System.Int32" /></BoundToTableFieldType></Number><Number
MaxValue="9999999999999998" MinValue="-9999999999999998"
BoundToTableField="decimal2" DisplayInLookup="True" XmlNodeBoundTo="itemTotal"
Name="Item Total"
Id="L11"><DecimalDigits><![CDATA[2]]></DecimalDigits><BoundToTableFieldType><x
:type typename="System.Decimal"
/></BoundToTableFieldType></Number></Controls><OnRowAddedScript><![CDATA[]]></
OnRowAddedScript><OnRowDeletedScript><![CDATA[]]></OnRowDeletedScript></BaseFo
rm></Controls></PanelForm><Number MaxValue="9999999999999998"
MinValue="-9999999999999998" BoundToTableField="decimal1"
DisplayInLookup="True" XmlNodeBoundTo="subtotal" Name="Sub Total"
Id="L12"><DecimalDigits><![CDATA[2]]></DecimalDigits><BoundToTableFieldType><x
:type typename="System.Decimal" /></BoundToTableFieldType></Number><Number
MaxValue="9999999999999998" MinValue="-9999999999999998"
BoundToTableField="decimal2" DisplayInLookup="True" XmlNodeBoundTo="taxes"
Name="Taxes"
Id="L13"><DecimalDigits><![CDATA[2]]></DecimalDigits><BoundToTableFieldType><x
:type typename="System.Decimal" /></BoundToTableFieldType></Number><Number
MaxValue="9999999999999998" MinValue="-9999999999999998"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
BoundToTableField="decimal3" DisplayInLookup="True" XmlNodeBoundTo="discount"
Name="Discount"
Id="L14"><DecimalDigits><![CDATA[2]]></DecimalDigits><BoundToTableFieldType><x
:type typename="System.Decimal" /></BoundToTableFieldType></Number><Number
MaxValue="9999999999999998" MinValue="-9999999999999998"
BoundToTableField="decimal4" DisplayInLookup="True" XmlNodeBoundTo="total"
Name="Total"
Id="L15"><DecimalDigits><![CDATA[2]]></DecimalDigits><BoundToTableFieldType><x
:type typename="System.Decimal" /></BoundToTableFieldType></Number><PanelForm
NumberOfColumns="1" FormTitle="" ProductVersion="" BuildVersion=""
Id="_sys_ApprovedPurchaseOrders_c3054ad789a349e680fa6da0372f12b7"
FormMode="JCSMode" XmlNodeBoundTo="approvalDetails" Name="Approval
Details"><Controls><TextInput MaxLength="255" DisplayInLookup="True"
XmlNodeBoundTo="comments" Name="Comments" Id="L17" /><UserLookUp
FilterExpressions="{x:Null}" FilterDependencyIds="{x:Null}"
IsFilterExpressionSetForProperty="False" BoundToTableField="uniqueidentifier1"
DisplayInLookup="True" XmlNodeBoundTo="approver" Name="Approver"
Id="L18"><BoundToTableFieldType><x:type typename="System.Guid"
/></BoundToTableFieldType></UserLookUp></Controls></PanelForm></Controls></Bas
eForm>
```

i. Submit and publish the form.

8. Execute and check.
   a. Log on to the **Enterprise Console** as Tim
   b. In the **Enterprise Console**, click ☰ menu icon, and then click **Fill Form**.

      The Fill Form page appears.
   c. Select the **Purchase Order Requisition** form, fill the details, and then submit the request.
   d. Go to the Inbox of John, leave a comment, and click **Apply Corrections**.

      The request is returned to Tim.

      When Tim resubmits, the changed details can be seen in John's Inbox.

      When John when keeps it on Hold, information is sent to Tim but the work item will be retained in John's Inbox.

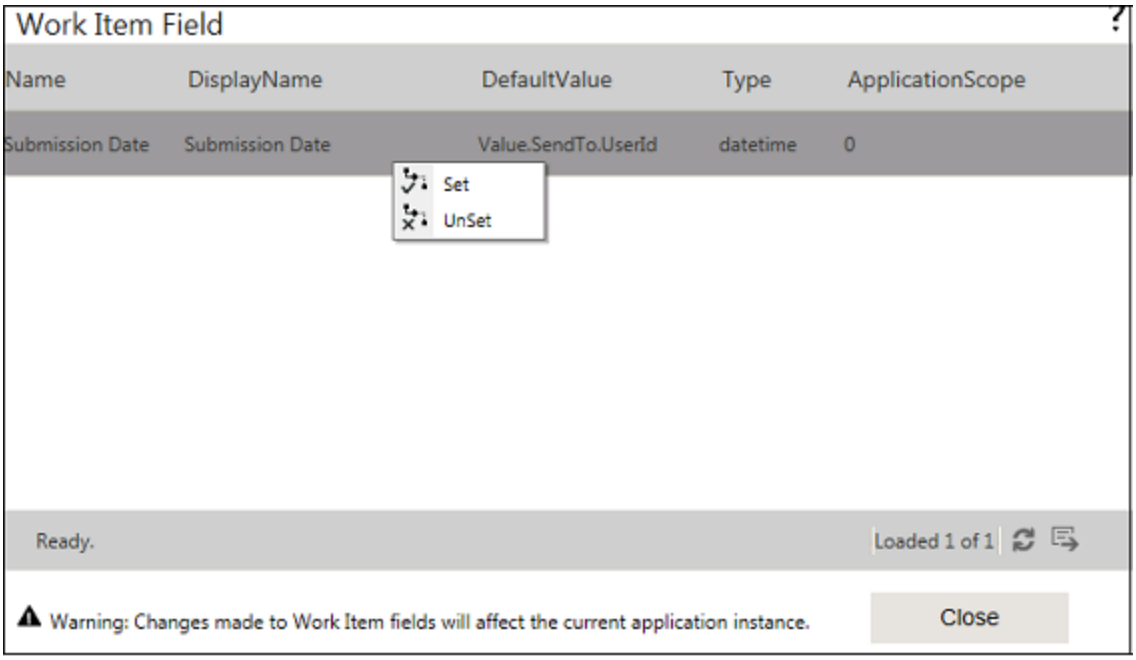
      When John Approves, the requested purchase order is added to the Approved Purchase Orders list.

Invoke Form - Work Item Fields

Purpose

You can set the Work Item Fields for this activity in this property window. To open this window, click the button for the **'Work Item Fields'** property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a grid listing all the Work Item Fields that have been set for the process.

- Each row in the grid represents a Work Item Field. You can click on a Work Item Field to select it.
- The context menu has the Set WorkItemField option, which you can use to set the selected Work Item Field for this activity. It also has the UnSet WorkItemField option to remove the Work Item Field setting.
- **Close** - You can use this button to close the window.

Note: While entering a value in the Work Item field, it is recommended to keep it less than the specified database column size. If there is a need to increase the size of the Work Item field, the user should manually increase the column size in the database.

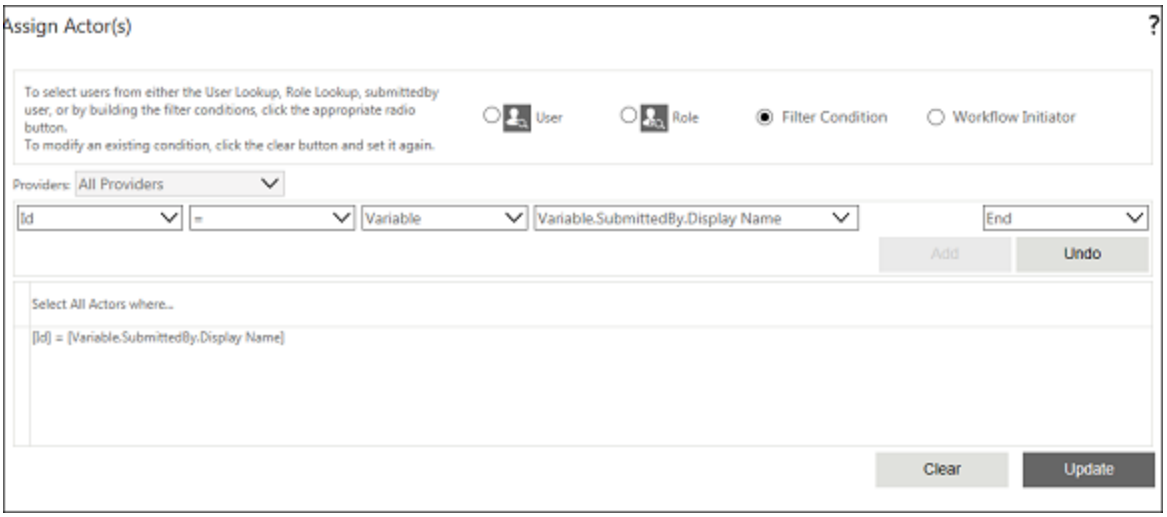
Invoke Form - Assign Actor(s)

Assign Actor(s) Property Window

You can set the actors to whom the Invoke Form work item is to be sent in this property window. To access this window, click the button in the 'Assign Actor(s)' property.

See Assigning a Task to a Manager for more details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Different Methods of Assigning an Actor

- **Assigning an Actor by building an expression:**

Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
- If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
- If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.

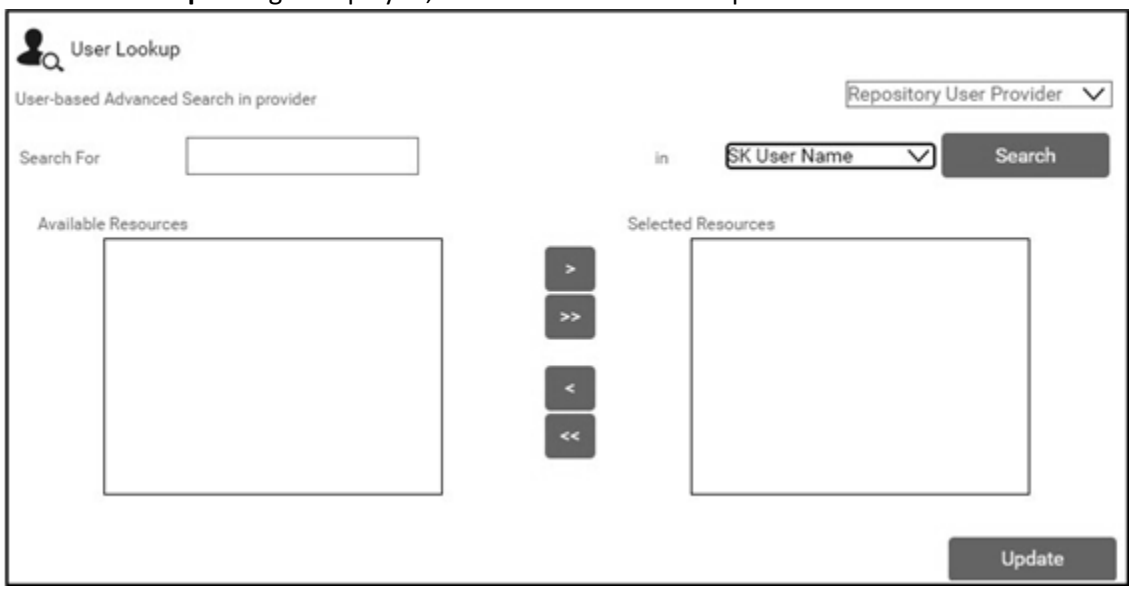
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Update** - You can click this button to save the expression that has been added to the text area.
- **Assigning an Actor through User Lookup**

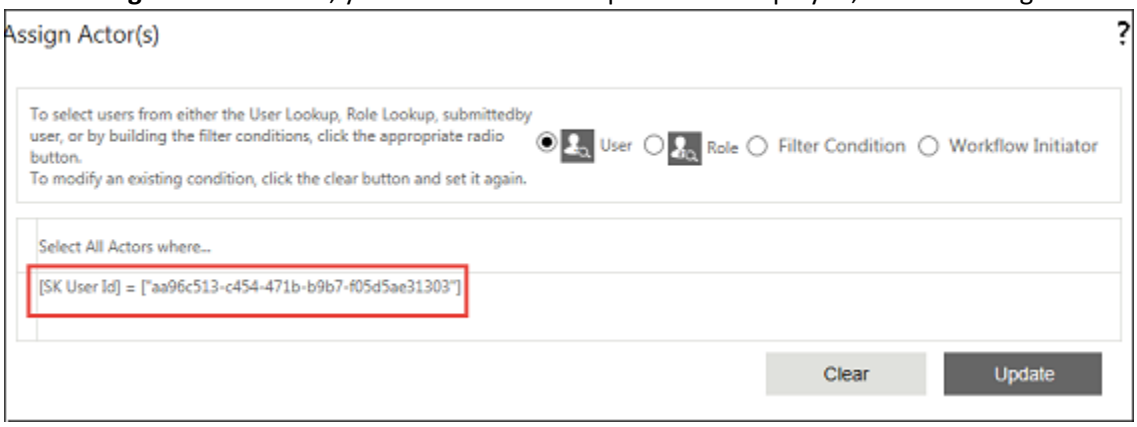
Through the User Lookup you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.

   The **User Lookup** dialog is displayed, search and select the required actor.



2. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:



3. Click **Update** in the Assign Actor(s) window to update the selected actor(s) in the **Assign Actor** property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

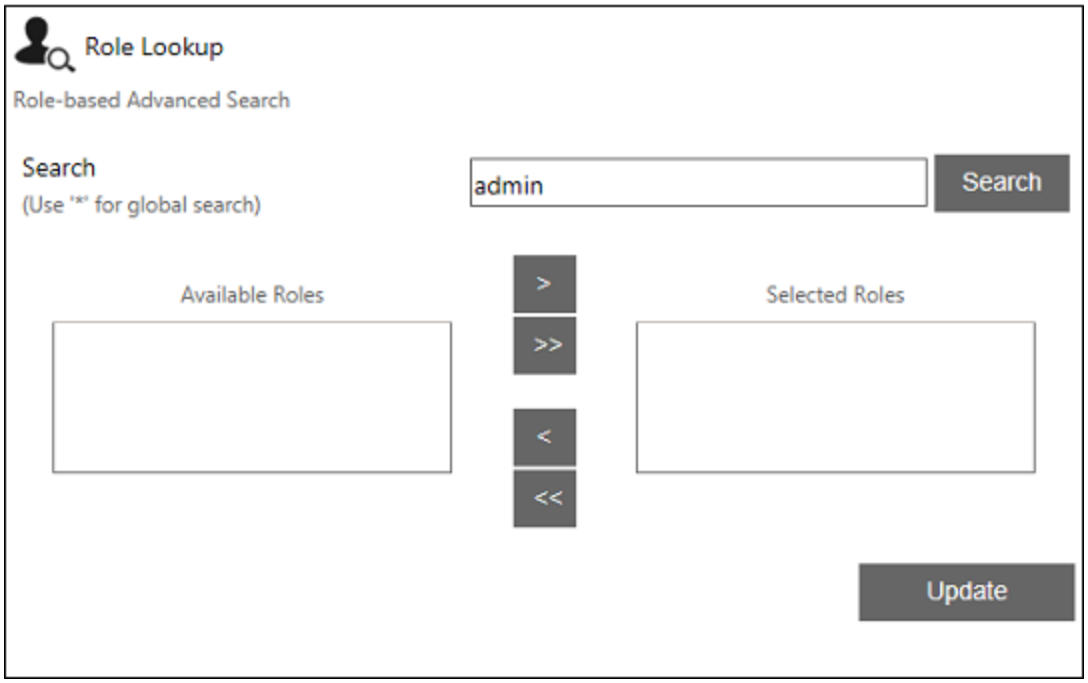
- **Assigning an Actor through Role Lookup**

Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

1. Select the **Role Lookup** option.

   The **Role Lookup** dialog is displayed, search and select for the required role.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



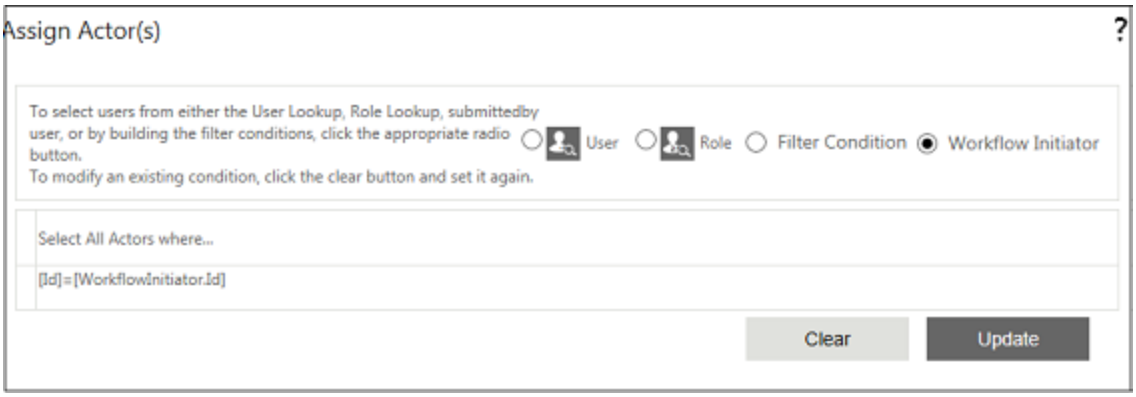
2. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

3. Click **Update** in the Assign Actor(s) window to update the **Assign Actor** property.

- **Assigning an Actor through Workflow Initiator**

Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

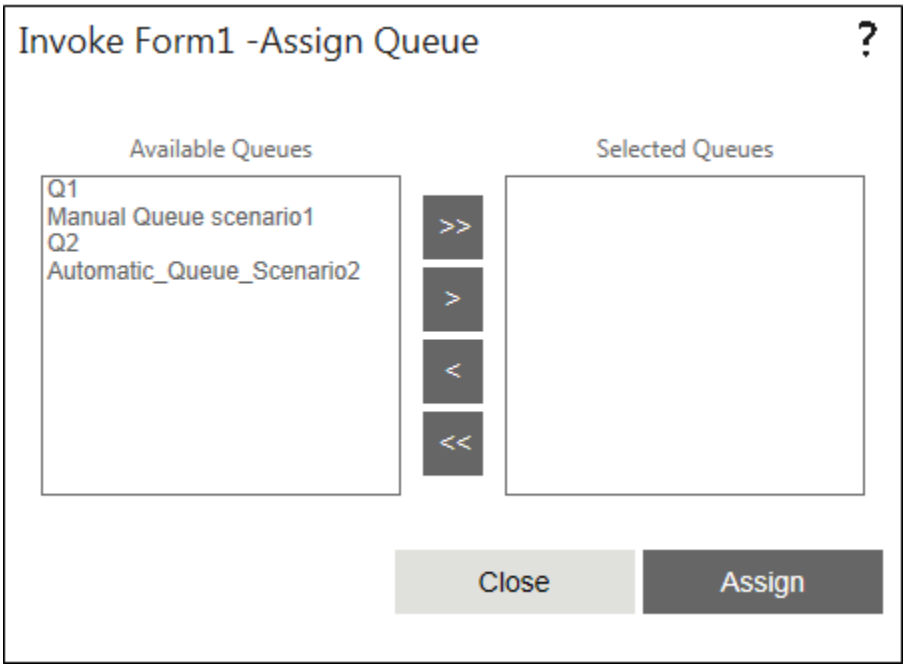
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Invoke Form - Assign Queue(s)

Assign Queue(s) Property Window

You can select the queues to which the Invoke Form work item has to be sent in this property window. To access this window, click the button in the 'Assign Queue(s)' property.



Assign Queue(s) Properties window for Invoke Form

The window has a pair of list boxes to specify the queue(s). The **Available Queues** list shows all the available queues. You can select the required queue(s) from this list box. The Selected Queues list shows all the selected queues to which the activity will be assigned.

You can use the following buttons located between the list boxes to move the queues to and from the list boxes.

- [>>] - You can click this button to add all the queues in the Available Queues list to the Selected Queues list.
- [>] - You can click this button to add the selected queue(s) in the Available Queues list to the Selected Queues list.
- [<] - You can click this button to add the selected queue(s) from the Selected Queues list to the Available

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Queues list.

- [<<] - You can click this button to add all the queues in the Selected Queues list to the Available Queues list.

At the bottom of the window you will find the following buttons:

- Assign - You can click this button to assign the activity to the selected queues.
- Close - You can click this button to close this window.

Assigning queue system variable

To assign queues instead of setting the queue through Assign Queue property, add a variable with a name prefixed by '^', with the following syntax:

"^"+ CurrentActivity.Name + "_QueueVariable".

For example, if the activity name is 'Approval1' then declare the variable as ^Approval1_QueueVariable in the **Start** activity with the Type as String. Using the "Update Variable" activity, assign the queue-id (GUID) for this variable in the workflow.

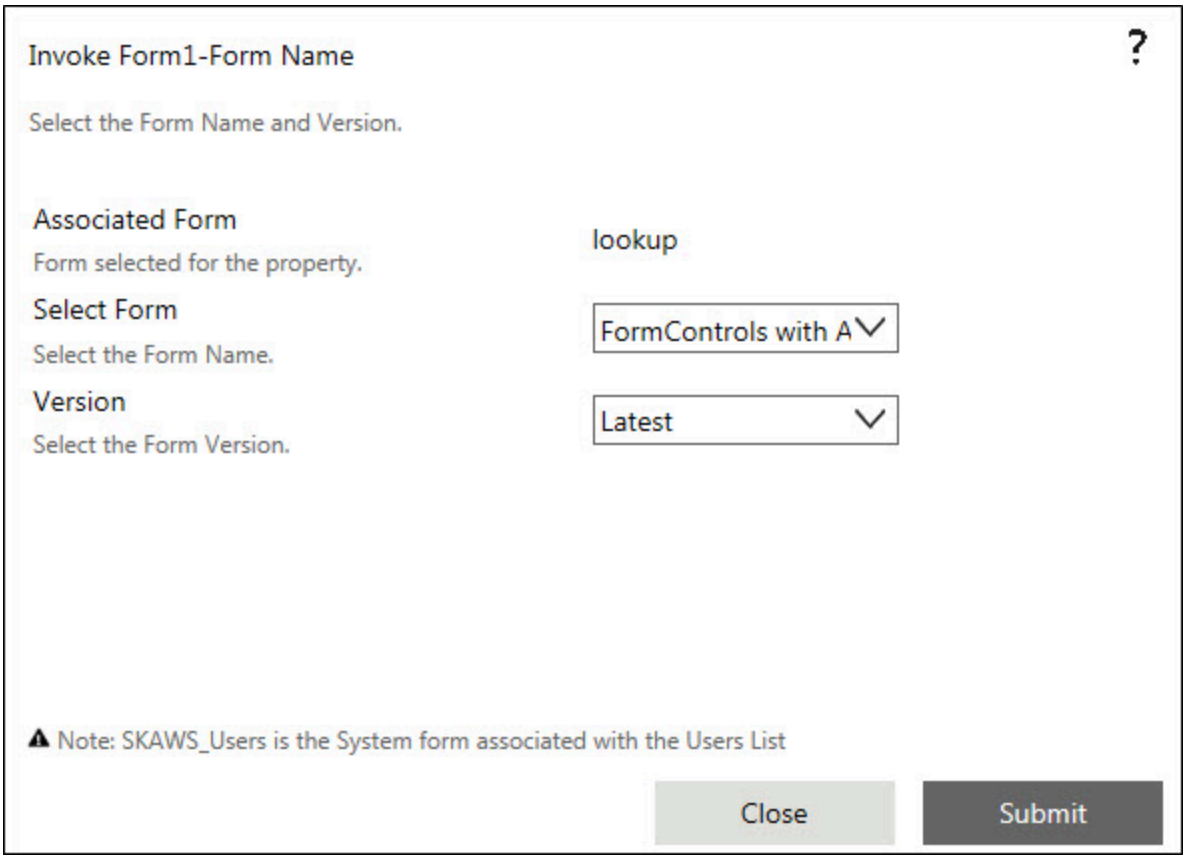
To assign multiple queues, specify the queue-ids with semicolon (;) as the separator. **Skelta.HWS.Queue.QueueCollection** class can be used to retrieve queue information.

Invoke Form - Form Name

Form Name Property Window

To open this window, click the button in the 'Form Name' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Form Name Properties window for Invoke Form

The Form Name window has the following fields:

Associated Form - Name of the associated form.

Form Name - You can select the Form name from this drop-down list. The pre-defined SKAWS_Users Form associated to the User list will also be available in this list.

Version - You can select the version from this drop-down list and default value is Latest Version.

Invoke Form - Output Instance XML

This property is used to get the data from the Forms which you have edited from the Work items page. Depending on the type of XML variable selected in this property, the XML data is stored, provided the output instance XML has the same schema of the Form.

For instance:

1. If the **Output Instance XML** is of type **XML List** , then the XML data received from the Form is stored in the respective List depending on the list item scope defined for the XML Variable.
   a. The list item scope should be New , if a new list item has to be added with the Form data in list.
   b. The list item scope should be Current, if the list item that initiated the workflow has to be updated.
   c. The list item scope should be List Query, if a specific list item has to be updated in the list.
2. If the **Output Instance XML** is of type **XML Document**, then the XML data received is stored in the workflow itself.

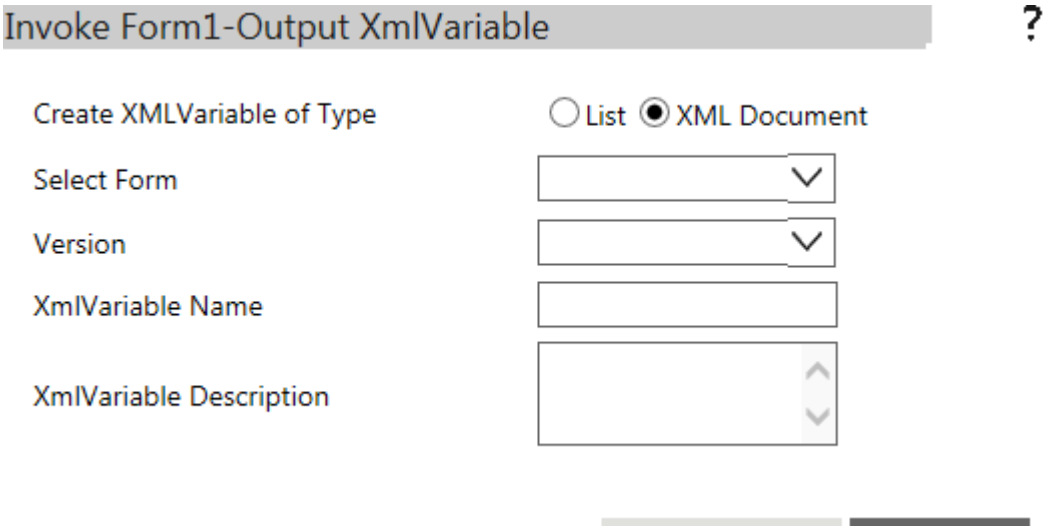
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scenario: Display an existing list item data using **Invoke Form** and then store the updated Form data in a different list (destination list): An intermediate XML Variable is required for such scenarios. Use an **Update Variable** activity before the **Invoke Form** activity, to assign the Source list XML values to the Intermediate XML Variable, Then use this intermediate XML variable as **Input Instance XML** and destination XML List as **Output Instance XML**.

Steps to create Output Instance XML variable

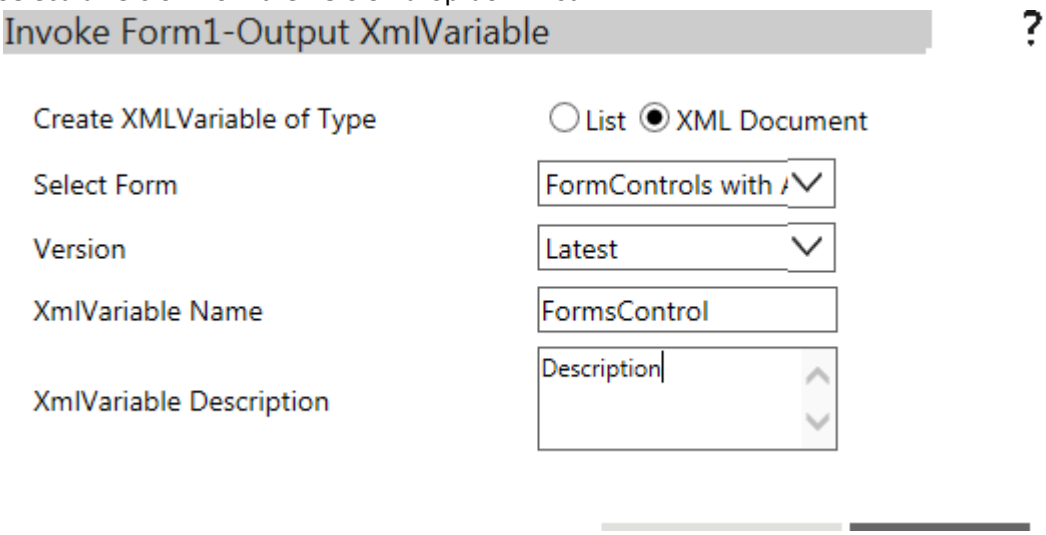
1. Select an existing output XML variable from the **Output Instance XML** drop-down list, Or Click the edit button corresponding to Create a **Output Instance XML** variable.

   The following screen will be displayed.



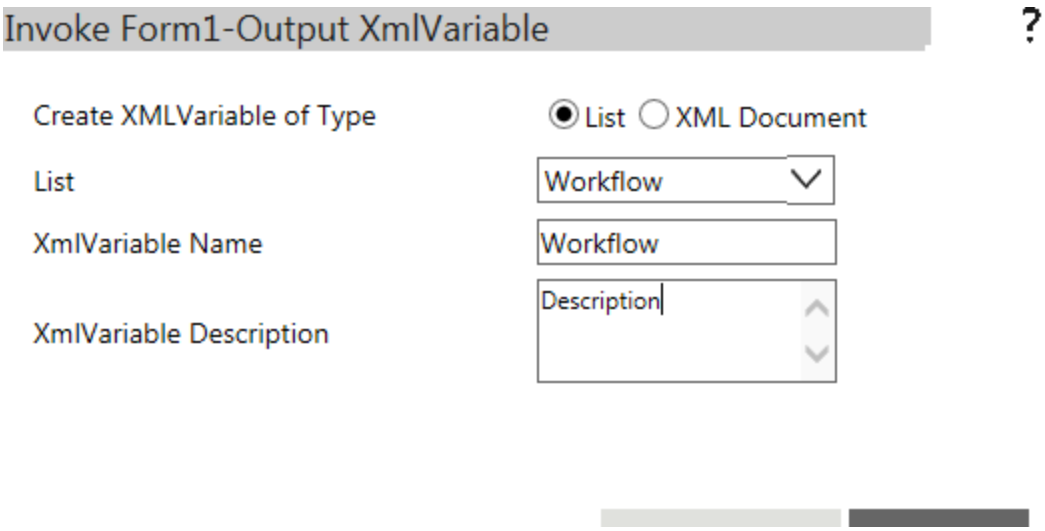
   Create a Output Instance XML Variable

2. Select a form from the **Form Name** drop-down list.
3. Select a version from the **Version** drop-down list.



Note: By default, the No option is selected and the screen as shown in the image above is displayed. If the Yes option is selected, the following screen is displayed. The difference, as you can notice, is that the No option allows you to select a form, whereas the Yes option allows you to select a list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



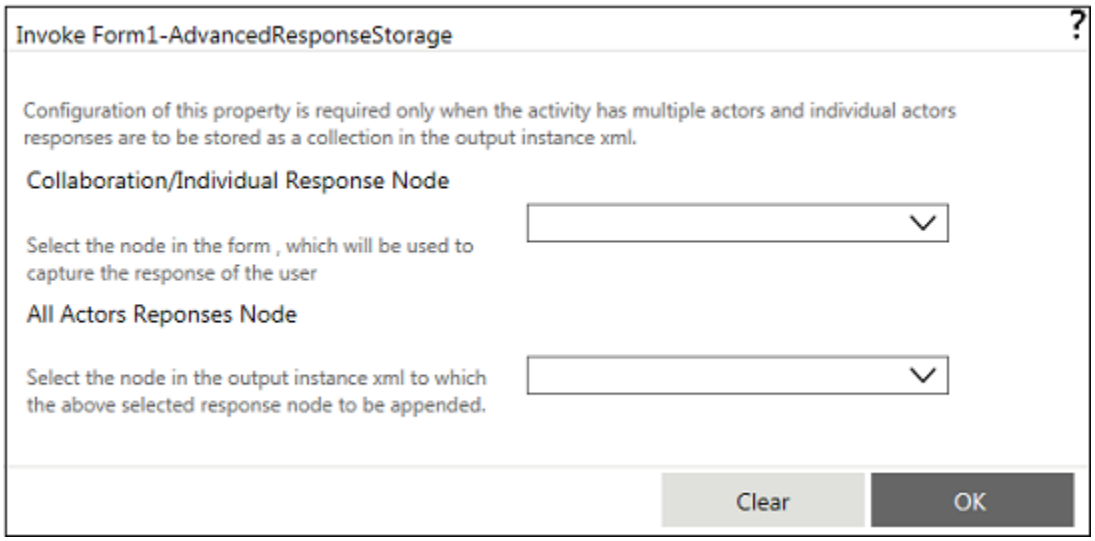
Note: On this screen, select a list from the List drop-down.

1. Enter a name for the output XML variable in the **XMLVariable** Name field.
2. Enter a description for the output XML variable in the **XMLVariable Description** field.
3. Click **Submit**.
4. The output XML variable is created. The created output XML variable will be available in Start activity.

Invoke Form - Individual Actor's Response Collaboration in Output Instance XML

You can use this property to collaborate with an individual actor's responses in the Invoke Form activity. You can configure this property only when the activity has multiple actors and you want to store the individual actor's responses as a collection in an Output Instance XML. When you use this property, the entire OutputXML is not updated. Only the nodes specified in the Collaboration/Individual Response Node is copied to the All Actors Responses Node of the Output instance XML.

The following image shows the input fields of the collaboration mode property:



If the Invoke Form Activity is not configured in collaboration mode, and if the task is assigned to multiple users, then only the submission changes made by the last user will be considered.

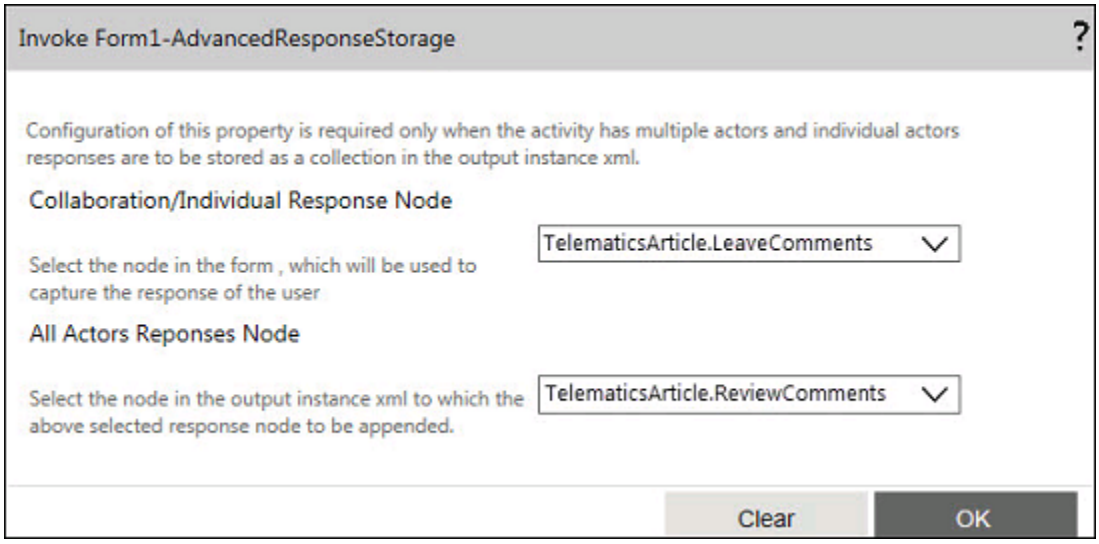
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Inputs for Collaboration/Individual Response Node and All Actors Responses Node must be either a base form or a panel form with XML node. You need to set Collaboration/Individual Response Node, which is the node used to capture the response of individual user. Also All Actors Responses Node is the node in the output instance XML to which the individual response node will be appended.

Steps to capture the Individual Actor's response collaboration in Output Instance XML

1. Click the edit button corresponding to Individual actor's response collaboration in Output Instance XML.

   The Invoke Form-AdvancedResponseStorage dialog box appears.



2. In the Collaboration/Individual Response Node, select the node in the form which will be used to capture the response of the user.
3. You can edit only the controls of the selected node. All the other controls will be read-only.
4. In the All Actors Responses Node, select the node in the output instance XML to which the above selected response node is to be appended.
5. Click OK.

Note: You can even include a User control to capture the User ID by default. This can be either visible or invisible to the end-user. To create a user control in the Form, you will have to populate the logged in user value in a form to a hidden control, so that in a workflow, a decision action can be performed based on the user who has filled the form.

Example scenario

The Telematics Company allows its employees to post articles. The articles posted can be viewed by a group of users defined in the workflow and these users can leave their comments on the article. These article viewers can also view the comments provided by others.

To achieve this, the user has to go through six steps. For each step, the respective XML file name with code is given below, which you can use.

Note:

1. In this scenario, we have used the repository with AVEVA Work Tasks List as the user provider.

2. Create XML files with the same filename as given below. Place all the files in a single folder for ease of use.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Create a list of users in the User's list with the group name 'ArticleSubscribers'.
   a. In the Enterprise Console, click the Menu button, click Settings, and then click Users. The Users page appears.
   b. Create some users with ArticleSubscribers as their role.
2. Create a List to store the submitted articles.
   a. In the Enterprise Console, click the Menu button, click Tools, and then click Manage List. The Manage List page appears.
   b. Create a new list named Telematics Article to store the submitted articles.
   c. Click on Save & Continue to go to the Select Table page.
   d. Select the Generic Table option.
   e. Click Next to go to the List Designer.
   f. Import the list definition from Telematics Article List.xml using Import menu option in the List Designer.
   g. Click Submit, to generate a default form for the list.
   h. Click Yes, to edit the default form which is generated. To change the look and feel of this default form, import the Telematics Article_ListForm.xml from Import menu option in forms designer. Now your form looks like the image given below in the forms designer.
   i. Submit and publish the form.

3. Create a Form to show the submitted article to the 'ArticleSubscribers' group, using which the users will be able to add a comment, as well as view comments given by other users.
   a. In the Enterprise Console, click the Menu button, click Tools, and then click Manage Forms. The Manage Forms page appears.
   b. Create a form named TelematicsArticlesReview.
   c. Import the form definition from TelematicsArticlesReview_Form.xml using the Import menu option in the Forms designer. If you notice, your form is again an exact copy of the Telematics Article form, but with an extra panel of 'Leave a Comment' section.
   d. You can see a tab in this form, where the first tab is to leave the comment by the Articlesubscriber group, and the second is to view the comments posted by other s in the group. If you do not need to show the others comment section, you can remove the views other's comment section.
   e. Check the mapped XML nodes (of parent and child controls) of both the 'Leave a comment' and 'Review Comments' control. Here you might have noticed that the child nodes have the same mapped XML nodes where as only for the parent it differs.
   f. To append/show a row by default for 'Leave a comment' panel, you can notice that a script is defined for the category drop down. Notice that this can also be achieved without scripting by setting the 'Rows to be added by default' as '1' and 'Allow default rows to render on XML Load?' as 'Yes' in the Advanced tab of 'Leave a comment' panel properties.
   g. Submit and Publish the form.
4. Create an Initial Form to submit the article.
   a. In the Enterprise Console, click the Menu button, click Tools, and then click Manage Forms. The Manage Forms page appears.
   b. Create a Workflow Initiator Form named Telematics Article Submission.
   c. Import the form definition from Telematics Article Submission Form.xml using the Import menu option in

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the Forms designer. You will notice that your form is an exact copy of the Telematics Article form, but without the Review comments section.

d. Submit and Publish the form.

5. Associate an Article Submission Form to the Article Review Workflow.
   a. On submission, a page related to Form Binding opens, where the user can select a task as designing a new workflow or selecting an existing one.
   b. Select Design a New Workflow and provide the workflow name as ArticleReview.
   c. Select Design this Workflow now in the next page.
   d. From the Process Designer, use the Import option from Tools menu and import the workflow from ArticleReview Workflow.xml.
   e. Publish the workflow.
6. Execute the form to see how each individual actor's responses collaborate.
   a. In the Enterprise Console, click the Menu button, and then click Fill Form. The Fill Form page appears.
   b. Select the Telematics Article Submission form, fill the details, and submit the form.
   c. Go to the Inbox of each ArticleSubscribers, leave a comment, and submit the form.
   d. You can view all user's comments under 'View Other's comments' section.

Telematics Article List.xml

```
<ListMainForm IsTableVersioned="False" DefaultField="Title" UniqueField=""
EnableRecordIds="True" LockEnabled="False" OnEditCreateNewVersion="False"
OnEditCreateNewVersionforVG="False" UniqueTitle="False" AllowChildItems="True"
OldTableName="" TableName="SKEListItemDataNV" TableId="60b121c4-821a-450d-
a353-2fd419416b29" FormTitle="" ProductVersion="" BuildVersion="" Id="_sys_telematics
article_list_skeltadynamiclist" FormMode="JCSMode" XmlNodeBoundTo="TelematicsArticle"
Name="Telematics Article" xmlns="clr-
namespace:Skelta.Forms.Core.Controls;assembly:Skelta.Forms.Core" xmlns:R="clr-
namespace:System;assembly:mscorlib" xmlns:x="http://uri.skelta.com/default/
serializer"><Controls><ListTextDataItem AllowSorting="True" DisplayInGridAs="{x:Null}"
BoundToTableField="Title" GridDisplayWidth="100" DisplayInLookup="True"
XmlNodeBoundTo="Title" Name="Title"
Id="_sys_ListForms_Title7d734be0c4d345278034644b83401d71" /><ListTextDataItem
DisplayInGridAs="{x:Null}" BoundToTableField="nvarchar1" XmlNodeBoundTo="Category"
Name="Category" Id="L8" /><ListTextDataItem DisplayInGridAs="{x:Null}"
BoundToTableField="nvarchar2" XmlNodeBoundTo="Overview" Name="Overview" Id="L1"
/><ListTextDataItem DisplayInGridAs="{x:Null}" BoundToTableField="ntext1"
XmlNodeBoundTo="Contents" Name="Contents" Id="L3" GridDisplay="ImageAndText"
/><ListTextDataItem DisplayInGridAs="{x:Null}" BoundToTableField="nvarchar3"
XmlNodeBoundTo="Summary" Name="Summary" Id="L2" /><ListTableForm OldTableName=""
IsTableVersioned="True" TableName="SKEListItemDataNV"
TableId="3e939db4-71d4-46b8-8cb3-3e26c0fd8939" FormTitle="" ProductVersion=""
BuildVersion="" Id="_sys_TelematicsArticles_fcb5aa8a7aa04caaae1961c8fa655e78"
FormMode="JCSMode" XmlNodeBoundTo="ReviewComments" Name="Review
Comments"><Controls><ListUserLookUp ShowDefaultUser="True" ForeignKeyQuery=""
MappedParentControl="" MappedParentControlName="" DisplayInGridAs="Name100"
BoundToTableField="uniqueidentifier1" XmlNodeBoundTo="Name" Name="Name"
Id="L5"><BoundToTableFieldType><x:type typename="System.Guid"
/></BoundToTableFieldType></ListUserLookUp><ListTextDataItem DisplayInGridAs="{x:Null}"
BoundToTableField="nvarchar1" XmlNodeBoundTo="Rating" Name="Rating" Id="L6"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/><ListTextDataItem DisplayInGridAs="{x:Null}" BoundToTableField="ntext1"
XmlNodeBoundTo="Comments" Name="Comments" Id="L7"
/></Controls></ListTableForm></Controls></ListMainForm>
```

Telematics Article_ListForm.xml

```
<BaseForm SaveButtonText="@@Finish@@" FormTitle="" ProductVersion="1"
BuildVersion="40107400" Id="NF1" FormMode="NextGenMode" XmlNodeBoundTo="TelematicsArticle"
Name="Telematics Article" xmlns="clr-
namespace:Skelta.Forms.Core.Controls;assembly:Skelta.Forms.Core" xmlns:R="clr-
namespace:System;assembly:mscorlib" xmlns:x="http://uri.skelta.com/default/
serializer"><ScriptCode><![CDATA[]]></ScriptCode><Controls><TextInput MaxLength="250"
DisplayInLookup="True" IsMandatory="True" XmlNodeBoundTo="Title" Name="Title"
Id="_sys_ListForms_Title7d734be0c4d345278034644b83401d71" /><TextInput MaxLength="255"
DisplayInLookup="True" XmlNodeBoundTo="Category" Name="Category" Id="L8" /><TextInput
MaxLength="255" DisplayInLookup="True" XmlNodeBoundTo="Overview" Name="Overview" Id="L1"
/><TextInput MaxLength="1073741823" DisplayInLookup="True" XmlNodeBoundTo="Contents"
Name="Contents" Id="L3" /><TextInput MaxLength="255" DisplayInLookup="True"
XmlNodeBoundTo="Summary" Name="Summary" Id="L2" /><BaseForm FormTitle="" ProductVersion=""
BuildVersion="" Id="_sys_TelematicsArticles_fcb5aa8a7aa04caaae1961c8fa655e78"
FormMode="JCSMode" XmlNodeBoundTo="ReviewComments" Name="Review Comments"
IsTableVersioned="True" ViewMode="Grid"><Controls><UserLookUp FilterExpressions="{x:Null}"
FilterDependencyIds="{x:Null}" IsFilterExpressionSetForProperty="False"
BoundToTableField="uniqueidentifier1" DisplayInLookup="True" XmlNodeBoundTo="Name"
Name="Name" Id="L5"><BoundToTableFieldType><x:type typename="System.Guid"
/></BoundToTableFieldType></UserLookUp><TextInput MaxLength="255" DisplayInLookup="True"
XmlNodeBoundTo="Rating" Name="Rating" Id="L6" /><TextInput MaxLength="1073741823"
DisplayInLookup="True" XmlNodeBoundTo="Comments" Name="Comments" Id="L7"
TextLine="MultiLine"
Rows="2"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput></Controls><OnR
owAddedScript><![CDATA[]]></OnRowAddedScript><OnRowDeletedScript><![CDATA[]]></OnRowDeleted
Script></BaseForm></Controls></BaseForm>
```

TelematicsArticlesReview_Form.xml

```
<BaseForm SaveButtonText="@@Finish@@" FormTitle="TelematicsArticlesReview"
ProductVersion="1" BuildVersion="40107400" Id="NF1" FormMode="NextGenMode"
XmlNodeBoundTo="TelematicsArticle" Name="Telematics Articles" xmlns="clr-
namespace:Skelta.Forms.Core.Controls;assembly:Skelta.Forms.Core" xmlns:L="clr-
namespace:System.Collections.Generic;assembly:mscorlib" xmlns:C="clr-
namespace:Skelta.Forms.Core.Classes;assembly:Skelta.Forms.Core"><ScriptCode><![CDATA[]]></S
criptCode><Controls><TextInput IsMandatory="True" XmlNodeBoundTo="Title" Name="Title"
Id="T1"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><DropDownChoice
XmlNodeBoundTo="Category" Name="Category" Id="D1"><Options><C:ColoredDictionary
Key="Programming" Value="Programming" ForeColor="" BackColor="" /><C:ColoredDictionary
Key="Database" Value="Database" ForeColor="" BackColor="" /><C:ColoredDictionary
Key="Graphics" Value="Graphics" ForeColor="" BackColor="" /><C:ColoredDictionary
Key="Process Improvement" Value="Process Improvement" ForeColor="" BackColor=""
/></Options><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></DropDownChoice><TextIn
put XmlNodeBoundTo="Overview" Name="Overview" Id="T2" TextLine="MultiLine" Rows="5"
LabelPosition="Top"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Tex
tInput XmlNodeBoundTo="Contents" Name="Contents" Id="T3" TextLine="MultiLine" Rows="5"
LabelPosition="Top"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Tex
tInput XmlNodeBoundTo="Summary" Name="Summary" Id="T4" TextLine="MultiLine" Rows="5"
LabelPosition="Top"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Pan
elForm NumberOfColumns="1" FormTitle="" ProductVersion="" BuildVersion="" Id="P1"
FormMode="NextGenMode" XmlNodeBoundTo="LeaveComments" Name="Leave a
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
comment"><Controls><UserLookUp FilterExpressions="{x:Null}" FilterDependencyIds="{x:Null}"
IsFilterExpressionSetForProperty="False" XmlNodeBoundTo="Name" Name="Name"
LabelPosition="Top"
Id="U1"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></UserLookUp><TextInput
XmlNodeBoundTo="Comments" Name="Comments" Id="T8" TextLine="MultiLine" Rows="5"
LabelPosition="Top"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Dro
pDownChoice XmlNodeBoundTo="Rating" Name="Rating" Id="D2"><Options><C:ColoredDictionary
Key="Below Average" Value="Below Average" ForeColor="#ea111c" BackColor=""
/><C:ColoredDictionary Key="Average" Value="Average" ForeColor="#f1ea18" BackColor=""
/><C:ColoredDictionary Key="Good" Value="Good" ForeColor="#f0a226" BackColor=""
/><C:ColoredDictionary Key="Excellent" Value="Excellent" ForeColor="#58ee1d" BackColor=""
/></Options><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></DropDownChoice></Contr
ols></PanelForm><Tabs ShowTabs="True" FormTitle="" ProductVersion="" BuildVersion=""
Id="T5" FormMode="NextGenMode"
XmlNodeBoundTo=""><OnTabIndexChangedScript><![CDATA[]]></OnTabIndexChangedScript><Controls>
<Tab FormTitle="" ProductVersion="" BuildVersion="" Id="T7" FormMode="NextGenMode"
XmlNodeBoundTo="" Name="View others comments"><Controls><BaseForm FormTitle=""
ProductVersion="" BuildVersion="" Id="P2" FormMode="JCSMode"
XmlNodeBoundTo="ReviewComments" Name="Review Comments" ViewMode="Grid" AddPosition="Bottom"
DefaultRows="1"><Controls><UserLookUp FilterExpressions="{x:Null}"
FilterDependencyIds="{x:Null}" IsFilterExpressionSetForProperty="False"
XmlNodeBoundTo="Name" Name="Name" LabelPosition="Top"
Id="U2"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></UserLookUp><TextInput
XmlNodeBoundTo="Comments" Name="Comments" Id="T9" TextLine="MultiLine" Rows="5"
LabelPosition="Top"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Dro
pDownChoice XmlNodeBoundTo="Rating" Name="Rating" Id="D3"><Options><C:ColoredDictionary
Key="Below Average" Value="Below Average" ForeColor="#ea111c" BackColor=""
/><C:ColoredDictionary Key="Average" Value="Average" ForeColor="#f1ea18" BackColor=""
/><C:ColoredDictionary Key="Good" Value="Good" ForeColor="#f0a226" BackColor=""
/><C:ColoredDictionary Key="Excellent" Value="Excellent" ForeColor="#58ee1d" BackColor=""
/></Options><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></DropDownChoice></Contr
ols><OnRowAddedScript><![CDATA[]]></OnRowAddedScript><OnRowDeletedScript><![CDATA[]]></OnRo
wDeletedScript></BaseForm></Controls></Tab></Controls></Tabs></Controls></BaseForm>
```

Telematics Article Submission Form.xml

```
<BaseForm SaveButtonText="@@Finish@@" FormTitle="Telematics Article Submission"
ProductVersion="1" BuildVersion="40107400" Id="NF1" FormMode="NextGenMode"
XmlNodeBoundTo="TelematicsArticle" Name="Telematics Articles" xmlns="clr-
namespace:Skelta.Forms.Core.Controls;assembly:Skelta.Forms.Core" xmlns:L="clr-
namespace:System.Collections.Generic;assembly:mscorlib" xmlns:C="clr-
namespace:Skelta.Forms.Core.Classes;assembly:Skelta.Forms.Core"><ScriptCode><![CDATA[]]></S
criptCode><Controls><TextInput IsMandatory="True" XmlNodeBoundTo="Title" Name="Title"
Id="T1"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><DropDownChoice
XmlNodeBoundTo="Category" Name="Category" Id="D1"><Options><C:ColoredDictionary
Key="Programming" Value="Programming" ForeColor="" BackColor="" /><C:ColoredDictionary
Key="Database" Value="Database" ForeColor="" BackColor="" /><C:ColoredDictionary
Key="Graphics" Value="Graphics" ForeColor="" BackColor="" /><C:ColoredDictionary
Key="Process Improvement" Value="Process Improvement" ForeColor="" BackColor=""
/></Options><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></DropDownChoice><TextIn
put XmlNodeBoundTo="Overview" Name="Overview" Id="T2" TextLine="MultiLine" Rows="5"
LabelPosition="Top"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Tex
tInput XmlNodeBoundTo="Contents" Name="Contents" Id="T3" TextLine="MultiLine" Rows="5"
LabelPosition="Top"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput><Tex
tInput XmlNodeBoundTo="Summary" Name="Summary" Id="T4" TextLine="MultiLine" Rows="5"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
LabelPosition="Top"><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput></Co
ntrols></BaseForm>
```

ArticleReview_Workflow.xml

```
<workflow name="" version="2">
<diagramdatacollection>
<diagramdata
key="DesignerInstanceId"><![CDATA[3de93f688a3144fdab82455fe8d75f8b]]></diagramdata>
</diagramdatacollection>
<action name="Add Article to List" type="UpdateVariableValue"
displayname="UpdateVariableValue" description="" looponoutput="">
<properties>
<property name="Set Variable">
<contentupdationfields>
<contentupdationfield name="XmlVariables.TelematicsArticle.TelematicsArticle" operator="="
mode="ObjTree"
methodName="bf7caa75c6424912ac138bf4ae06bbe2"><![CDATA[XmlVariables.SFFormData.TelematicsAr
ticle]]></contentupdationfield>
<classname><![CDATA[497ab399b39843028ae22fc387df2ecf]]></classname>
<jstring><![CDATA[[{"RHSMode":"ObjTree",
"LHSColumn":"XmlVariables.TelematicsArticle.TelematicsArticle",
"RHSColumn":"XmlVariables.SFFormData.TelematicsArticle",
"OPRColumn":"",
"ColumnExpression":{"DisplayExpressionString":"",
"ActualExprtessionString":"",
"MethodParameterList":"",
"HtmlVerexpString":"",
"ReferenceList":"",
"ReferencedFunctionList":"",
"NodeInfo":""}}]]]></jstring>
</contentupdationfields>
</property>
<property name="StopOnExceptions"><![CDATA[Yes]]></property>
<property name="RaiseError">
<raiseerror>
<enableraiseerrors><![CDATA[False]]></enableraiseerrors>
<value><![CDATA[]]></value>
<excludederroroutputs><![CDATA[]]></excludederroroutputs>
<includederroroutputs><![CDATA[]]></includederroroutputs>
<stoponerror><![CDATA[True]]></stoponerror>
<stoponexceptionpropertyoverrided><![CDATA[False]]></stoponexceptionpropertyoverrided>
</raiseerror>
</property>
</properties>
<visual-attributes>
<left>275</left>
<isloop>false</isloop>
<height>65</height>
<top>75</top>
<width>65</width>
</visual-attributes>
</action>
<action name="Article Send for Review" type="Invoke Form" displayname="Invoke Form"
description="" looponoutput="">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<properties>
<property name="Action Display Name"><![CDATA[]]></property>
<property name="Delivery Channels">
<propertychannel selectedchannelscount="1">
<channels>
<channel name="WebWorkList">
<config url="" />
<forms>
<form name="Invoke Form" type="InvokeWebForm">
<actors grantaccess="True" />
<roles grantaccess="True" />
</form>
</forms>
</channel>
<removedchannels list="" />
</channels>
</propertychannel>
</property>
<property name="Work Item Fields">
<Workitemfields />
</property>
<property name="To">
<resource>
<providername><![CDATA[]]></providername>
<globalfilter><![CDATA[]]></globalfilter>
<dispglobalfilter><![CDATA[]]></dispglobalfilter>
<filter><![CDATA[[Role] = ["ArticleSubscribers"]]]></filter>
<dispfilter><![CDATA[[SK Role] = ["ArticleSubscribers"]]]></dispfilter>
<multipleproviderfilter><![CDATA[[Role] =
["ArticleSubscribers"]]]></multipleproviderfilter>
<multipleproviderdispfilter><![CDATA[[SK Role] =
["ArticleSubscribers"]]]></multipleproviderdispfilter>
<isuserconfiguredthroughuserlookup><![CDATA[False]]></isuserconfiguredthroughuserlookup>
<usersconfiguredthroughuserlookup><![CDATA[]]></usersconfiguredthroughuserlookup>
<isuserconfiguredthroughrolelookup><![CDATA[False]]></isuserconfiguredthroughrolelookup>
<rolesconfiguredthroughrolelookup><![CDATA[]]></rolesconfiguredthroughrolelookup>
<isuserconfiguredthroughsubmitter><![CDATA[False]]></isuserconfiguredthroughsubmitter>
</resource>
</property>
<property name="Assign Queue">
<queuevalue>
<value><![CDATA[]]></value>
<valueID><![CDATA[]]></valueID>
</queuevalue>
</property>
<property name="Min. Acknowledgements [% or Number]"><![CDATA[1]]></property>
<property name="Max. Acknowledgements [% or Number]"><![CDATA[100%]]></property>
<property name="Enable TakeOwnership"><![CDATA[No]]></property>
<property name="Forwarding Allowed?"><![CDATA[No]]></property>
<property name="Ignore Unavailable Resources?"><![CDATA[No]]></property>
<property name="Alternate Resource Allowed?"><![CDATA[Yes]]></property>
<property
name="FormDetails"><![CDATA[TelematicsArticlesReview?0?53bb5ab3-9fa9-4046-a4d8-3b8644b9fce1
?Article Send for Review]]></property>
<property name="InputInstanceXML"><![CDATA[TelematicsArticle]]></property>
<property name="FormParameters">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<contentupdationfields>
<classname><![CDATA[ee9afe5385504d75aa662275e774341b]]></classname>
<jstring><![CDATA[]]></jstring>
</contentupdationfields>
</property>
<property name="SaveAsDraft"><![CDATA[Yes]]></property>
<property name="CloseOnSaveAsDraft"><![CDATA[Yes]]></property>
<property name="MaximizeOnLoad"><![CDATA[No]]></property>
<property name="MultiSelectFormDetails"><![CDATA[]]></property>
<property name="ResponsesFormDetails"><![CDATA[]]></property>
<property name="Subject">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[Review :<%#XmlVariables.TelematicsArticle.TelematicsArticle.Title%>
]]></value>
<valueHTML><![CDATA[Review
:&lt;%#XmlVariables.TelematicsArticle.TelematicsArticle.Title%&gt;]]></valueHTML>
<classname><![CDATA[ee64d9dbe447484ebcfb15960cbcfe01]]></classname>
</workflowmemovalue>
</property>
<property name="Notes">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[2f7cb22a5ed448b0a1f6297d399e10ab]]></classname>
</workflowmemovalue>
</property>
<property name="WorkitemDetailViewURL"><![CDATA[]]></property>
<property name="CloseOnFormSubmit"><![CDATA[No]]></property>
<property name="From Email Address"><![CDATA[admin@domain.com]]></property>
<property name="Show Custom Document View Window"><![CDATA[0]]></property>
<property name="Priority"><![CDATA[Medium]]></property>
<property name="Send Notification Email"><![CDATA[No]]></property>
<property name="Notify if alternate considered"><![CDATA[No]]></property>
<property name="Redirect URL">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[http://<%ApplicationPath%>/]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[d19574f3113a487d9f5a166b7efd2203]]></classname>
</workflowmemovalue>
</property>
<property name="Custom Document View URL">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[http://<%ApplicationPath%>/]]></value>
<valueHTML><![CDATA[]]></valueHTML>
<classname><![CDATA[fb6c2d3e9b874300abba9fa334f4a006]]></classname>
</workflowmemovalue>
</property>
<property name="OutputInstanceXML"><![CDATA[TelematicsArticle]]></property>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<property
name="AdvancedResponseStorage"><![CDATA[<ResponseStorage><IndRespNode>TelematicsArticle.Lea
veComments</IndRespNode><AllRespsNode>TelematicsArticle.ReviewComments</AllRespsNode></Resp
onseStorage>]]></property>
<property name="Hide Responses?"><![CDATA[0]]></property>
<property name="Affirmative Answer for Approval [% or Number]"><![CDATA[100%]]></property>
<property name="Action Limit[% or Number]"><![CDATA[100%]]></property>
<property name="Wait For All The Participants To Act"><![CDATA[No]]></property>
<property name="Decisions">
<Decisions>
<JSDecisonString><![CDATA[]]></JSDecisonString>
<CompletionMaker><![CDATA[]]></CompletionMaker>
<AffirmativeOutput><![CDATA[]]></AffirmativeOutput>
<AffirmativePercentage><![CDATA[]]></AffirmativePercentage>
<AffirmativeNegativeOutput><![CDATA[]]></AffirmativeNegativeOutput>
<classname><![CDATA[4c9aac58ef0248a6a8abc57fb3318611]]></classname>
</Decisions>
</property>
<property name="Set Calendar"><![CDATA[]]></property>
<property name="Minimum Time for Action"><![CDATA[0]]></property>
<property name="Timeout warning for Acknowledgement"><![CDATA[0]]></property>
<property name="Timeout for Acknowledgement"><![CDATA[0]]></property>
<property name="Timeout warning for Action"><![CDATA[0]]></property>
<property name="Timeout for Action"><![CDATA[0]]></property>
<property name="Multiple TimeOut Warning"><![CDATA[]]></property>
<property name="Enable Redirection?"><![CDATA[No]]></property>
<property name="RaiseError">
<raiseerror>
<enableraiseerrors><![CDATA[False]]></enableraiseerrors>
<value><![CDATA[]]></value>
<excludederroroutputs><![CDATA[]]></excludederroroutputs>
<includederroroutputs><![CDATA[]]></includederroroutputs>
<stoponerror><![CDATA[True]]></stoponerror>
<stoponexceptionpropertyoverrided><![CDATA[False]]></stoponexceptionpropertyoverrided>
</raiseerror>
</property>
</properties>
<visual-attributes>
<left>450</left>
<isloop>false</isloop>
<height>65</height>
<top>75</top>
<width>65</width>
</visual-attributes>
</action>
<action name="Start" type="Start" displayname="" description="" looponoutput="">
<properties>
<property name="Variables"><![CDATA[]]></property>
<property name="Contents"><![CDATA[]]></property>
<property name="XmlVariables">
<XmlVariables>
<XmlVariable>
<Name><![CDATA[SKEventData]]></Name>
<Description><![CDATA[]]></Description>
<StorageType><![CDATA[XmlDocument]]></StorageType>
<Scope><![CDATA[global]]></Scope>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<Schema><![CDATA[<?xml version="1.0" encoding="utf-16"?>
<xsd:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
attributeFormDefault="unqualified" elementFormDefault="qualified">
<xs:element name="SKEventData">
<xs:complexType>
<xs:sequence>
<xs:element name="EventProperties" />
<xs:element name="EventType" />
<xs:element name="EventProvider" />
<xs:element name="EventDateTimeUTC" type="xs:string" />
<xs:element name="EventDateTime" type="xs:string" />
<xs:element name="UserId" />
<xs:element name="UserName" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xsd:schema>]]></Schema>
<Xsl><![CDATA[]]></Xsl>
<Storage>
<InternalStorage><![CDATA[variables]]></InternalStorage>
</Storage>
</XmlVariable>
<XmlVariable>
<Name><![CDATA[TelematicsArticle]]></Name>
<Description><![CDATA[]]></Description>
<StorageType><![CDATA[XmlSkeltaList]]></StorageType>
<Scope><![CDATA[global]]></Scope>
<Schema><![CDATA[<?xml version="1.0" encoding="utf-16"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="TelematicsArticle">
<xs:complexType>
<xs:sequence>
<xs:element name="Title" type="xs:string" />
<xs:element name="Category" type="xs:string" />
<xs:element name="Overview" type="xs:string" />
<xs:element name="Contents" type="xs:string" />
<xs:element name="Summary" type="xs:string" />
<xs:element maxOccurs="unbounded" name="ReviewComments">
<xs:complexType>
<xs:sequence>
<xs:element name="Name" type="xs:string" />
<xs:element name="Rating" type="xs:string" />
<xs:element name="Comments" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="System_x003A_Internal">
<xs:complexType>
<xs:sequence>
<xs:element name="SaveAs">
<xs:complexType>
<xs:sequence />
<xs:attribute name="Published" type="xs:boolean" />
<xs:attribute name="NextVersionAsDraft" type="xs:boolean" />
<xs:attribute name="NextVersionAsPublished" type="xs:boolean" />
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<xs:attribute name="NextMajorVersionAsDraft" type="xs:boolean" />
<xs:attribute name="NextMajorVersionAsPublished" type="xs:boolean" />
<xs:attribute name="NextMinorVersionAsDraft" type="xs:boolean" />
<xs:attribute name="NextMinorVersionAsPublished" type="xs:boolean" />
<xs:attribute name="SpecificVersionAsDraft" type="xs:string" />
<xs:attribute name="SpecificVersionAsPublished" type="xs:string" />
<xs:attribute name="AsFolder" type="xs:boolean" />
</xs:complexType>
</xs:element>
<xs:element name="Operations">
<xs:complexType>
<xs:sequence />
<xs:attribute name="MoveToFolder" type="xs:string" />
<xs:attribute name="Delete" type="xs:boolean" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>]]></Schema>
<Xsl><![CDATA[]]></Xsl>
<Storage>
<ListName><![CDATA[Telematics Article]]></ListName>
<ListId><![CDATA[83f84415-86d6-4ef1-86a5-df9a58f2e515]]></ListId>
<FolderName><![CDATA[]]></FolderName>
<FolderId><![CDATA[00000000-0000-0000-0000-000000000000]]></FolderId>
<ListItemScope><![CDATA[New]]></ListItemScope>
<ListItemId><![CDATA[00000000-0000-0000-0000-000000000000]]></ListItemId>
<XmlTemplate><![CDATA[<?xml version="1.0"?><TelematicsArticle><Title /><Category
/><Overview /><Contents /><Summary /><ReviewComments><Name /><Rating /><Comments
/></ReviewComments><System_x003A_Internal><SaveAs><Published /><NextVersionAsDraft
/><NextVersionAsPublished /><NextMajorVersionAsDraft /><NextMajorVersionAsPublished
/><NextMinorVersionAsDraft /><NextMinorVersionAsPublished /><SpecificVersionAsDraft
/><SpecificVersionAsPublished /><AsFolder /></SaveAs><Operations><MoveToFolder /><Delete
/></Operations></System_x003A_Internal></TelematicsArticle>]]></XmlTemplate>
<ListQuery><![CDATA[]]></ListQuery>
<ListItemVersion><![CDATA[]]></ListItemVersion>
</Storage>
</XmlVariable>
<XmlVariable>
<Name><![CDATA[SFFormData]]></Name>
<Description><![CDATA[Generated by Skelta Forms]]></Description>
<StorageType><![CDATA[XmlDocument]]></StorageType>
<Scope><![CDATA[global]]></Scope>
<Schema><![CDATA[<?xml version="1.0" encoding="utf-16"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="TelematicsArticle">
<xs:complexType>
<xs:sequence>
<xs:element name="Title" type="xs:string" />
<xs:element name="Category" type="xs:string" />
<xs:element name="Overview" type="xs:string" />
<xs:element name="Contents" type="xs:string" />
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<xs:element name="Summary" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>]]></Schema>
<Xsl><![CDATA[ ]]></Xsl>
<Storage>
<InternalStorage><![CDATA[variables]]></InternalStorage>
</Storage>
</XmlVariable>
</XmlVariables>
</property>
<property name="ExecutionView"><![CDATA[]]></property>
<property name="SOAObjects">
<SOAObjects />
</property>
<property name="Initialize Variable(s)">
<contentupdationfields>
<classname><![CDATA[ddf7acf755974772bf5cfb0a3421da0a]]></classname>
<jstring><![CDATA[]]></jstring>
</contentupdationfields>
</property>
<property name="Raise Exception if Completion Maker Output is not linked?"><![CDATA[Default
[Backward Compatible]]]></property>
<property name="ExceptionTemplate">
<workflowmemovalue>
<height><![CDATA[350]]></height>
<width><![CDATA[650]]></width>
<value><![CDATA[]]></value>
<valueHTML><![CDATA[]]></valueHTML>
</workflowmemovalue>
</property>
</properties>
<visual-attributes>
<left>100</left>
<isloop>false</isloop>
<height>65</height>
<top>75</top>
<width>65</width>
</visual-attributes>
</action>
<link from="Start" to="Add Article to List" condition="" description="" name="Link0">
<visual-attributes>
<sourceport>r</sourceport>
<sourceportside>dynamic</sourceportside>
<sourceportindex>0</sourceportindex>
<sourceportfinalside>r</sourceportfinalside>
<sourceportfinalindex>0</sourceportfinalindex>
<sourceportfinalx>165</sourceportfinalx>
<sourceportfinaly>108</sourceportfinaly>
<targetport>l</targetport>
<targetportside>dynamic</targetportside>
<targetportindex>0</targetportindex>
<targetportfinalside>l</targetportfinalside>
<targetportfinalindex>0</targetportfinalindex>
<targetportfinalx>275</targetportfinalx>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<targetportfinaly>108</targetportfinaly>
<serializedpath>Link0;false;2;220;108;Link0,right,0,187,108,165,108,false,false,NaN,NaN,!,,
Link0_0,;Link0,,0,187,108,187,108,false,false,NaN,NaN,!,,Link0_1,;Link0,right,0,253,108,187
,108,false,true,220,108,!,,Link0_2,;Link0,,0,253,108,253,108,false,false,NaN,NaN,!,,Link0_3
,;Link0,right,0,275,108,253,108,true,false,NaN,NaN,!,,Link0_4,</serializedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
<link from="Add Article to List" to="Article Send for Review" condition="Updated"
description="" name="Link1">
<visual-attributes>
<sourceport>r</sourceport>
<sourceportside>dynamic</sourceportside>
<sourceportindex>0</sourceportindex>
<sourceportfinalside>r</sourceportfinalside>
<sourceportfinalindex>0</sourceportfinalindex>
<sourceportfinalx>340</sourceportfinalx>
<sourceportfinaly>108</sourceportfinaly>
<targetport>l</targetport>
<targetportside>dynamic</targetportside>
<targetportindex>0</targetportindex>
<targetportfinalside>l</targetportfinalside>
<targetportfinalindex>0</targetportfinalindex>
<targetportfinalx>450</targetportfinalx>
<targetportfinaly>108</targetportfinaly>
<serializedpath>Link1;false;2;395;108;Link1,right,0,362,108,340,108,false,false,NaN,NaN,!,,
Link1_0,;Link1,,0,362,108,362,108,false,false,NaN,NaN,!,,Link1_1,;Link1,right,0,428,108,362
,108,false,true,395,108,!,,Link1_2,;Link1,,0,428,108,428,108,false,false,NaN,NaN,!,,Link1_3
,;Link1,right,0,450,108,428,108,true,false,NaN,NaN,!,,Link1_4,</serializedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
</workflow>
```

Note: In multi-user scenarios to view the response details of other users ensure that the Input Instance XML name and Output Instance XML name is same for the Invoke Form activity.

Invoke Form - Update Form Parameters

A parameter represents a value that is passed to a variable. The parameter value is defined when the variable is declared in the procedure. A procedure can be defined with no parameters, one parameter, or more than one. The part of the procedure definition that specifies the parameters is called the parameter list.

Here, parameters represent the form variables where the user can define while designing the form. These parameters can be used in Forms scripts, in user look up controls to build the filter expression, and as a parameter for the lookup query.

Note: Form parameters values are not stored either in form definition or in database. These values are rendered when the form is rendered.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

How to define parameters: To work with parameters, the Form should have parameters defined. To define the Form parameter, on the corresponding Forms Designer screen, click **Edit properties of the form** on the top. Click the **Advanced** tab. You can define the parameter in the **Form Properties** section. You can define the parameter clicking the **Add Row**.

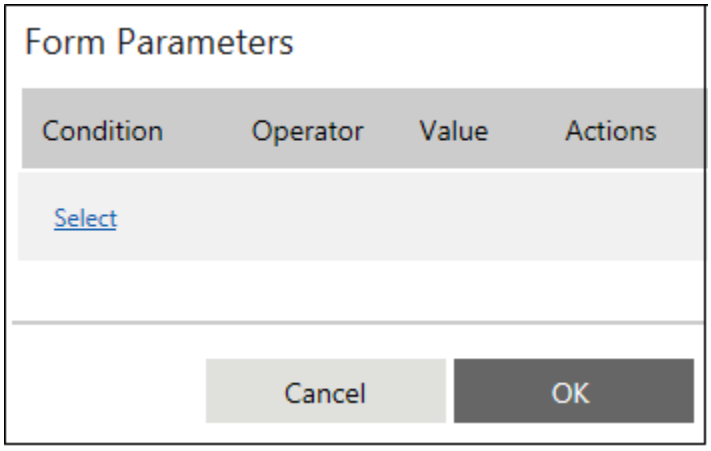
How Form parameters can be used?

Form parameter can be used in Form scripts, in user look up controls to build the filter expression, and as a parameter for the lookup query. Below is the sample scenario for how parameters can be used in form scripts.

Suppose in our form we have two controls Employee id & Manager Name. Based on the Employee id entered, we can populate the Manager name. To achieve this, we need to write the script (C#) to update the Manager name value from Form parameters. This parameter value can be updated from workflow.

Steps to set Update form parameters:

1. Click the edit button corresponding to **Update form parameters**. The following screen will be displayed.



2. To select a form parameter from the drop-down list, click the drop-down. The following screen is displayed.
3. Expand the tree by clicking on the '+' sign. The associated parameters of the form are displayed.
4. Select the desired parameter by clicking the name. The following screen will be displayed.
5. Click **Enter value** to display a text box. Enter the new value for the parameter in the text box.

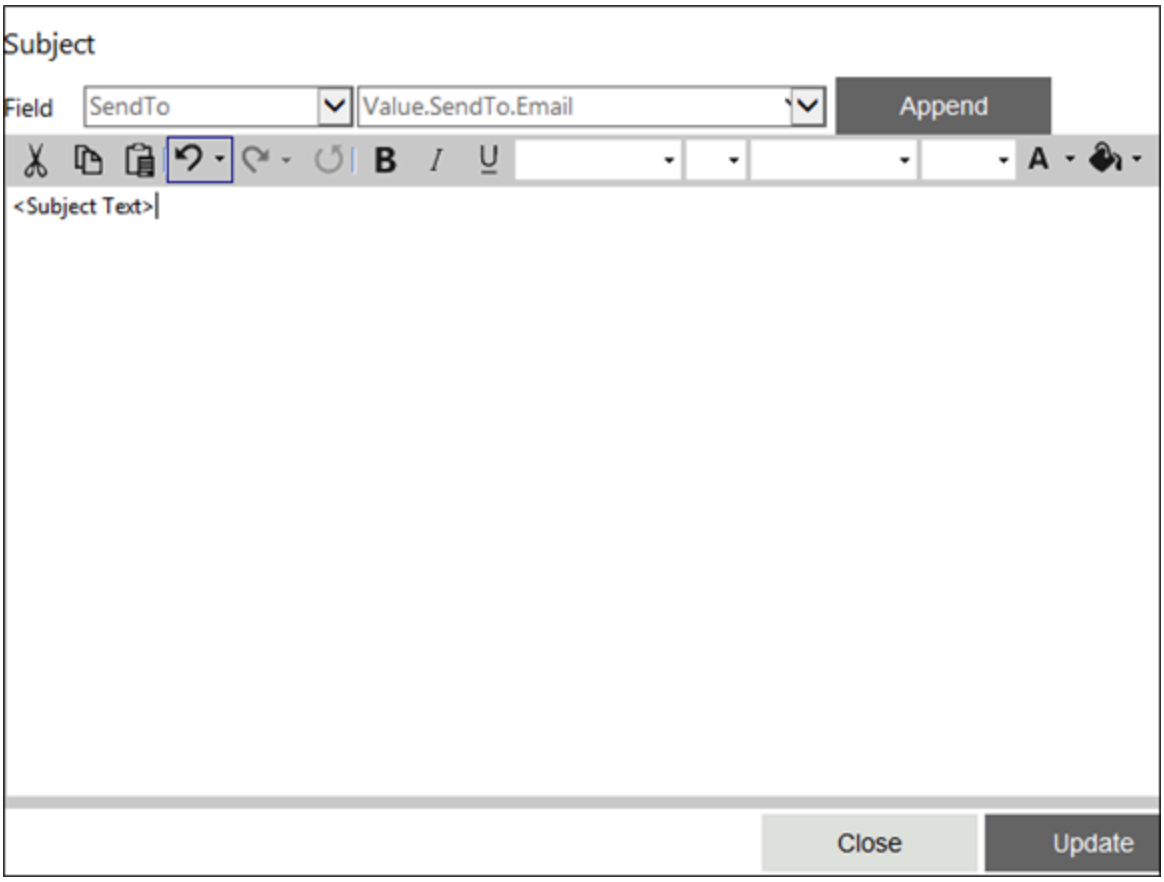
   You can also select a new value by clicking .
6. Select the desired new parameter to be updated by clicking it.
7. Click **OK**. The new parameter is updated.

Invoke Form - Subject

Subject Property Window

You can specify a subject for the Invoke Form WorkItem assignment in this property window. To access this window, click the button in the 'Subject' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the subject. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/ drop down - You can specify the actual value of the subject. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the subject in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the subject.
- **Append** - You can click on this button to add the specified value, variable or content to the text editor.

Tip: You can specify the subject using a combination of discrete value and variable or content. For example, in the image above, the subject entered is "Submitted By <%#Variable.SubmittedBy.Email%>". Here, the string "Submitted By" has been typed directly in the text editor. The second half of the subject is a placeholder, which will be replaced by the variable value during run-time. This has been added by selecting the appropriate variable option from the **Field** drop down and clicking on **Append**.

The text tool bar above the text editor allows you to manage and format the subject. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

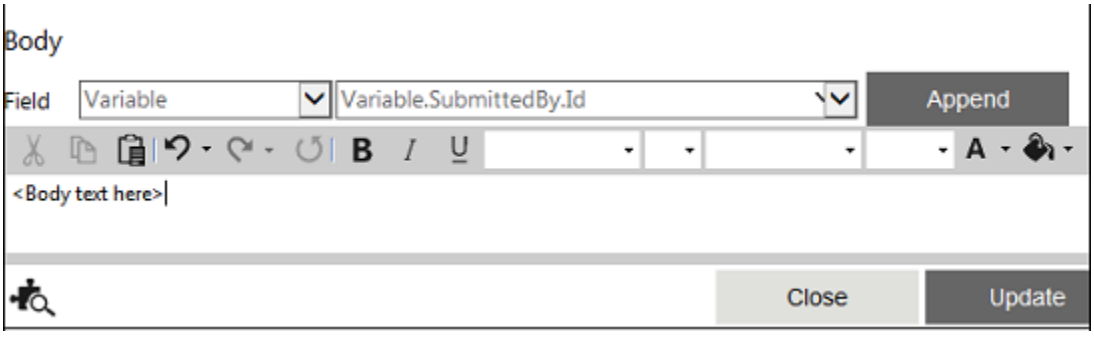
At the bottom of the window, you will find the following buttons:

- **Update** - You can click this button to save the subject.
- **Close** - You can click this button to close this window.

Invoke Form - Body

Body Property Window

You can specify a message Body for the Invoke Form work item assignment in this property window. To access this window, click the button in the 'Body' property.



The window has a text editor to enter the body text. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/ drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.

- **Append** - You can click this button to add the specified value, variable or content to the text editor.

You can specify the body using a combination of discrete value and variable or content. For example, in the image above, the body message entered is "Hi, Please approve the leave application submitted by <%#Variable.SubmittedBy.name%>". Here the string "Hi, Please approve the leave application submitted by" has been typed directly in the text editor. The second half of the body is a placeholder, which will be replaced by the variable value during run-time. This has been added by selecting the appropriate variable option from the Field drop down and clicking Append.

The text tool bar above the text editor allows you to manage and format the body text. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting and other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

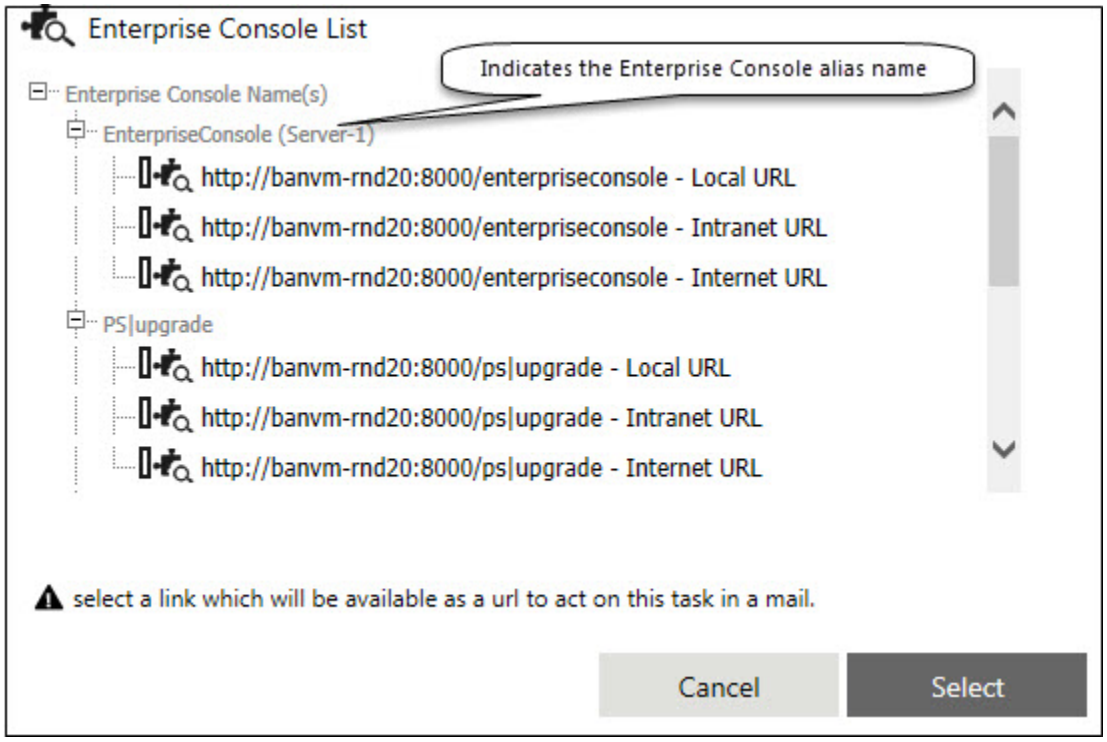
- **Update** - You can click this button to save the body message.
- **Close** - You can click this button to close this window.

To compose the body text, do the following steps:

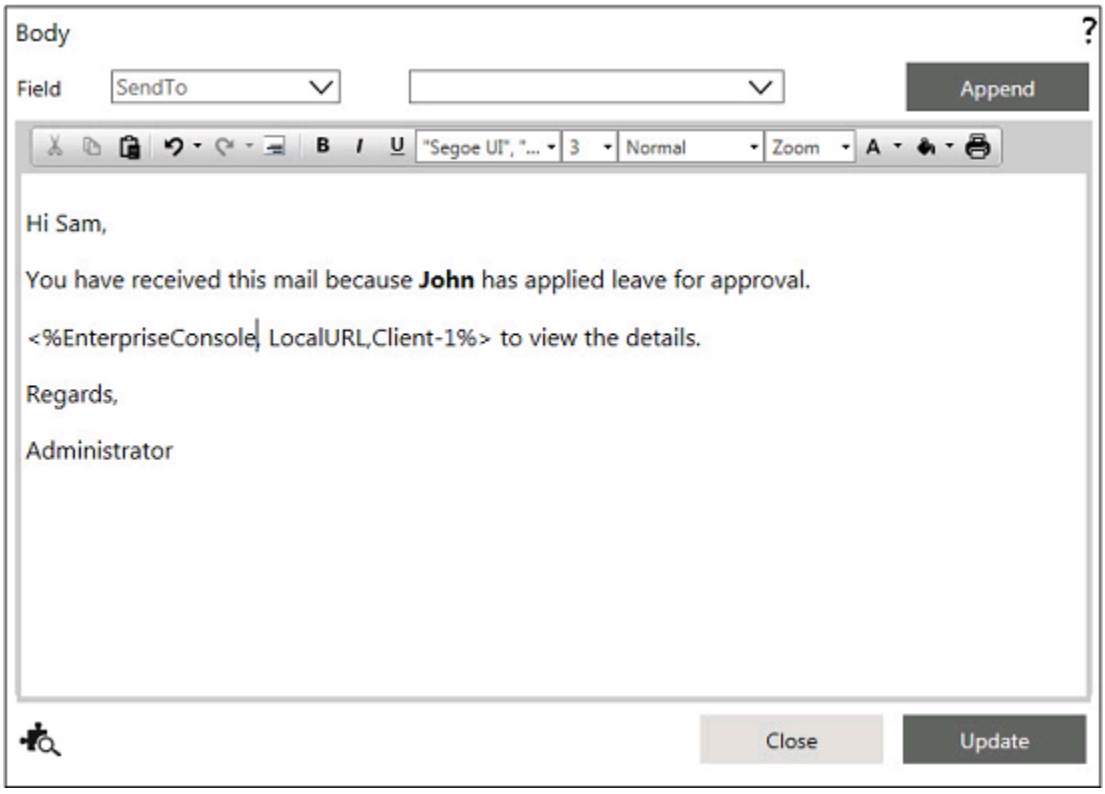
- **Body:** Click the edit icon to open the **Body** property window.
  1. Compose the body text.
  2. To get a direct link to the AVEVA Work Tasks Form as an email, click **Enterprise Console Explorer**(plug-in)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

icon to select the required URL and then click **Select** as shown in the following figure. See ResourceSet - Subject for more information.



The composed mail is shown in the following figure



The above composed mail will be displayed both in Email as well as in the Work item Details View page.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: If you do not want to display the above composed mail, Click Here text information in the Work items Details View page, then enclose the above text within "<!== ==>" as shown below.

<!==
Hi Sam,
You have received this mail because **John** has applied leave for approval.
<%EnterpriseConsole, LocalURL,Client-1%> to view the details.
Regards,
Administrator
==>

1. Click **Update** of the Body window property to save the contents.

Invoke Form - Template Path

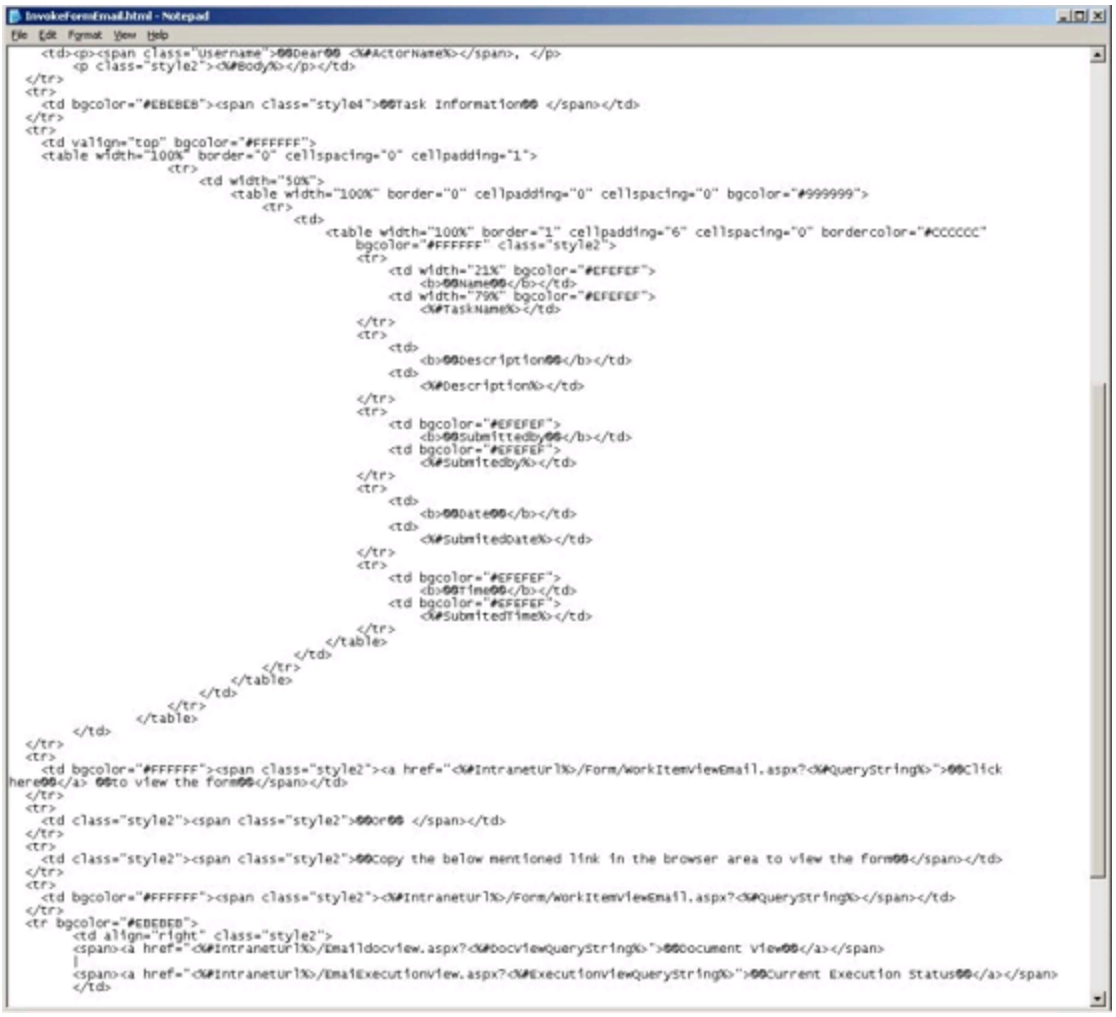
Standard Template

If you are using the template in general, then you can use the generic template which is placed in the physical path across the workflow.

The standard template is available in *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Templates\FormEmailTemplate* folder.

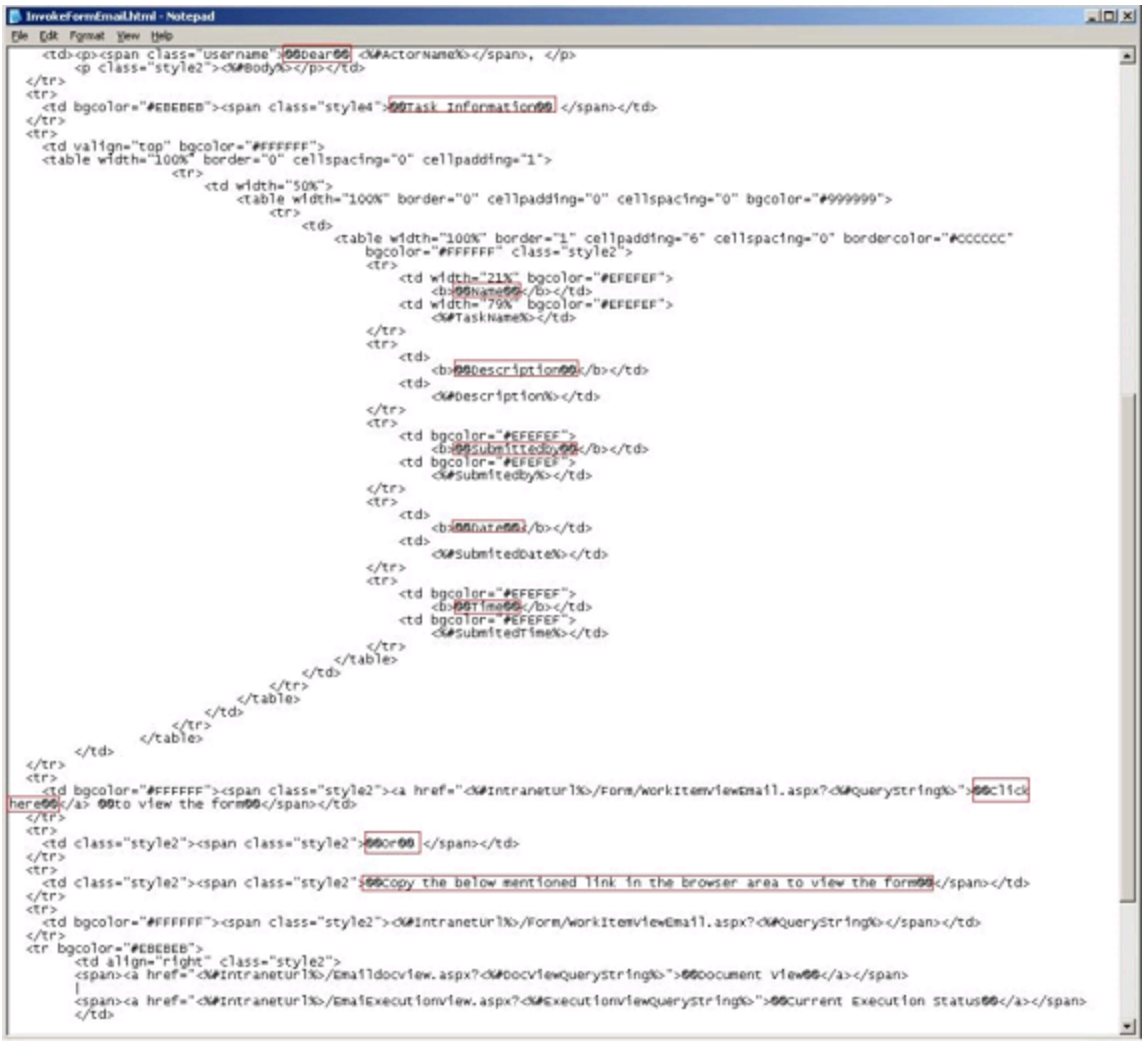
The standard template 'InvokeFormEmail.html' is shown in the figure below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The text enclosed within @@ are taken from Locales folder *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Locales\en-US* and you can customize these text. These are marked in the figure below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



To customize text within '@@', follow the steps given below:

1. Go to the following path *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Locales\en-US*
2. Open the required file and customize the required text.

Customizing Template

If you want to specify the template for different activities then you can create different templates for different activities. You can store the created templates wherever engine is running. Example: You can create a manger template for manager, Employee template for employee, leave application template for all employees and salary template etc.

Note: In an Enterprise Server scenario, if there are two servers running, then the template path will be considered for which engine has got the request.

To create a new template, there are two ways:

First Option:

Create a HTML file and place it in any place wherever the engine is running. Enter the entire path in the Template Path property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: Make sure that, while creating a template, you should use the marked text in the template in order to get the Work Items from AVEVA Work Tasks.

Second Option:

Take a copy of **'InvokeFormEmail.html'** and place it in any physical path and customize the text however you want.

The following table depicts the Placeholder which you can customize in your template:

Placeholder	Description	Example
<%#ActorName%>	Name of the user to whom the email has to be sent, using the template.	Sam
<%#Submitedby%>	Name of the user who submitted the workflow	John
<%#Body%>	The content specified in the Body property will be reflected.	Please fill a form below to process your tax declaration.
<%#TaskName%>	Activity name	Leave Application Form
<%#Description%>	Activity description	Detailed leave information of the employee
<%#SubmitedDate%>	Date when workflow was submitted	12/10/2008
<%#SubmitedTime%>	Time when workflow was submitted	04:00:24
<%#LocalUrl%>	Enterprise Console local URL to send form link as an email.	http://localhost:8000/enterpriseconsole/BPMUITemplates/Default/Email/
<%#IntranetUrl%>	Enterprise Console Intranet URL to send form link as an email.	http://machinename:8000/enterpriseconsole/BPMUITemplates/Default/Email/
<%#InternetUrl%>	Enterprise Console Internet URL to send form link as an email.	http://www.skelta.com/enterpriseconsole/BPMUITemplates/Default/Email/
<%#ApplicationPath%>	Enterprise Console path will be considered from SKAddInProviderSettings table for the respective repository. If you are not getting the mail from SMTP exchange server , you need to do some manual entry in the SKAddInProviderSettings table.	http://skeltanb24:8000/enterpriseconsole/BPMUITemplates/Default/Email/

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Placeholder	Description	Example
	In the Settings column of the SKAddInProviderSettings table, change the entry 'http://[ApplicationPath]/BPMUITemplates/Default/Email/' to your local machine name/IP address with port. Example: Change ApplicationPath to EnterpriseConsole url In the Application column of the SKAddInProviderSettings table, change the entry 'Default' to repository name. Example: Change Application to EmployeeRepo	
<%#QueryString%>	The query string which contains the Work Item information in encrypted format.	mcuri=uSR0tBXvLACdY2J-e1EPXjPewxFYxLx6G5Rg2zPgzpr8EYiyP63v6QSY0jdq26OnLOrwfy8iDwUhUUCdeEsOzKWgI4D7MF_9Y-LWRlb05T0pfbBP12oTqK0hRvyZsnCBxDeX190sSTkyt2mGY3nCDzCkVPrLMu82msEhQdm_g8HeO5SY_fWhiWgGDKjPai0O
<%#DocViewQueryString%>	Document View page details	Emaildocview.aspx?mailviewuri=nLOVXOa2m4x43xcG4kWt0Zi7gsePioiAfQsnck92w-Fr-uMXWj5scuZy-xVNkzCQhOKoFPTpRGuRQobt4kbSlJD4TXQXi6e9mDkPXOrR5UGBQiGOq88IvRJqleRkwK5PXKgWPIbdLChdMYl1Xl5IIebMMjeVsHi4KXFuiw9Maa1vp2805TjpiBAbODvy1dO6NX8oBEaoiKu1XiHUKX1Lwpd2nkCb26C_jEpvKDBKAwHvqpHWZViDFoDlKmUa8YvgHfrm3Xf-xAWy02z5IJ4WvPUe6kgQoFwLDleRfF-W6hw!
<%#ExecutionViewQueryString%>	Process Execution page details	EmaiExecutionView.aspx?mailviewuri=nLOVXOa2m4x43xcG4kWt0Zi7gsePioiAfQsnck92w-Fr-uMXWj5scuZy-xVNkzCQhOKoFPTpRGuRQobt4kbSl

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Placeholder	Description	Example
		JD4TXQXi6e9mDkPXOrR5UGBQiGOq88IvRJqleRkwK5PXKgWPIbdLChdMYl1Xl5IIebMMjeVsHi4KXFuiw9Maa1vp2805TjpiBAbODvy1dO6ROMtM-aBMo5QlRzS6Ad9MrGhjuLxkHy4ebMtncEUuH3zXPnGUpSzT-qRBduN4iOWF53PJF1MHI_nIyLmNBJloxMz5RqzgDfKM8BLur3QOwtc3lUZBT0M-JKH-EYpbAfgDHUq9uhDWhBbBhMsxYN09M30_NyJmL20HiBVbDQrxwHQrZrxBLpQRlJKSymutzHM
For configuring workflow variables	Using Variables in Invoke Form Activity	
For configuring workflow contents	Using Contents in Invoke Form Activity	
For configuring workflow Xml variables	Using XML Variables in Invoke Form Activity	

Invoke Form Activity Email Template

AVEVA Work Tasks Engine accesses the approval activity's email template from the following physical path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Templates\FormEmailTemplate\InvokeFormEmail.html

This template can be modified as per the UI requirement. The mails received by the actor is based on the above mentioned mail template.

Below is the default email template text for the **InvokeFormEmail.html** file of the mail received by the actor.

InvokeFormEmail.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title></title>
<style type="text/css">
<!--
.style1 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 14px;
}
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight: bold; }

-->
</style>
</head>

<body>
<table width="700" border="0" cellpadding="0" cellspacing="1" bgcolor="#CCCCCC" >
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><p><span class="Username">@@Email_Salutation@@ <%#ActorName%></span>, </p>
<p class="style2"><%#Body%></p></td>
</tr>
 <tr>
<td bgcolor="#FFFFFF" class="style2"> </td>
 </tr>
<tr>
<td bgcolor="#EBEBEB"><span class="style4">@@Email_TaskInformation@@</span></td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="0" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b>@@Email_TaskName@@</b></td>
```

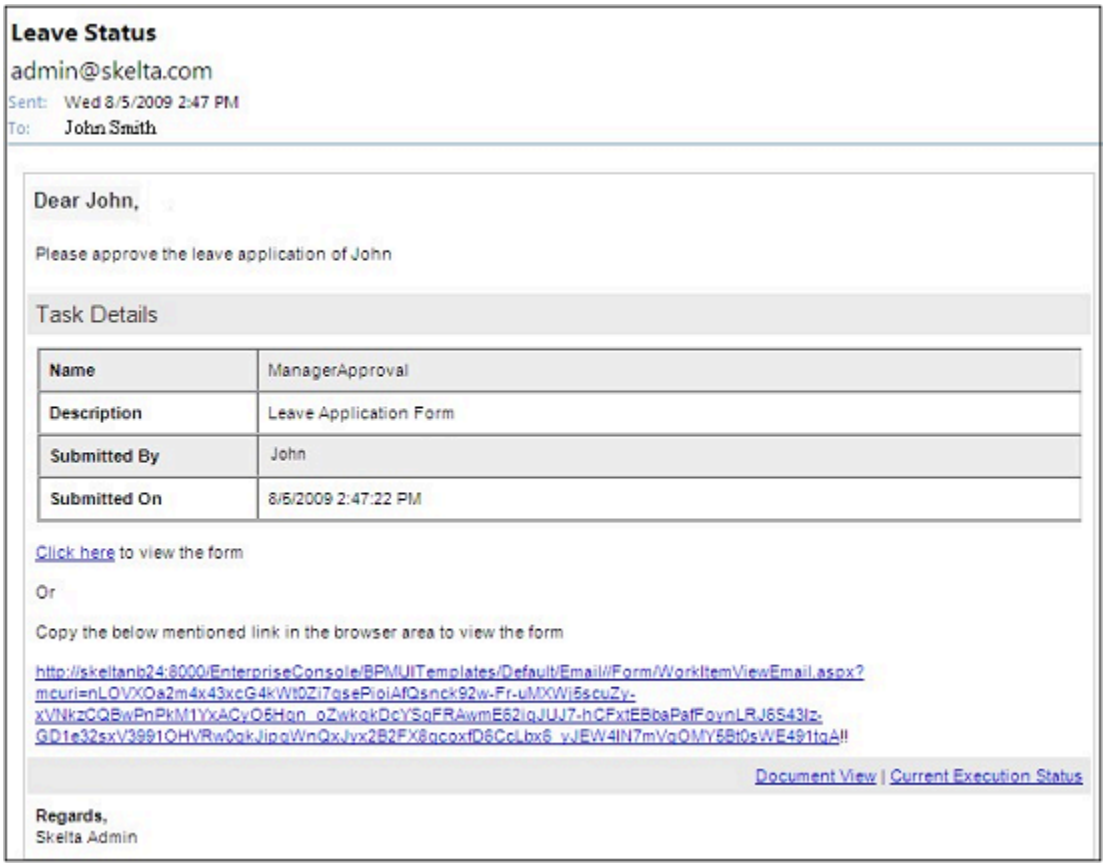
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td width="79%" bgcolor="#EFEFEF">
  <%#TaskName%></td>
</tr>
<tr>
<td>
<b>@@Email_Description@@</b></td>
<td>
  <%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>@@Email_Submitedby@@</b></td>
<td bgcolor="#EFEFEF">
  <%#Submitedby%></td>
</tr>
<tr>
<td>
<b>@@Email_SubmitedOn@@</b></td>
<td>
  <%#SubmitedOn%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
 </td>
</tr>
<tr>
<td bgcolor="#FFFFFF"><span class="style2"><a href="<%#IntranetUrl%>/Form/
WorkItemViewEmail.aspx?<%#QueryString%>">@@Click here@@</a> @@to view the
form@@</span></td>
</tr>
<tr>
<td class="style2"><span class="style2">@@Or@@ </span></td>
</tr>
<tr>
<td class="style2"><span class="style2">@@Copy the below mentioned link in the browser area
to view the form@@</span></td>
</tr>
<tr>
<td bgcolor="#FFFFFF"><span class="style2"><%#IntranetUrl%>/Form/
WorkItemViewEmail.aspx?<%#QueryString%></span></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a
href="<%#IntranetUrl%>/Emaildocview.aspx?<%#DocViewQueryString%>">@@Email_Document_ViewURLT
ext@@</a></span>
|
<span><a
href="<%#IntranetUrl%>/EmaiExecutionView.aspx?<%#ExecutionViewQueryString%>">@@Email_Execut
ion_ViewURLText@@</a></span>
</td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2">@@Email_Signature@@</td>
</tr>
</td>
</tr>
</table>
</body>
</html>
```

The following figure shows the Invoke Form activity Email template:



Note:

The QueryString placeholder is mandatory and should not be modified.

```
<a href="<%#IntranetUrl%>/Form/WorkItemViewEmail.aspx?<%#QueryString%>">@@Click here@@</a>
```

There are four types of placeholders. They are :

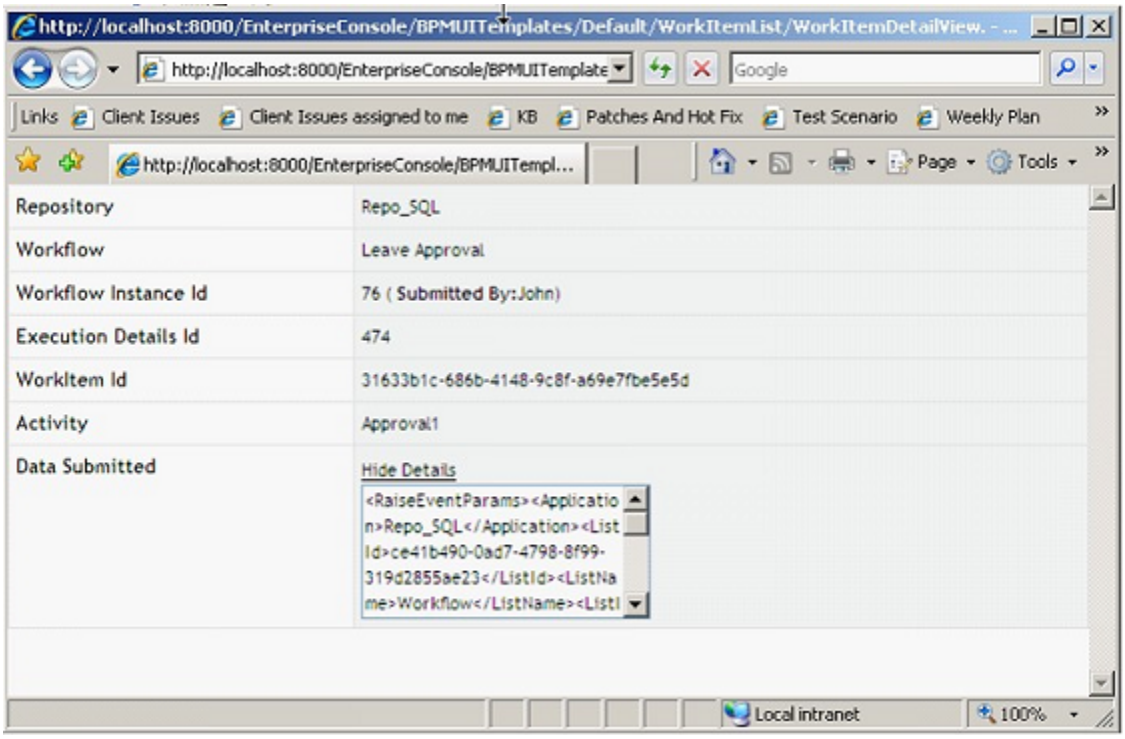
```
<%#LocalUrl%>, <%#IntranetUrl%>, <%#InternetUrl%> and <%#ApplicationPath%>.  You should use
at least any one of these placeholders based on the requirement.  By default
the <%#IntranetUrl%> is used as shown below.
<a href="<%#IntranetUrl%>/Form/WorkItemViewEmail.aspx?<%#QueryString%>">@@Click here@@</a>
```

Document View:

When you click the **Document View** link, the document view with execution details is displayed as shown in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

following figure.



Document View

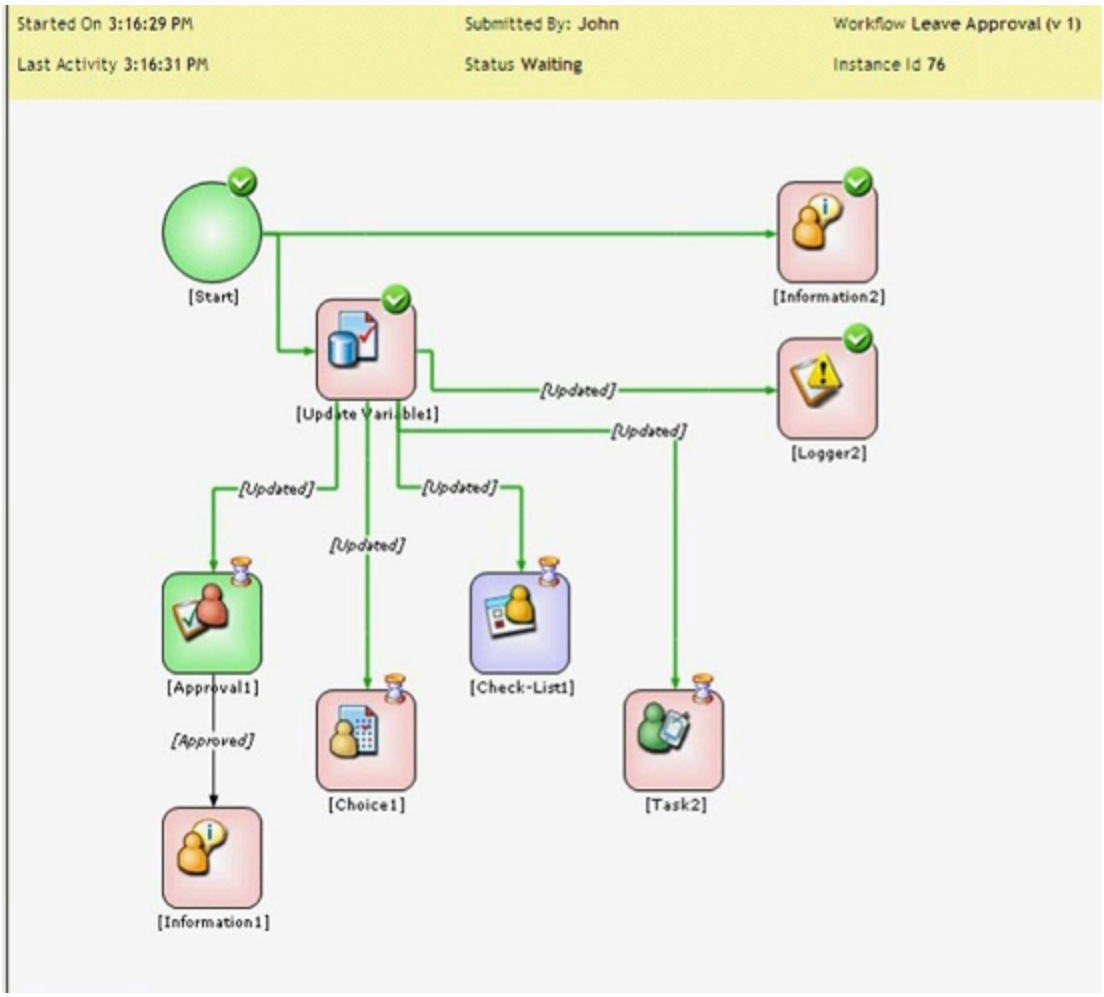
Current Execution Status:

An actor will be able to access the **Process Execution View** and their work item from an Email which is sent to the actor.

Pre-requisites:

- To access the **Process Execution View**, you have to be connected to the application server via LAN or WAN or Internet. In case, not connected to the application server, then the Process Execution View will not be accessible. When you click the **Current Execution Status** link, the **Process Execution View** of the **Analysts View** will be displayed as shown below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Use Variables in Invoke Form Activity

Email templates can be used to display the values of the variables that have been created in a workflow.

Scenario

Scenario Requirements:

Assume that, variable of type string by name '**Status**' is created in **Start** activity and its value is updated as **"Variable, status is updated".**

You want to,

1. Display the variable's value in Invoke Form email.
2. Display the variable's value after "Submitted On" row in the Task Details Table.

Scenario Solution

To display this variable's value in Invoke Form email and to display the variable's value after "Submitted On" row in the Task Details Table, the following steps need to be followed:

Steps:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Copy **InvokeFormEmail.html** file.
2. Use the **<%#Variable.**Variablename%> placeholder in the html template for viewing the variable's value. Where <Variablename> is a name of the variable. Since the variable name is "Status", the placeholder to be used is <%#Variable.Status%> where the value is to be displayed. The highlighted portion in the below code displays the position where the **<%#Variable.Status%>** placeholder is added.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title></title>
<style type="text/css">
<!--
.style1 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 14px;
}
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight: bold;
}

-->
</style>
</head>

<body>
<table width="700" border="0" cellpadding="0" cellspacing="1" bgcolor="#CCCCCC" >
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

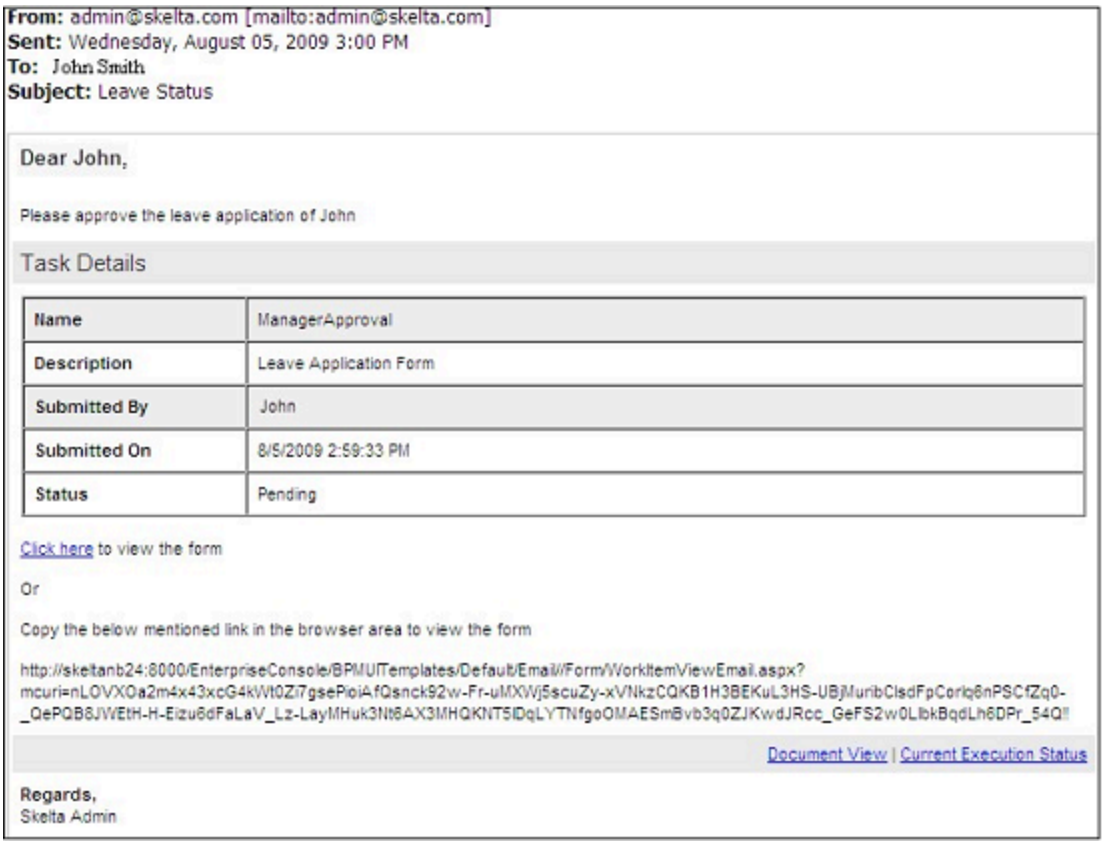
```
<td><p><span class="Username">@@Dear@@ <%#ActorName%></span>, </p>
<p class="style2"><%#Body%></p></td>
</tr>
<tr>
<td bgcolor="#EBEBEB"><span class="style4">@@Task Information@@ </span></td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="0" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b>@@Name@@</b></td>
<td width="79%" bgcolor="#EFEFEF">
  <%#TaskName%></td>
</tr>
<tr>
<td>
<b>@@Description@@</b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>@@Submittedby@@</b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b>@@Date@@</b></td>
<td>
<%#SubmitedDate%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>@@Time@@</b></td>
<td bgcolor="#EFEFEF">
<%#SubmitedTime%></td>
</tr>
 <tr>
<td>
<b>Status</b></td>
<td>
<%#Variable.Status%></td>
</tr>
</table>
</td>
</tr>
</table>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</td>
</tr>
</table>
 </td>
</tr>
<tr>
<td bgcolor="#FFFFFF"><span class="style2"><a href="<%#IntranetUrl%>/Form/
WorkItemViewEmail.aspx?<%#QueryString%>">@@Click here@@</a> @@to view the
form@@</span></td>
</tr>
<tr>
<td class="style2"><span class="style2">@@Or@@ </span></td>
</tr>
<tr>
<td class="style2"><span class="style2">@@Copy the below mentioned link in the browser
area to view the form@@</span></td>
</tr>
<tr>
<td bgcolor="#FFFFFF"><span class="style2"><%#IntranetUrl%>/Form/
WorkItemViewEmail.aspx?<%#QueryString%></span></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a href="<%#IntranetUrl%>/Emaildocview.aspx?<%#DocViewQueryString%>">@@Document
View@@</a></span>
|
<span><a
href="<%#IntranetUrl%>/EmaiExecutionView.aspx?<%#ExecutionViewQueryString%>">@@Current E
xecution Status@@</a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><strong>@@Regards@@,</strong><br />
@@Skelta Admin@@</td>
</tr>
</td>
</tr>
</table>
</body>
</html>
```

The actor will get the following Invoke Form email format for the above template.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Note: Out of box variables such as <%#Variable.SubmittedBy.UserId%> , <%#Variable.SubmittedBy.UserName%>, and <%#Variable.SubmittedBy.UserDisplayName%> can be used for Active Directory provider.

Use Contents in Invoke Form Activity

Email templates can be used to display the values of the contents that have been created in a workflow.

Scenario

Scenario Requirements:

Assume that, a **Content** of type number and by name 'EmployeeAge' is created in Start activity and at runtime its value is 30.

You want to:

1. Display the content's value in Invoke Form email.
2. Display the variable's value after "Submitted On" row in the Task Details Table.

Scenario Solution

To display this the content's value in Invoke Form email and to display the variable's value after "Submitted On" row in the Task Details Table, the following steps need to be followed:

Steps:

1. Copy **InvokeFormEmail.html** file.
2. Use the **<%#Variable.**Contentname%> placeholder in the html template for viewing the content's value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Where <Contentname> is a name of the content. Use the **<%#Content.EmployeeAge%>** placeholder in the html template, where the value is to be displayed. Since the Content name is "EmployeeAge", the placeholder to be used is <%#Content.EmployeeAge%>. The highlighted portion in the below code displays the position where the <%#Content.EmployeeAge%> placeholder is added.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title></title>
<style type="text/css">
<!--
.style1 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 14px;
}
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight: bold;
}

-->
</style>
</head>

<body>
<table width="700" border="0" cellpadding="0" cellspacing="1" bgcolor="#CCCCCC" >
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><p><span class="Username">@@Dear@@ <%#ActorName%></span>, </p>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<p class="style2"><%#Body%></p></td>
</tr>
<tr>
<td bgcolor="#EBEBEB"><span class="style4">@@Task Information@@ </span></td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="0" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b>@@Name@@</b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b>@@Description@@</b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>@@Submittedby@@</b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b>@@Date@@</b></td>
<td>
<%#SubmitedDate%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>@@Time@@</b></td>
<td bgcolor="#EFEFEF">
<%#SubmitedTime%></td>
</tr>
 <tr>
<td>
<b>Employee Age</b></td>
<td>
<%#Content.EmployeeAge%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>
</table>
 </td>
</tr>
<tr>
<td bgcolor="#FFFFFF"><span class="style2"><a href="<%#IntranetUrl%>/Form/
WorkItemViewEmail.aspx?<%#QueryString%>">@@Click here@@</a> @@to view the
form@@</span></td>
</tr>
<tr>
<td class="style2"><span class="style2">@@Or@@ </span></td>
</tr>
<tr>
<td class="style2"><span class="style2">@@Copy the below mentioned link in the browser
area to view the form@@</span></td>
</tr>
<tr>
<td bgcolor="#FFFFFF"><span class="style2"><%#IntranetUrl%>/Form/
WorkItemViewEmail.aspx?<%#QueryString%></span></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a href="<%#IntranetUrl%>/Emaildocview.aspx?<%#DocViewQueryString%>">@@Document
View@@</a></span>
|
<span><a
href="<%#IntranetUrl%>/EmaiExecutionView.aspx?<%#ExecutionViewQueryString%>">@@Current
Execution Status@@</a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><strong>@@Regards@@,</strong><br />
@@Skelta Admin@@</td>
</tr>
</td>
</tr>
</table>
</body>
</html>
```

The actor will get the following mail format for the above template:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

From: admin@skelta.com [mailto:admin@skelta.com]
Sent: Wednesday, August 05, 2009 3:01 PM
To: John Smith
Subject: Leave Status

Dear John,

Please approve the leave application of John

Task Details

Name	ManagerApproval
Description	Leave Application Form
Submitted By	John
Submitted On	8/5/2009 3:01:26 PM
Employee Age	30

Click here to view the form

Or

Copy the below mentioned link in the browser area to view the form

http://skeltanb24:8000/EnterpriseConsole/BPMUITemplates/Default/Email/Form/WorkItemViewEmail.aspx?mcuri=nLOVXOa2m4x43xcG4kWt0Zi7gsePioiAfQsnck92w-Fr-uMXWj5scuZy-xVNkzCQgAhxg7DAehGoI-CZBdiZBhWQlGn3VzOp_2BwYfiL-ozNe6APiA5tj7fXETJPw8_GQD-_j0jJoNxcH41HA7Ebdi3GxTtNTRdYZb80B30dUlcDX2XNilrUkmSNsXe6-lQWO9fqgABSrGR-TJs7cOZ8lw!!

Document View | Current Execution Status

Regards,
Skelta Admin

Use XML Variables in Invoke Form Activity

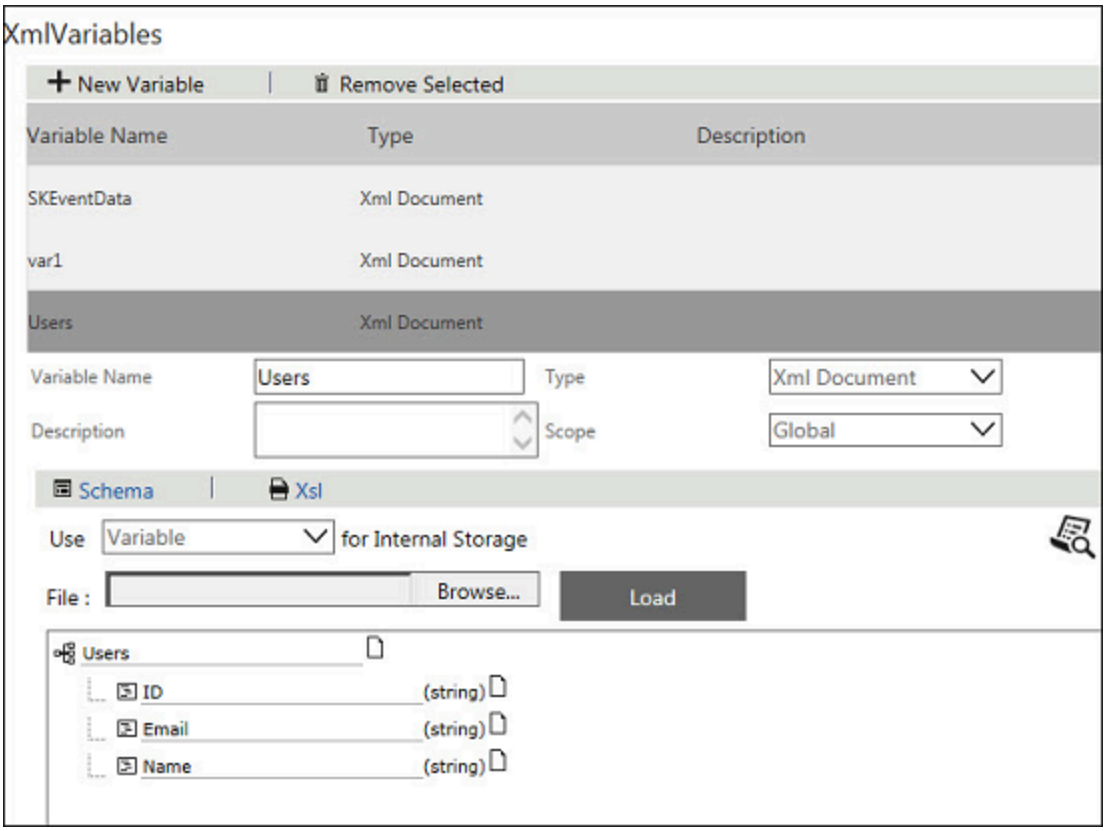
Email templates can be used to display the values of the XML variables that have been created in a workflow.

Scenario

Scenario Requirements:

Assume that, 'Users' XML Variable of type XMLDocument is created in **Start** activity and schema is as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The scenario requirement is that, during the workflow execution, you want to display the following values during the workflow execution.

1. **ID** is updated to 1029.
2. **Email** is updated to John.Smith@domain.com.
3. **Name** is updated to John.
4. Display these XML variable's value after **"Submitted On"** row in the Task Details Table.

Scenario Solution

To display the XML Variable's value in Invoke Form email and to display these variable's value after "Submitted on" row in the Task Details Table, the following steps need to be followed:

Steps:

1. Copy **InvokeFormEmail.html** file.
2. Use the **<%#XmlVariables.**variablename**%>** placeholder in the html template for viewing the XML Variable's value. Where <variablename> is a name of the XML variable.

   Since the 'Users' XML Variable contains the ID, Email, and Name, use the following XML Variables in the InvokeFormEmail.html template, where the values are to be displayed.

   ```
   <%#XmlVariables.Users.User.ID%>
   <%#XmlVariables.Users.User.Email%>
   <%#XmlVariables.Users.User.Name%>
   ```

   The highlighted portion in the below code displays the position where the above placeholders to be added.

   ```
   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
   xhtml1/DTD/xhtml1-transitional.dtd">
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title></title>
<style type="text/css">
<!--
.style1 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 14px;
}
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight: bold;
}

-->
</style>
</head>

<body>
<table width="700" border="0" cellpadding="0" cellspacing="1" bgcolor="#CCCCCC" >
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><p><span class="Username">@@Dear@@ <%#ActorName%></span>, </p>
<p class="style2"><%#Body%></p></td>
</tr>
<tr>
<td bgcolor="#EBEBEB"><span class="style4">@@Task Information@@ </span></td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

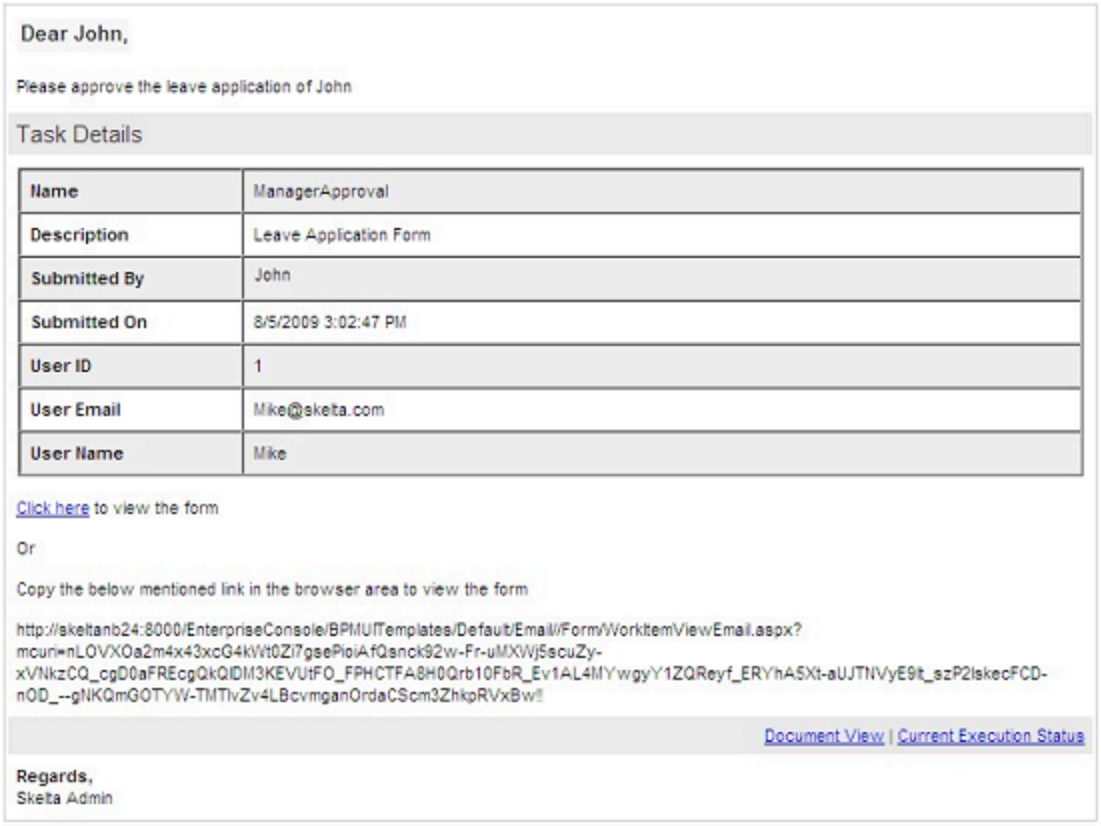
```
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="0" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b>@@Name@@</b></td>
<td width="79%" bgcolor="#EFEFEF">
  <%#TaskName%></td>
</tr>
<tr>
<td>
<b>@@Description@@</b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>@@Submittedby@@</b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b>@@Date@@</b></td>
<td>
<%#SubmitedDate%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>@@Time@@</b></td>
<td bgcolor="#EFEFEF">
<%#SubmitedTime%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>User ID</b></td>
<td bgcolor="#EFEFEF">
<%#XmlVariables.Users.User.ID%></td>
</tr>
<tr>
<td>
<b>User Email</b></td>
<td>
<%#XmlVariables.Users.User.Email%></td>
</tr>
<tr>
<td  bgcolor="#EFEFEF">
<b>User Name</b></td>
<td  bgcolor="#EFEFEF">
<%#XmlVariables.Users.User.Name%></td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
 </td>
</tr>
<tr>
<td bgcolor="#FFFFFF"><span class="style2"><a href="<%#IntranetUrl%>/Form/
WorkItemViewEmail.aspx?<%#QueryString%>">@@Click here@@</a> @@to view the
form@@</span></td>
</tr>
<tr>
<td class="style2"><span class="style2">@@Or@@ </span></td>
</tr>
<tr>
<td class="style2"><span class="style2">@@Copy the below mentioned link in the browser
area to view the form@@</span></td>
</tr>
<tr>
<td bgcolor="#FFFFFF"><span class="style2"><%#IntranetUrl%>/Form/
WorkItemViewEmail.aspx?<%#QueryString%></span></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a href="<%#IntranetUrl%>/Emaildocview.aspx?<%#DocViewQueryString%>">@@Document
View@@</a></span>
|
<span><a
href="<%#IntranetUrl%>/EmaiExecutionView.aspx?<%#ExecutionViewQueryString%>">@@Current
Execution Status@@</a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><strong>@@Regards@@,</strong><br />
@@Skelta Admin@@</td>
</tr>
</td>
</tr>
</table>
</body>
</html>
```

The actor will get the following mail format for the above template:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.


Dear John,

Please approve the leave application of John

Task Details

Name	ManagerApproval
Description	Leave Application Form
Submitted By	John
Submitted On	8/5/2009 3:02:47 PM
User ID	1
User Email	Mike@skelta.com
User Name	Mike

Click here to view the form

Or

Copy the below mentioned link in the browser area to view the form

http://skeltanb24:8000/EnterpriseConsole/BPMUITemplates/Default/Email/FormWorkItemViewEmail.aspx?mcuri=nLOVXOa2m4x43xcG4kWt0Zi7gsePioiAfQsnck92w-Fr-uMXWj5scuZy-xVNkzCQ_cgD0aFREcgQkQIDM3KEVUtFO_FPHCTFA8H0Qrb10FbR_Ev1AL4MYwgyY1ZQReyf_ERYhA5Xt-aUJTNVyE9lt_szP2IskecFCD-nOD_--gNKQmGOTYW-TMTlvZv4LBcvmganOrdaCScm3ZhkpRVxBw!!

Document View | Current Execution Status

Regards,
Skelta Admin

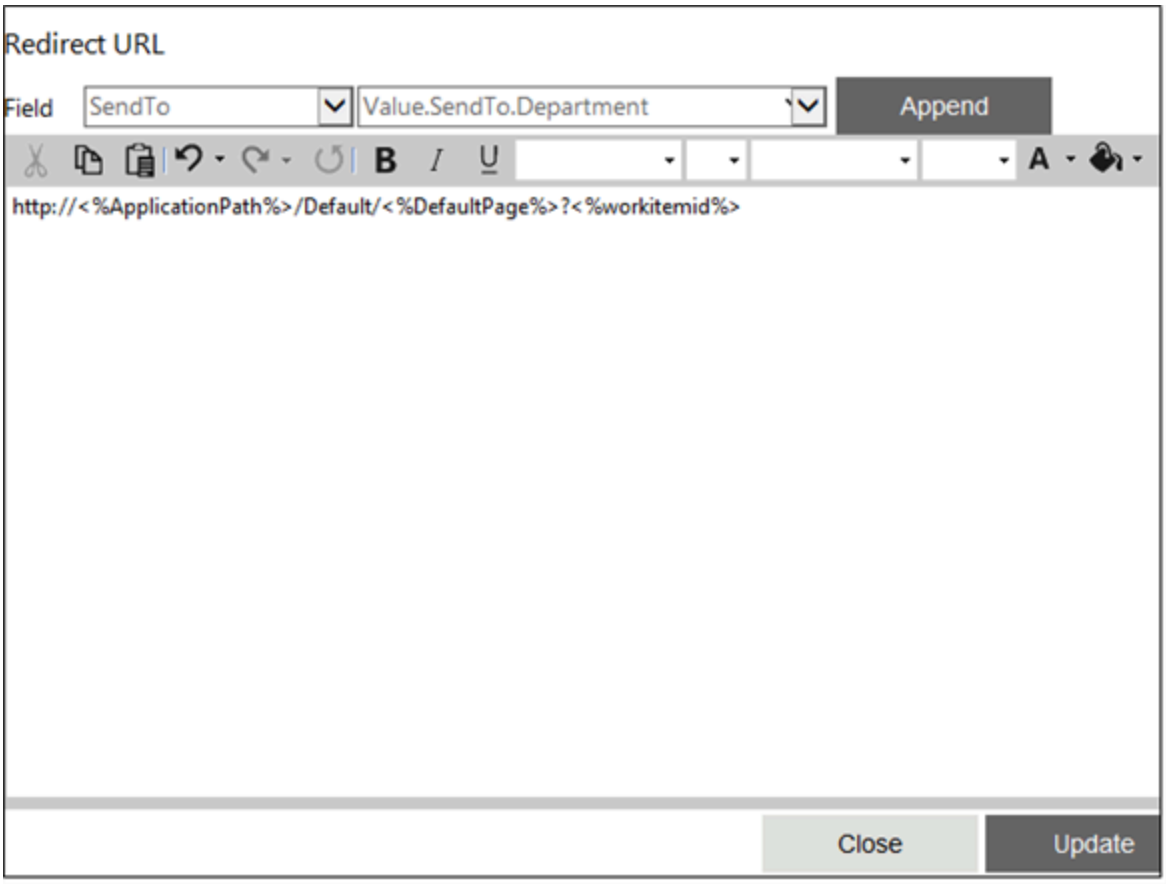
Invoke Form - Redirect URL

Redirect URL Property Window

You can specify a message Redirect URL for the Invoke Form activity in this property window. To access this window, click the button in the 'Redirect URL' property.

Note: The redirection feature may not work on the Application Request Routing (ARR) server.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the Redirect URL. You can do this either by entering a URL directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/ drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Redirect URL in this box. If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Redirect URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Redirect URL. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to redo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

It also has the following formatting and other tools –

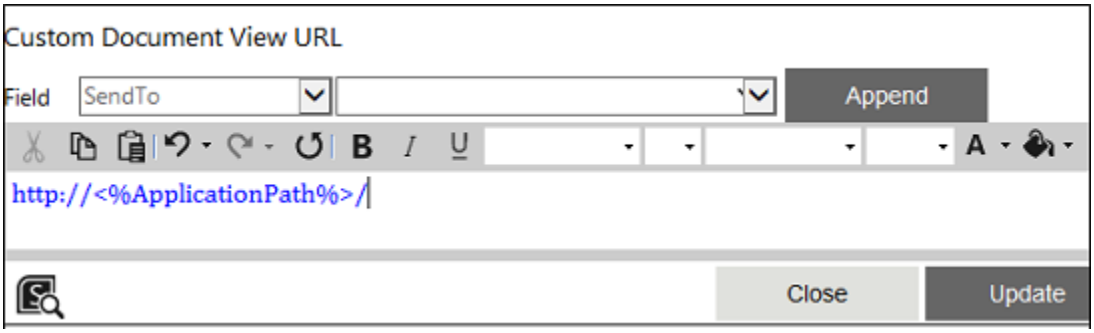
- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the Redirect URL.
- **Close** - You can click this button to close this window.

Invoke Form - Custom Document View URL

You can specify a message in the **Custom Document View URL** for the Invoke Form activity in this property window. To access this window, click the edit button for the 'Custom Document View URL' property.



The **Custom Document View URL** window has a text editor to enter the Custom Document View URL. You can either enter a URL directly in the text editor or use the following **Field** controls to enter the URL:

- **Field Type**: Select the value type. You can specify SendTo, Variable, Content, or XMLVariables.
- **Field Text**: You can specify the actual value of the Redirect URL.

  The control display depends on the option selected in the first drop-down menu. For any type selected in the first drop-down menu, there will be another drop-down menu in the second field. The second drop-down will have the variable collection, content collection, or XMLVariables collection from which you can select an appropriate option to assign the Custom Document View URL.
- **Append**: Click this button to add the specified value, variable, or content to the text editor.

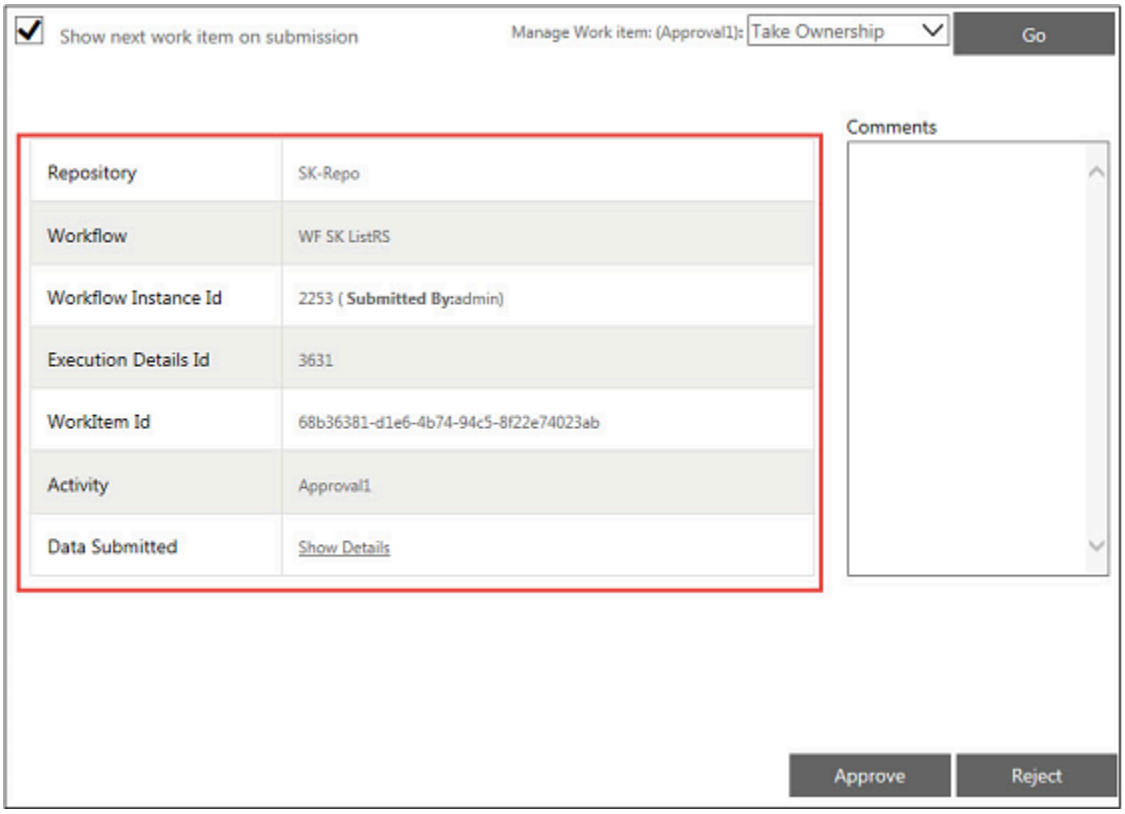
For more details, see Custom Document View URL.

Steps to set the Custom Document View URL property

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Custom Document View

Custom Document View can be viewed on the left side of the WorkItem Detail View page inside the frame as shown in the following image:



AVEVA Work Tasks allows you to provide a custom document view while using any human activities. This can be achieved by entering the proper URL for the Custom Document View URL property.

By default, the custom document view for any human resource activity normally shows the item metadata which alerted the workflow.

1. If the workflow is alerted due to a Enterprise Console event association, the document view shows the list item metadata that alerted the workflow. For example, if the workflow is alerted due to a workflow event association to a list, document view shows the list item that alerted the workflow.
2. If the workflow is alerted due to a workflow association in Forms, the document view shows the form that alerted the workflow.

You can change the document view, if required. If a URL is provided to the custom document view URL property, this is overridden with the default document view. If you want to show a different form in the Custom Document View, see *Custom Document View URL for Forms*.

Any URL can be provided to this property and it can be a URL to your custom page also.

Custom Document View URL for Forms

If the workflow is alerted due to a workflow association in forms, the document view shows the form that alerted the workflow. AVEVA Work Tasks supports the following URL with parameters. You can pass the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

parameters based on the requirement. Do not enter any internet links.

Use the following URL with respective parameters to show any forms:

http://<%ApplicationPath%>/BPMUITemplates/Default/SkeltaForms/FormDocumentView.aspx?FormName=<Form Name>&FormVersion=<Form Version>&FormInstanceXmlVariable=<Xml Variable Name>

Note: Remove the extra spaces between 'http:' and '//<%ApplicationPath%>', if any.

- FormName: Name of the form to be loaded in document view.
- FormVersion: Version of the form to be loaded. By default the latest published version will be loaded. This parameter need to be used only if a specific version has to be loaded.
- FormInstanceXMLVariable: XML variable name which is to be used for populating the form. If this parameter is not passed, SFFormData XML Variable will be considered as default.

Example:

For showing any form and populate data from XML variable, pass the following parameters in query string provided both having same XML schemas.

```
FormName=<Form Name>& FormInstanceXmlVariable=<Xml Variable Name>
```

Sample URL:

http://<%ApplicationPath%>/BPMUITemplates/Default/SkeltaForms/FormDocumentView.aspx?FormName=Leave Report&FormInstanceXmlVariable=LeaveReportxml

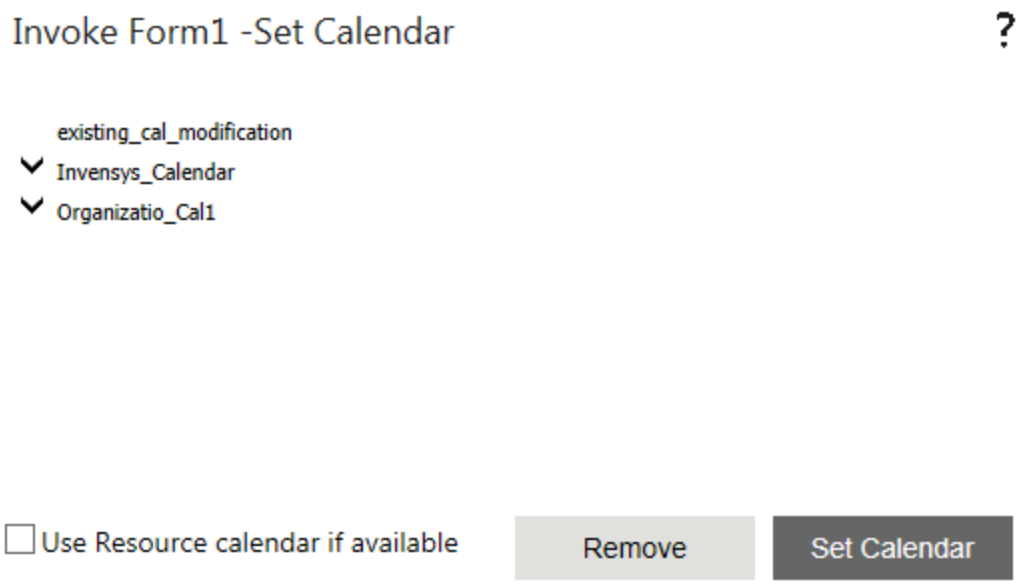
When the workflow is executed, the **Custom Document View** of the specified form populated using XML variable is displayed in the Work Item detail View page.

Invoke Form - Set Calendar for Timeout

Set Calendar for Timeout Property Window

You can set a Calendar to calculate the timeout for the Invoke Form activity in this property window. To access this window, click the button in the 'Set Calendar for Timeout' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Set Calendar for Timeout property window

This window lists the available calendars in a tree structure, with lower-level calendars nested within the respective top-level calendar. You can click on a calendar to select it for setting.

There are two buttons in this window.

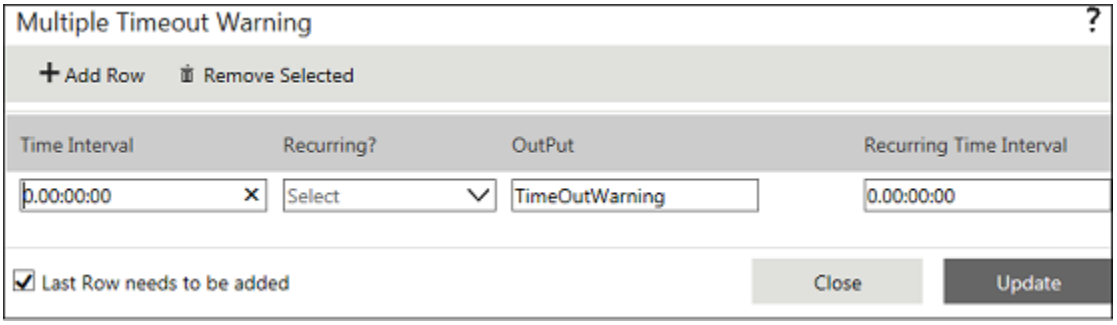
- **No Calendar** - You can click this button to indicate that there is no calendar set for the activity
- **Set Calendar** - You can click this button to set the selected calendar.

The **User Resource calendar if available** check box can be checked to enable the use of resource calendars, if none are displayed in this window for selection.

Invoke Form - Multiple Timeout Warning

Multiple TimeOut Warning Property Window

You can set the Multiple Timeout Warning in this property window. To access this window, click the button for the 'Multiple TimeOut Warning' property.



The window has two tabs at the top to manage the rows of timeout settings in the grid.

- **Add Row** - You can click this tab to add the newly entered timeout setting and display a new row to the grid

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

where you can add another timeout

- **Remove Selected** - You can click this tab to remove the timeout setting row selected in the grid.

The grid has the following parameters that you need to specify for the new timeout setting.

- **Time Interval** - You can use this field to enter the time interval in which you want to generate a timeout warning. You must enter the time in d.hh:mm:ss format.
- **Recurring?** - You can use the drop down in this field to indicate whether the timeout is recurring or not. You can select "Yes" to set the timeout warning to recur.
- **Output** - You can enter the output message corresponding to the timeout warning here.
- **Recurring Time Interval** - You can set the time interval in which the timeout warning has to recur here. Enter the time in d.hh:mm:ss format.

There are two buttons and a check box at the bottom of the screen.

- **Update** - You can click this button to save the newly added timeout settings
- **Close** - You can click this to close the property window without saving the newly added timeout settings
- **Last Row needs to be added** - You need to check this check box to save the last row added.

Invoke Form - Mail Integration Scenario

AVEVA Work Tasks provides a powerful feature to integrate the invoked form in a mail. This is useful for user to get a direct link to the Form as an email. Without logging into Enterprise Console, the user can click the link that will open Forms in a browser.

Pre-requisites

1. To configure your machine IP address for receiving emails, see Configuring Wonderware Skelta BPM Farm Configuration Wizard for Receiving Emails.
2. The **Send Notification Email** property should be set to **Yes** for this activity.

Scenario Description

Consider a scenario where the manager has to approve or reject the leave application, sent by the subordinates. Where the manager requires direct link to access this leave application form which is filled by the subordinate as an email.

Scenario Requirements

Design a form called 'Leave Form' with following controls and invoke this form using Invoke Form activity.

- Employee Name (Text Input control)
- From Date (Date Input control)
- To Date (Date Input control)
- Reason (Text Input Control)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Grid
- BaseForm
- Approve (Button)
- Reject (Button)

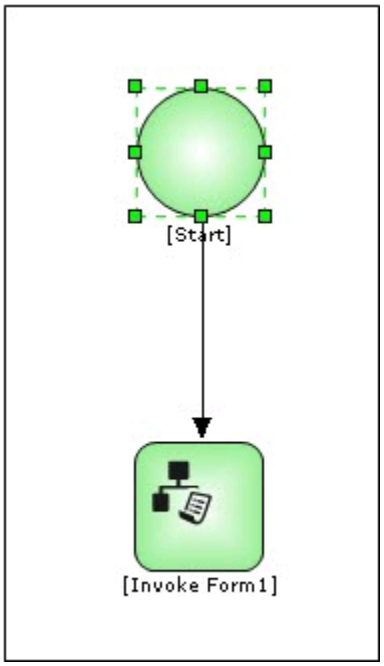
There are two ways to send notification mail:

1. Using Body property
2. Using Template Path property

Use 'Body' Property

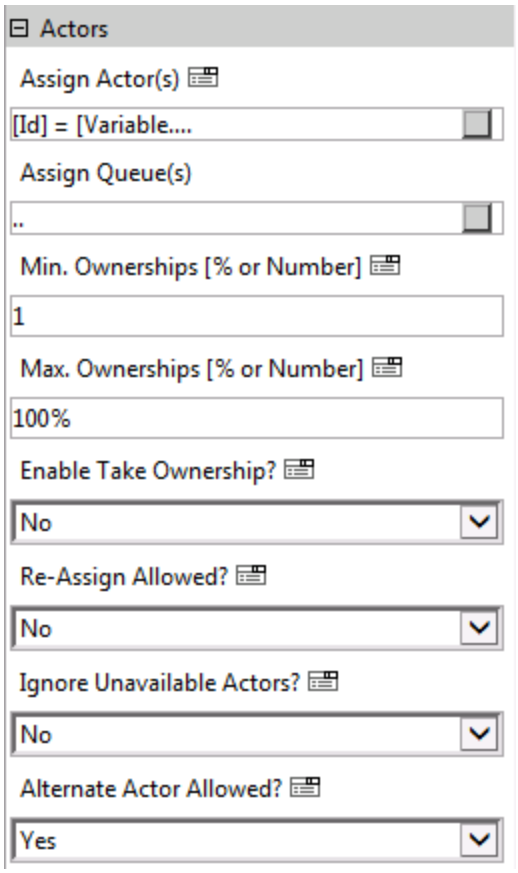
Scenario Solution:

1. Design a workflow named 'LeaveWF'.

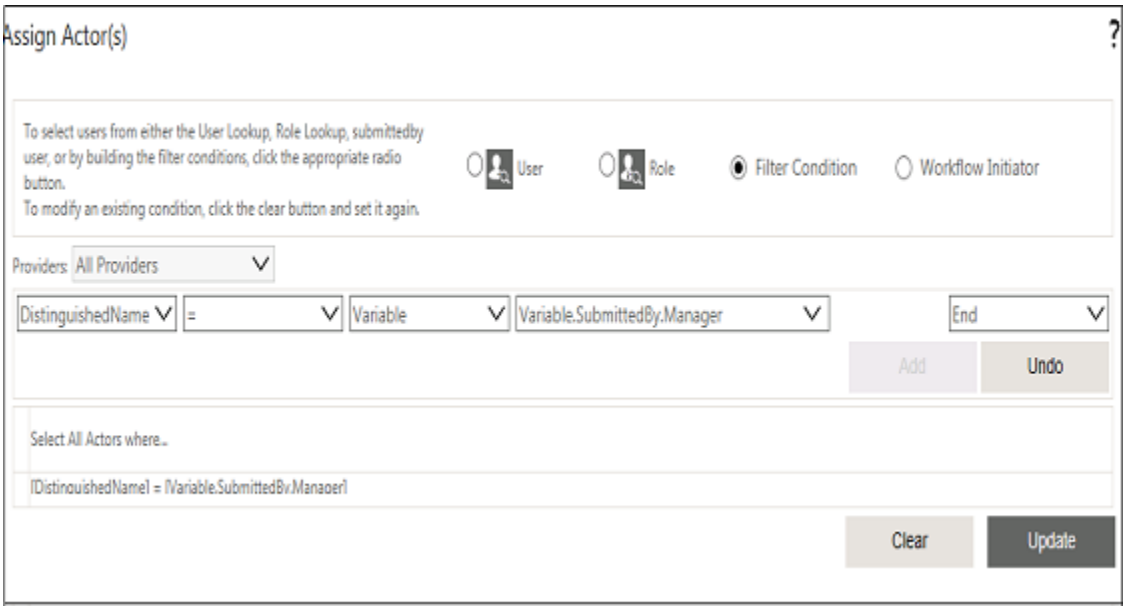


2. Right-click Invoke Form activity and then select **Activity Properties**.
3. Set the properties for **Actors** group as shown in the figure below.
   Here only required properties are explained.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



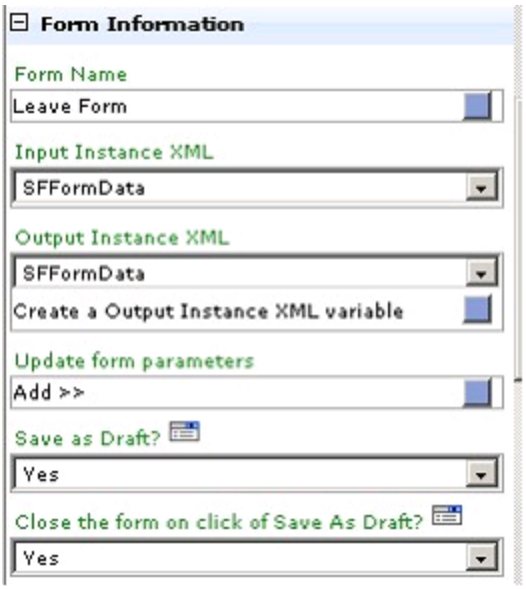
- **Assign Actor(s):** Set this property as shown in the following figure.



4. Set the properties for Form Information group as shown in the following figure.

Note: Here only required properties are explained.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

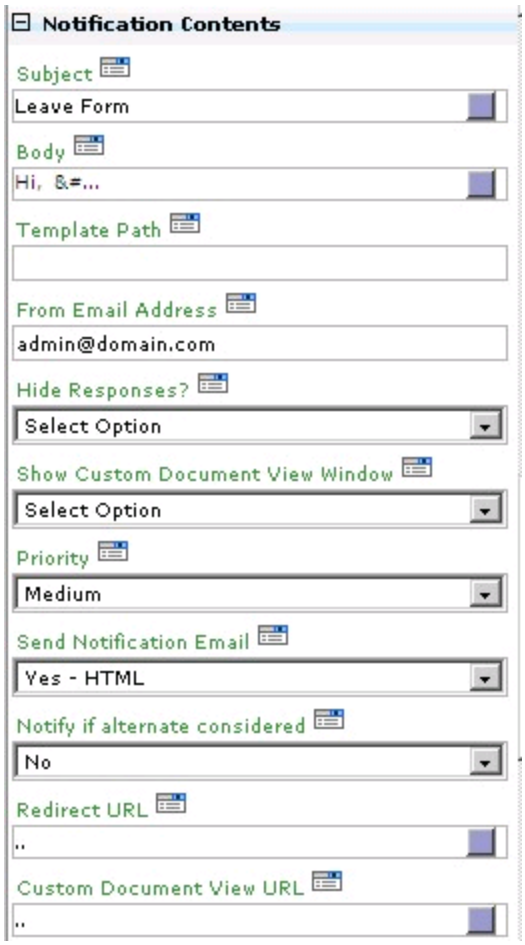


- **Form Name** : Select 'Leave Form' and its version from **Form Name** property window.
- **Input Instance XML**: Select 'SFFormData' from **Input Instance XML** drop-down list.
- **Output Instance XML**: Select 'SFFormData' from **Output Instance XML** drop-down list.
- **Save as Draft?:** Select **Yes** from **Save as Draft?** drop-down list. By setting this property as Yes, the Save As Draft option at the bottom right corner in the Form will be seen. This is useful for filling the form and saving it as draft till final submission of the form.
- **Close the form on click of Save As Draft?:** Select **Yes** from **Close the form on click of Save As Draft?** drop-down list. By setting this property as **Yes**, the Form will close on clicking the **Save As Draft** option.

5. Set the properties for **Notification Contents** group as shown in the following figure.

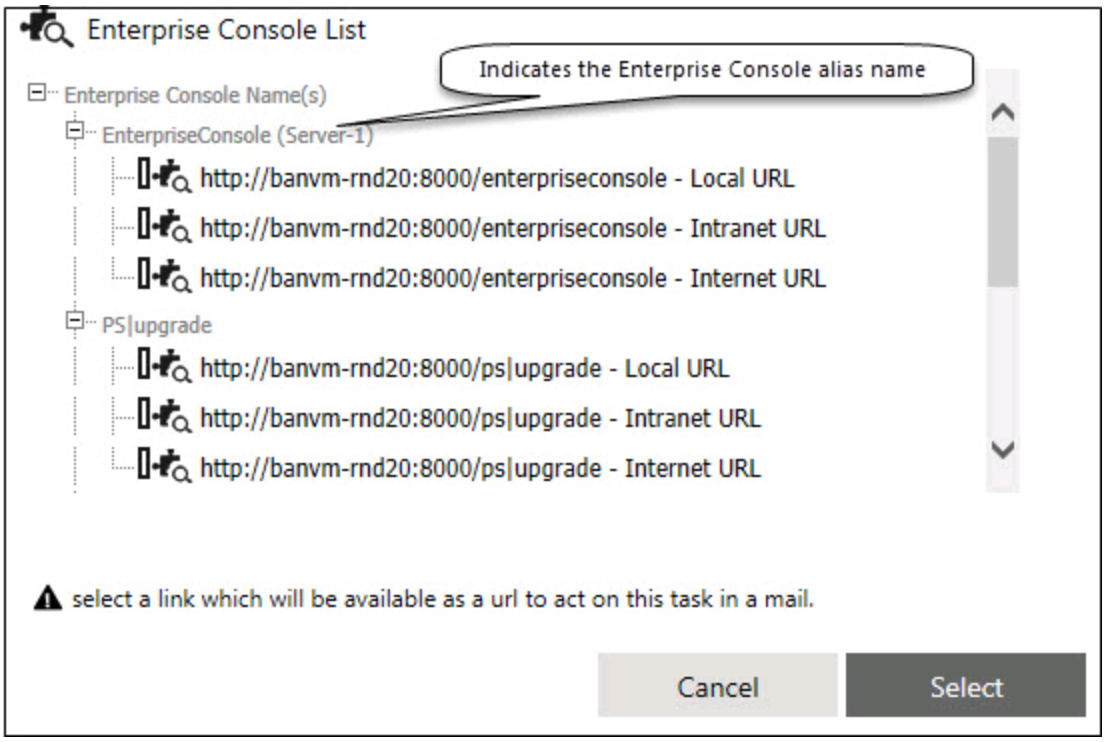
Note: Here only required properties are explained.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

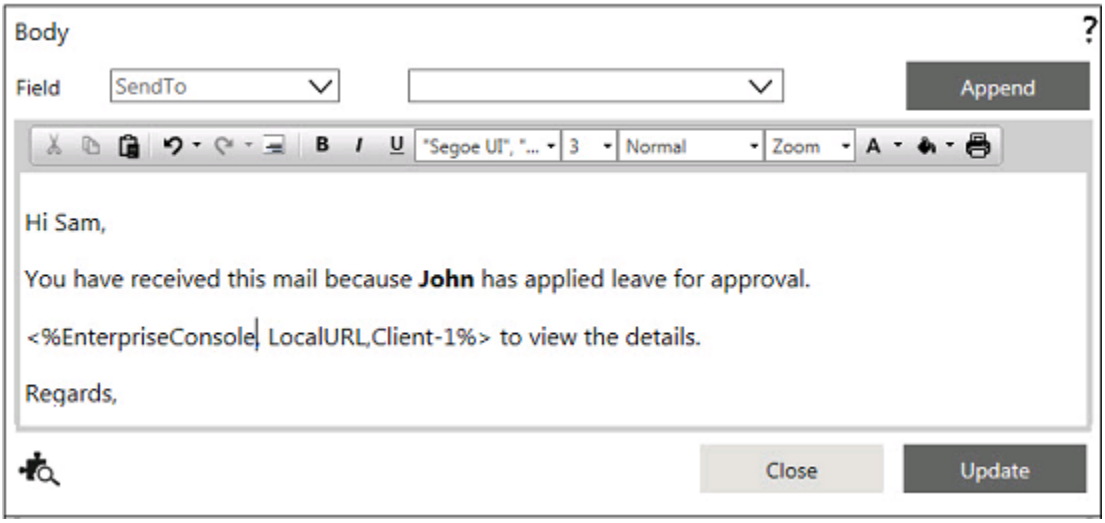


- **Subject** : Specify the appropriate subject from **Subject** property window.
- **Body**: Click the edit button in the **Body** property to open the Body property window.
  a. Compose the body text.
  b. Click **Enterprise Console Explorer**(plug-in) icon to select the URL. The following window is opened with list of Enterprise Console names. Select the Local URL and then click Select. The selected URL will be appended in the Body window wherever the cursor is pointed as <%EnterpriseConsole,LocalURL%>.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The composed mail will be as shown in the figure below.



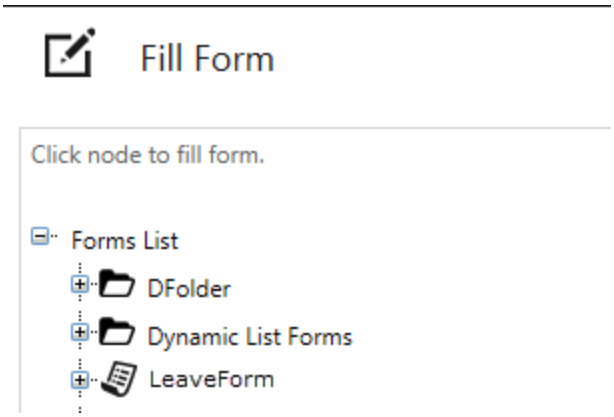
Click Update of the Body window property to save the contents.

- **Send Notification Email**: Select **Yes - HTML** option from the drop-down list.

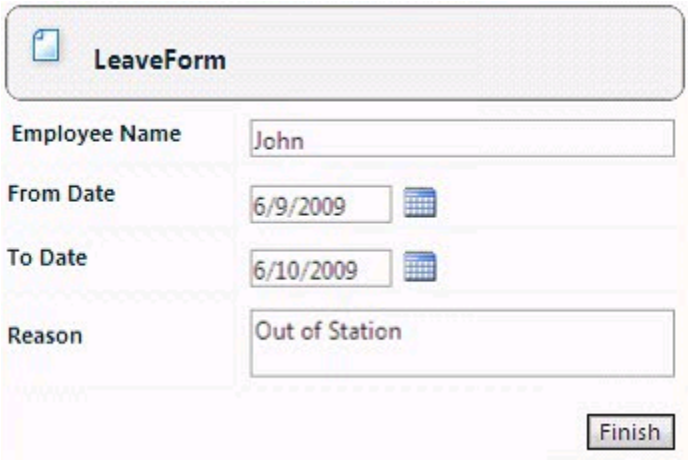
  Note: You will be able to send an email only if, this property is set as Yes - HTML or Yes. If it is Yes, then the mail will be sent as a plain text. If it is Yes- HTML, then the mail will be sent as HTML format.

6. After setting these properties, click **Save** of the **Activity Properties** window.
7. Publish the 'LeaveWF' workflow.
8. Go to **Enterprise Console > Fill Form**. The **Fill Form** window is displayed as shown in the following figure.

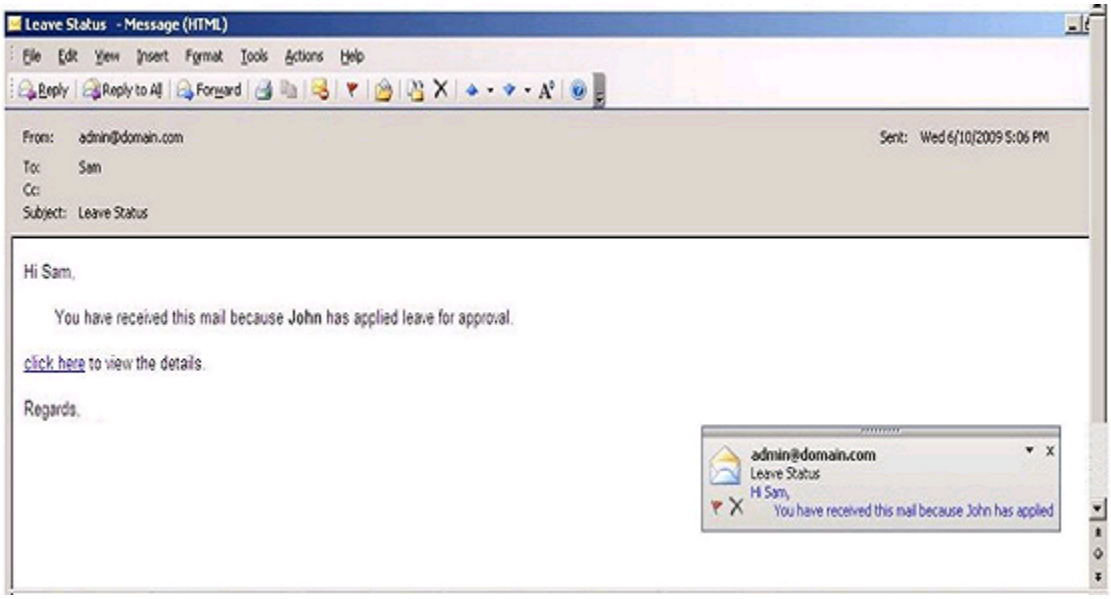
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



9. Click on the 'LeaveForm'. The 'LeaveForm' form is displayed. Fill this form and then click **Finish**.



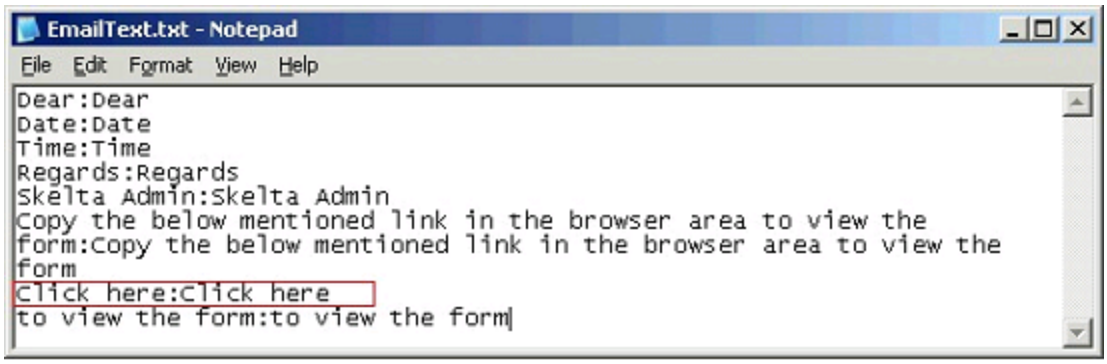
10. On filling the form, the mail will be sent to the manager for approval.



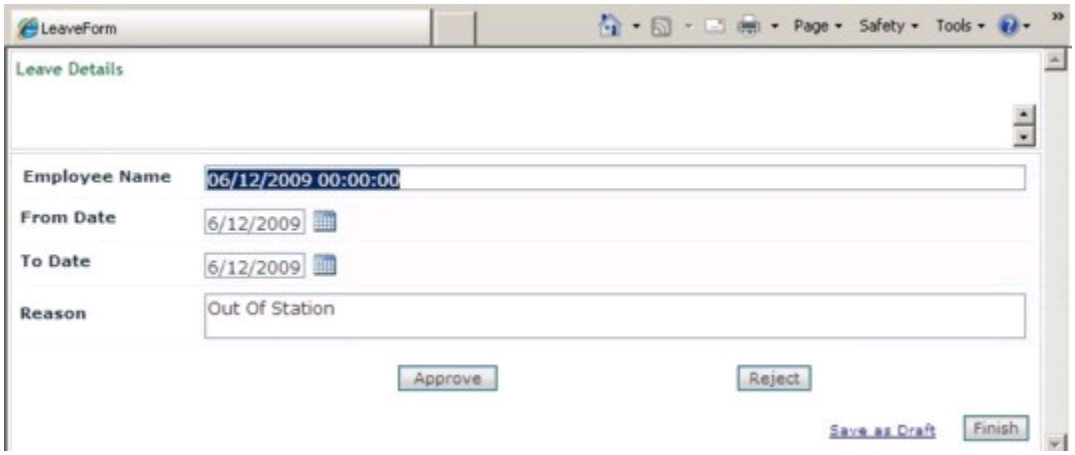
<%EnterpriseConsole,LocalURL%> text added in the Body tag is replaced with Click here hyperlink text.

Click here text can be changed from the [AVEVA Work Tasks Installed Path]\Locales\en-US\EmailText.txt to the required text.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



11. In the mail, just click on **'Click here'** link to open the form in a browser as shown in the following figure.



Conclusion: Here manager will get the direct link to the leave form which is applied by the employee and here manager can approve or reject leave application instead of logging in to the Enterprise Console.

Note: You can act on a task only first time. Next time, it will display the following message:

"Duplicate Response

Your response for the task <Activity Name> of <Date> application's <Workflow Name> workflow was recorded earlier! Workflow engine did not consider this duplicate response. "

Use 'Template Path' Property

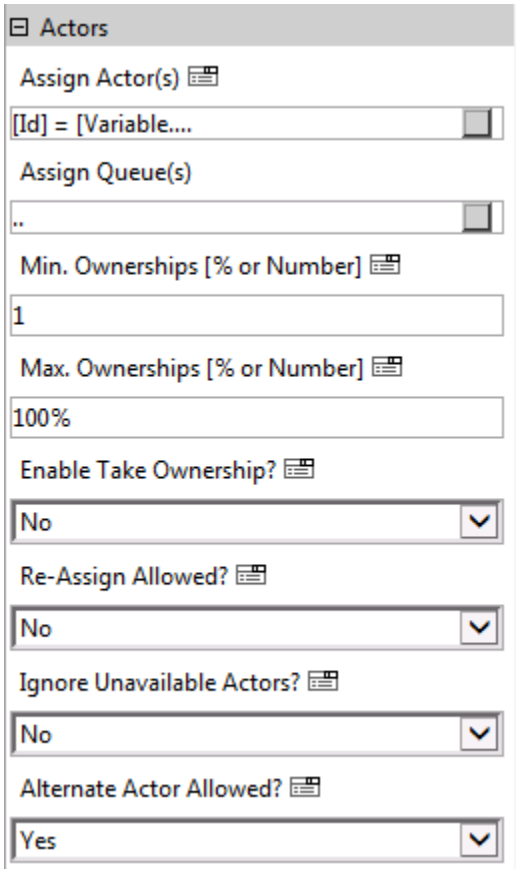
Scenario Solution:

1. Design a workflow named 'LeaveWF' as shown in the figure below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Right-click the Invoke Form activity and then select Activity Properties.
3. Set the properties for Actors group as shown in the following figure.

   Note: Only the required properties are explained here.

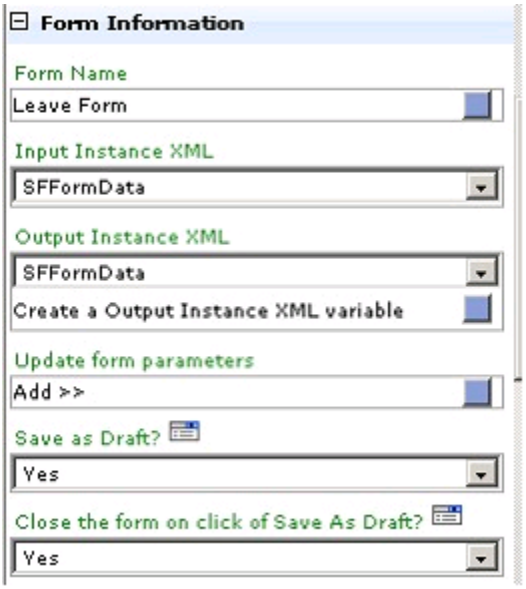


4. Assign Actor(s): Set this property to send notification mail to the manager as shown in the following figure.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Set the properties for **Form Information** group as shown in the following figure.

Note: Only the required properties are explained here.



a. **Form Name**: Select 'Leave Form' and its version from **Form Name** property window.

b. **Input Instance XML**: Select 'SFFormData' from **Input Instance XML** drop-down list.

c. **Output Instance XML**: Select 'SFFormData' from **Output Instance XML** drop-down list.

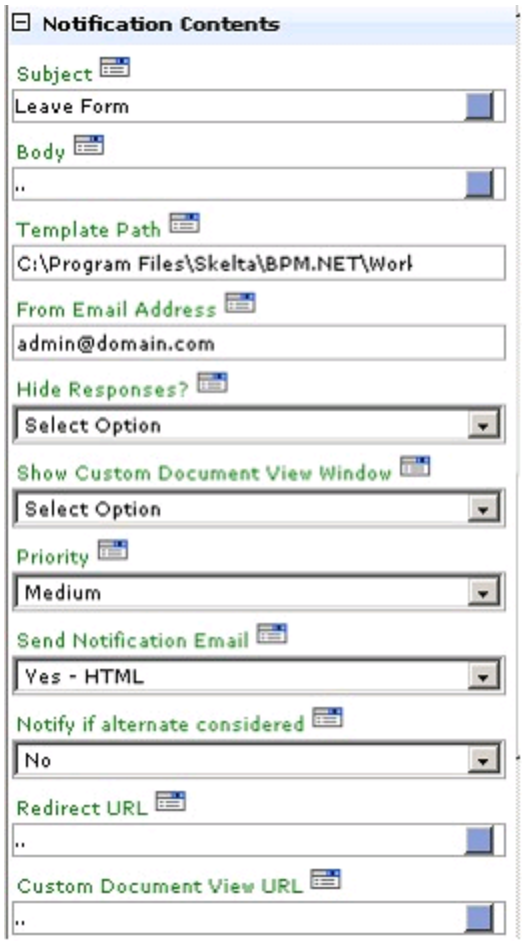
d. **Save as Draft?:** Select Yes from Save as Draft? drop-down list. By setting this property as **Yes**, the **Save As Draft** option at the bottom right corner in the Form will be provided. This is useful for filling the form and saving it as draft till final submission of the form.

e. **Close the form on click of Save As Draft?:** Select **Yes** from **Close the form on click of Save As Draft?** drop-down list. By setting this property as **Yes**, the form will close on clicking the Save As Draft option.

f. Set the properties for **Notification Contents** group as shown in the following figure.

Note: Only required properties are explained here.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



a. **Subject**: Select 'Leave Form' from Subject property window.

b. **Body**: Specified content will be used in the <%#Body%> placeholder in the template file.

c. **Template Path**: Select the 'InvokeFormEmail.html' from the path *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks*\WorkflowElements\Default\en-US\Templates\FormEmailTemplate\InvokeFormEmail.html and type the same path in the Template Path property.
Example1: *C:**Program files**AVEVA**Work Tasks*\\WorkflowElements\Default\en-US\Templates\FormEmailTemplate\InvokeFormEmail.html
Example2:<%#AVEVA Work Tasks InstallPath%>\WorkflowElements\Default\en-US\Templates\FormEmailTemplate\InvokeFormEmail.html

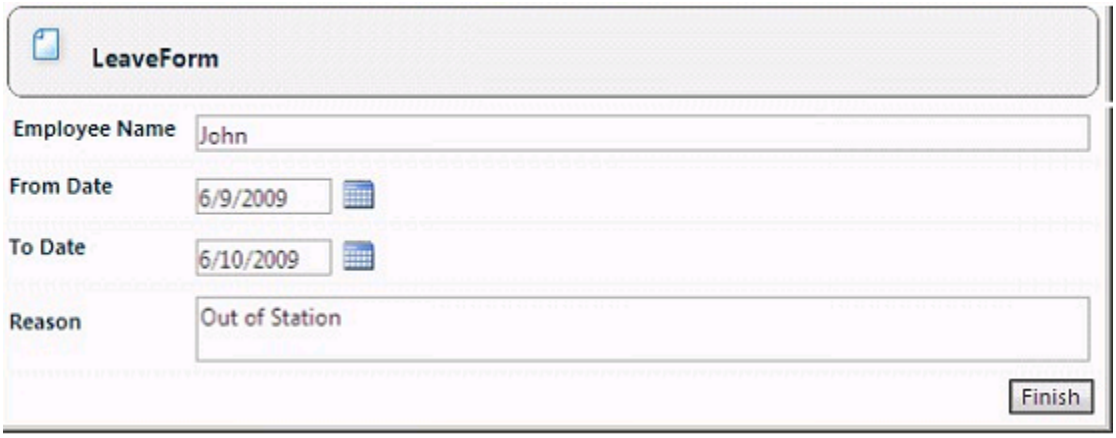
d. **Send Notification Email**: Select **Yes - HTML** option from the drop-down list.

Note: The user will be able to send an email only if this property is set as Yes - HTML.

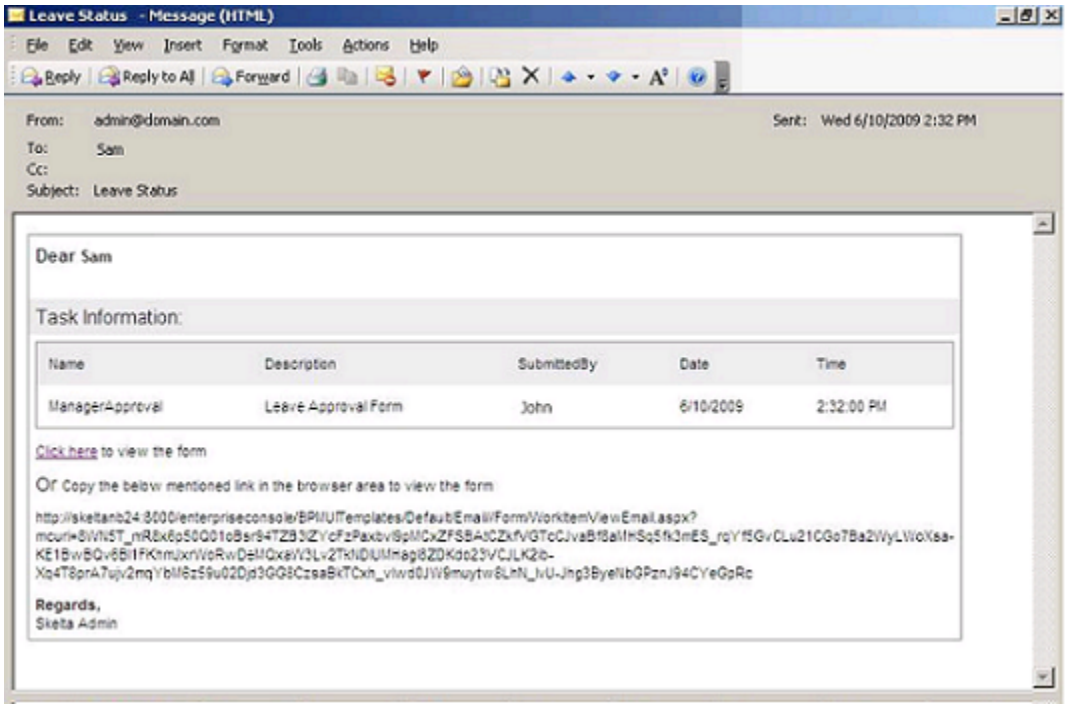
6. After setting these properties, click **Save** of the **Activity Properties** window.
7. Publish the 'LeaveWF' workflow.
8. Go to **Enterprise Console -> Fill Form**. The **Fill Form** window is displayed as shown in the following figure.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Click the 'LeaveForm'. The 'LeaveForm' form is displayed. Fill this form and click **Finish**.

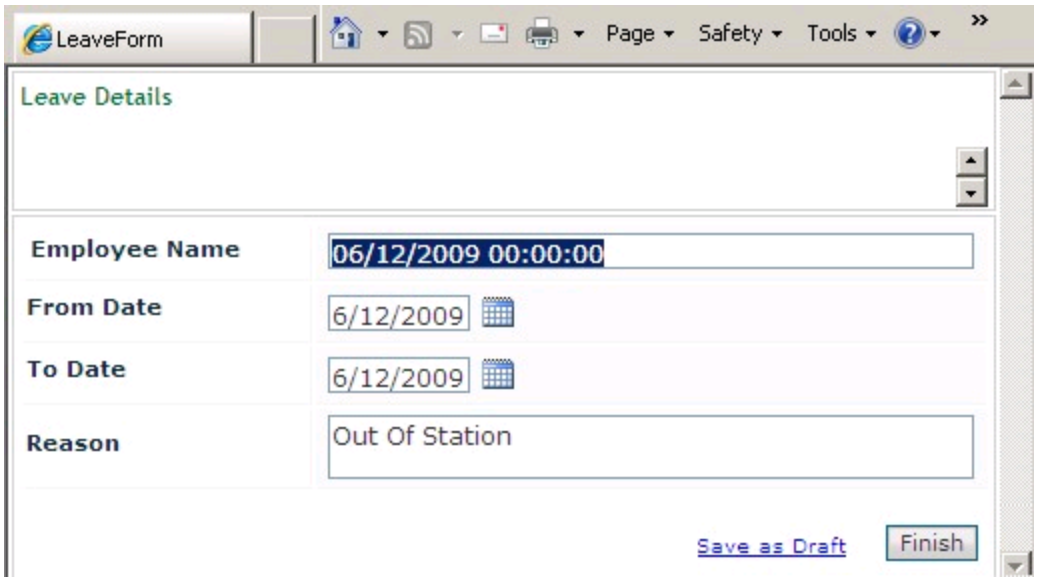


10. On filling the form, the mail will be sent to the manager for approval.



11. In the mail, just click **'Click here'** link to open the form in a browser as shown in the following figure.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Conclusion: Here, the manager will get the direct link to the leave form which is applied by the employee and here manager can approve or reject leave application instead of logging in to the Enterprise Console.

Note: The user can act on task only one time. Next time, it will display the following message:

"Duplicate Response

Your response for the task

Assign a Task to a Manager

'DistinguishedName' Property

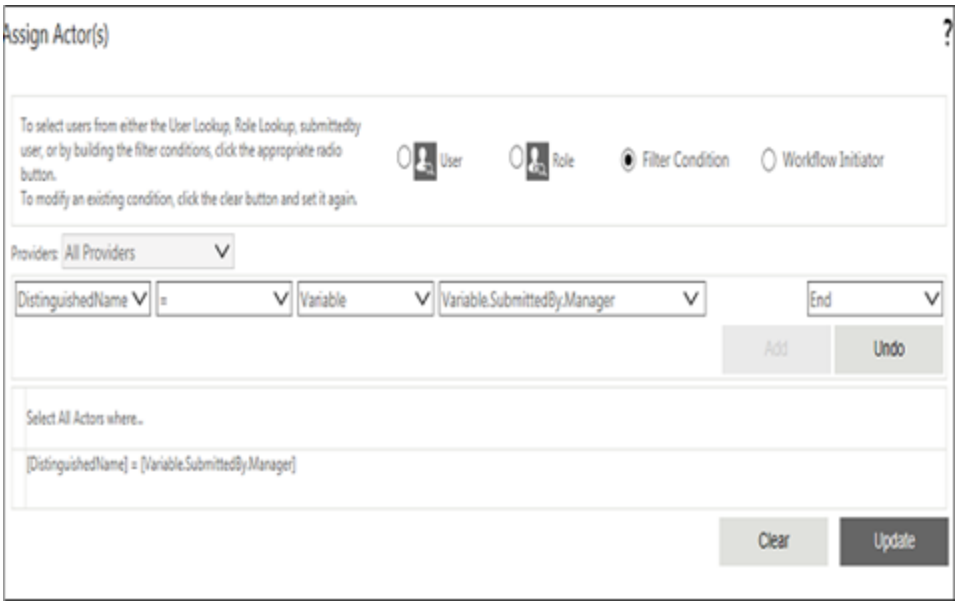
AVEVA Work Tasks returns the 'DistinguishedName' property of a user which uniquely identifies an entry in active directory for the Manager property. The property 'DistinguishedName' is added for active directory providers to query an entity or user with his DistinguishedName.

For example, the DistinguishedName value for a User Bob will be "CN=Bob,OU=HR Team,DC=Domain1,DC=dom (Note:DC will change according to the domain information)

Usage of this property: To assign a task to a manager, use this property in Assign Actor(s) property for all Human Activities as follows:

DistinguishedName = Variable.SubmittedBy.Manager'

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Enterprise Console Control (EC Plug-in)

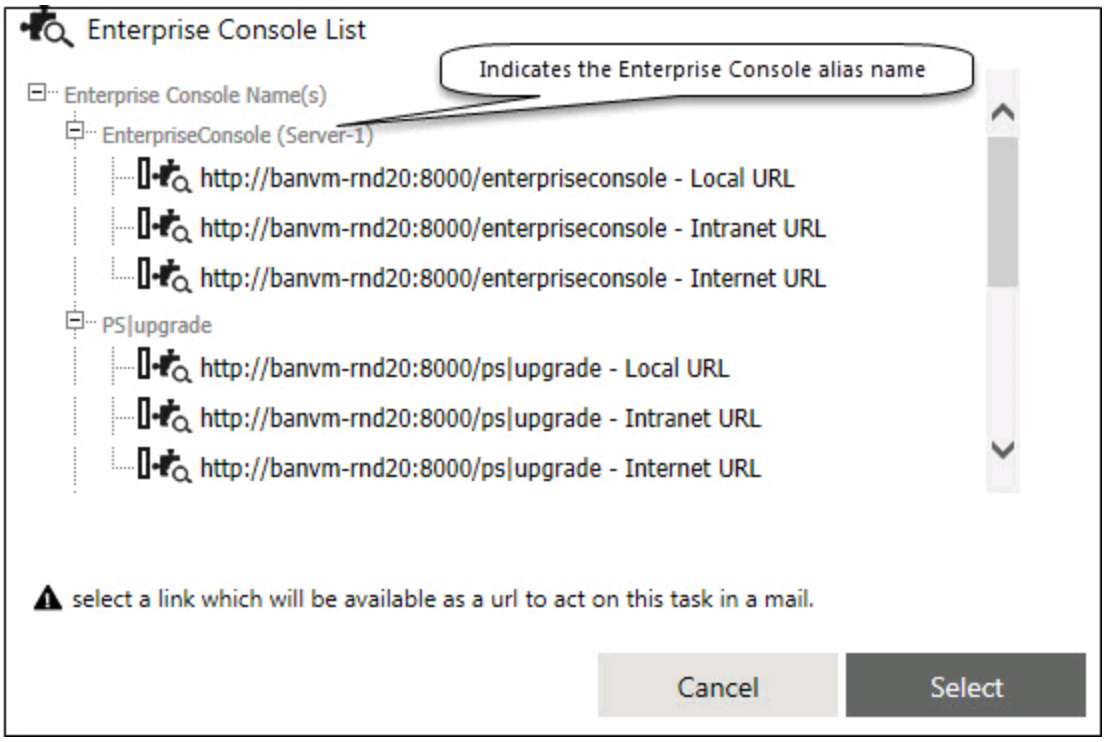
The **Body** property of all human activities has an EC plug-in. Click the **Enterprise Console Explorer** (plug-in) icon to get the required URL.

Note: This is applicable only if, **Sent Notification Email** property is set to **Yes-HTML** or **Yes**.

To compose the body text, perform the following steps:

1. **Body**: Click the **Edit** button in Body property, to open the **Body Property** window.
   a. Compose the body text.
   b. To get a direct Work Item details view link, click the **Enterprise Console Explorer** (plug-in) icon to select the required URL and then click **Select** as shown in the following figure.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



In a client-server scenario, the Enterprise Console site name is same for the client and server web site. To differentiate the Enterprise console site name of the client and the server, the **SiteAliasName** column is used in the **SKWebSite** table.

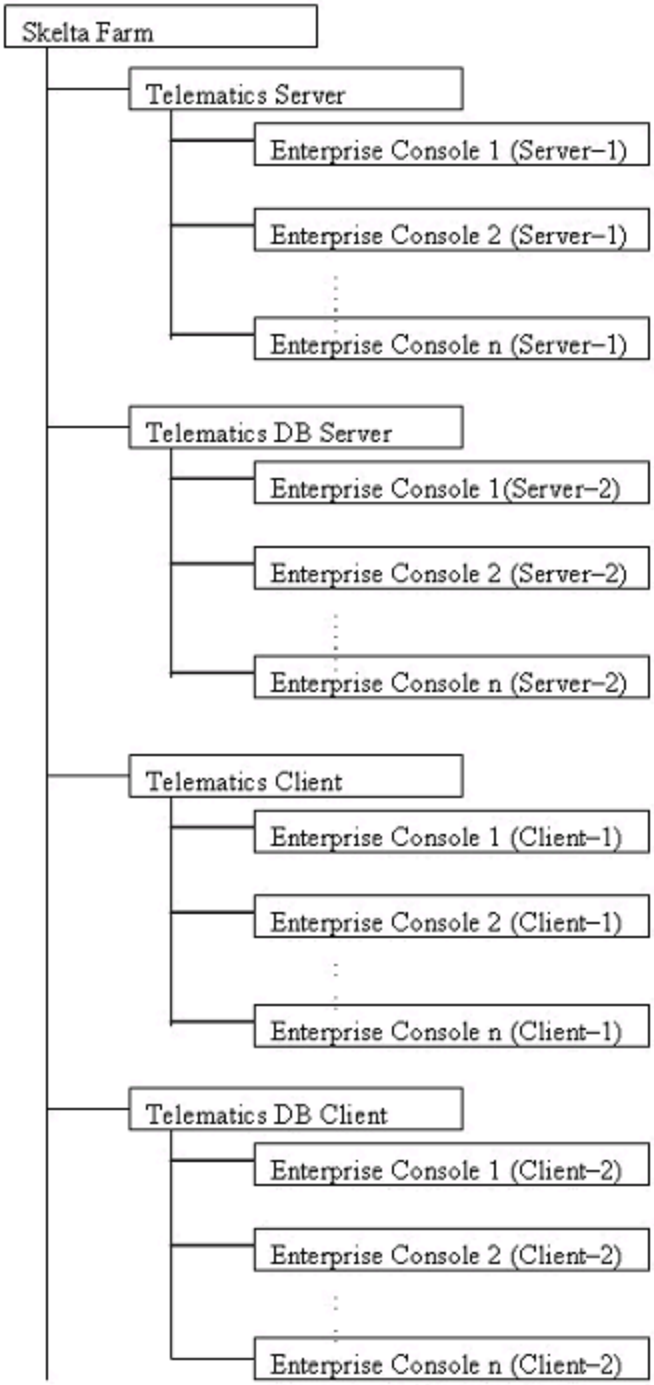
Example: Assume that there are four machines: **Telematics Server**, **Telematics DB Server**, **Telematics Client** and **Telematics DB Client**.

Telematics Server is connected to the Farm and 'n' number of Enterprise Console sites are created on the Telematics Server machine. By default the machine will be named as 'Server-1'. Telematics DB Server is connected to the Farm and 'n' number of Enterprise Console sites are created on the Telematics DB Server machine. By default the machine will be named as 'Server-2' and so on.

Telematics Client is connected to the Farm and 'n' number of Enterprise Console sites are created on the Telematics Client machine. By default the machine will be named as 'Client-1'. Telematics DB Client is connected to the Farm and 'n' number of Enterprise Console sites are created on the Telematics DB Client machine. By default the machine will be named as 'Client-2' and so on.

The pictorial diagram of the above explanation is shown below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

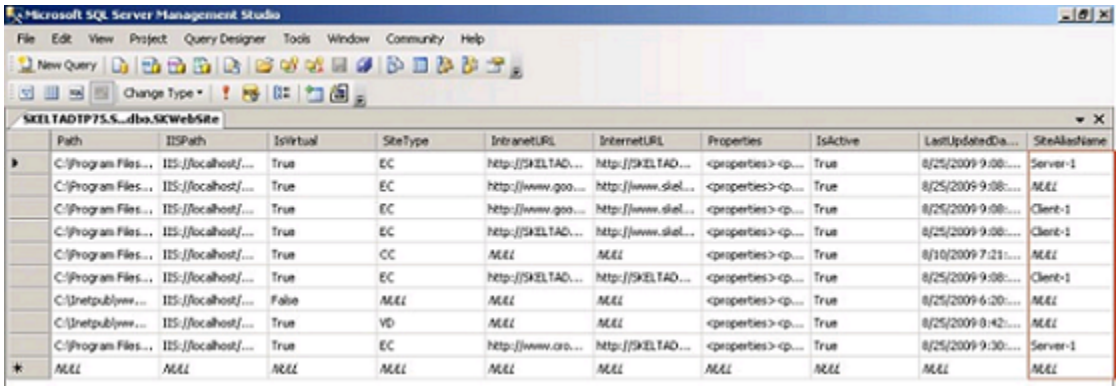


How to change Enterprise Console Alias Name?

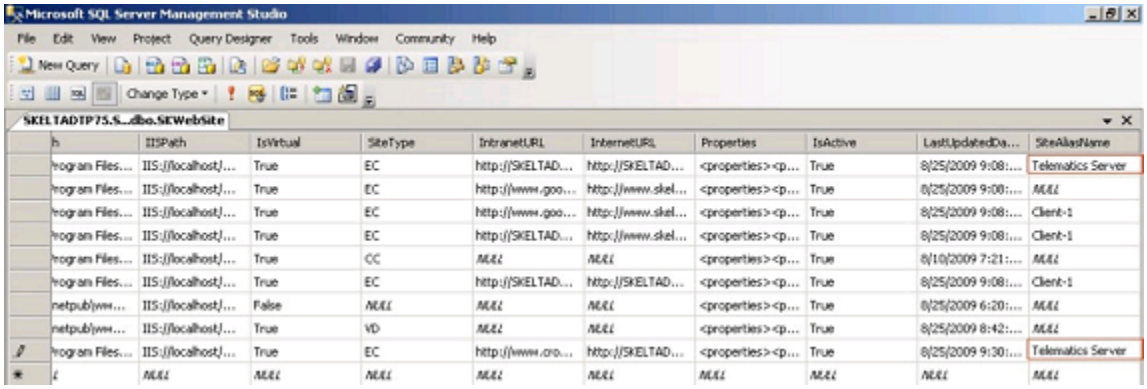
1. Go to the Farm Database.
2. Open the **SKWebSite** table.
3. Change the required Enterprise Console site alias name. Make sure that the name is unique across the machines. Example: If you want to change Enterprise Console alias name from Server-1 to Telematics Server, then you have to change the entry across the table in the SiteAliasName column to Telematics Server wherever Server-1 entry exists.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Before Changing



After changing the entry

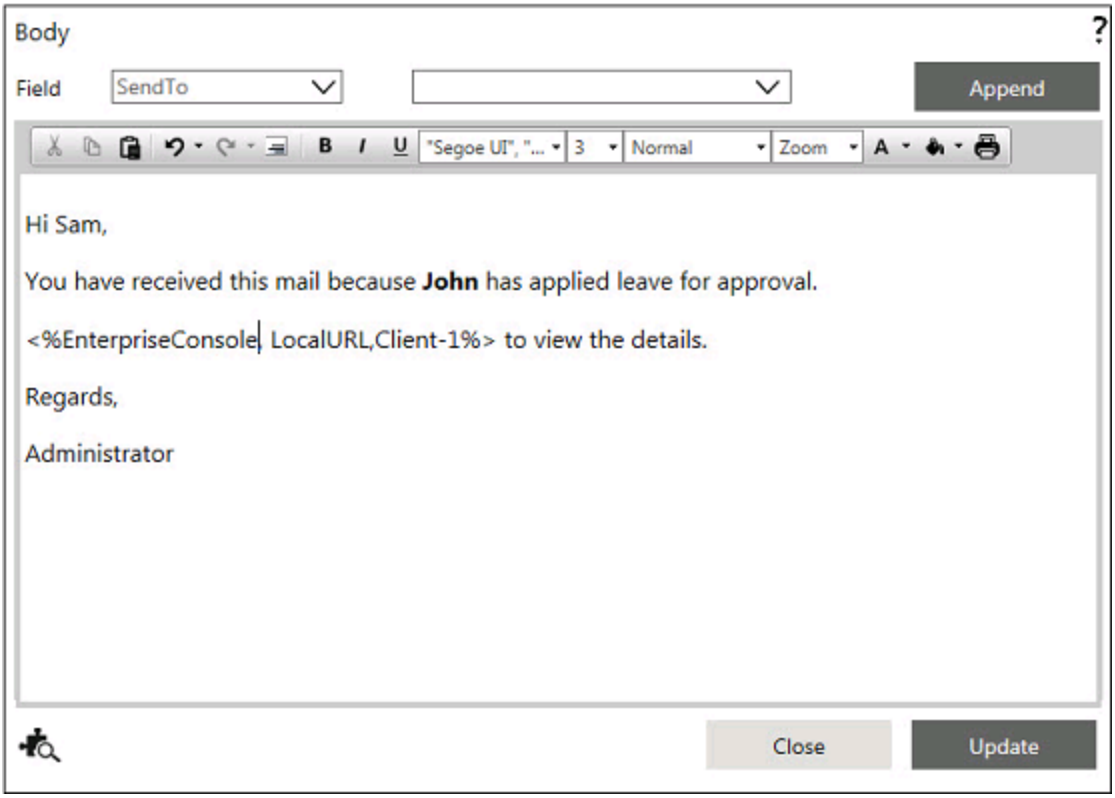


After renaming the web site name, once again you need to reconfigure the Body property wherever this placeholder has been used in a workflows.

Note: If the workflow is imported from the other machine then make sure that the Enterprise Console name should be same while selecting URL in the EC plug-in window. If it is not the same then follow the steps instructed above to change the Enterprise Console site alias name.

4. The composed mail as shown in the following figure.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



5. Click **Update** of the Body window property to save the contents.

Integration Activities

The following activities are found under Integration Activities:

- **Create-Open-Office-Document Activity**: You can create Microsoft 2010 Word and Microsoft 2013 Word documents from the workflow using the Create-Office-Document activity.
- **InvokeWebService Activity**: The Invoke-Web-Service Activity can be used to invoke a web service from the workflow.
- **Invoke Web API Activity**: You can use the Invoke Web API Activity to invoke a web API from the workflow.
- **Script Activity**: The Script Activity enables the execution of script written in languages, such as JavaScript and VBScript, from the workflow.
- **File Operations Activity**: The File Operation Activity can be used to perform file transfers from within the workflow.
- **Change Workflow Status Activity**: The Change Workflow Status Activity updates the status of the workflow in the database.
- **XML Action Activity**: The XML Action Activity can be used to assign XML data for activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create-Open-Office-Document Activity

Activity Description

You can create Microsoft 2010 Word and Microsoft 2013 Word documents from the workflow using the Create-Office-Document activity. When you include this activity in the workflow, the activity uses the selected template and creates a new document in the specified location with the relevant data.

Creating MS Office Templates

Steps to create an MS Word template

1. Open a new Word document, and save it with a relevant name.
2. Enter the required field names for each insert placeholders as shown below.
   a. Drop down the 'Insert' menu and click the 'Fields' option.
   b. In the 'insert fields' window, select 'MergeField' option from the 'Field names' list box.
   c. Enter a name for the new field and click 'OK'.

   Note: Create a merge field name without spaces in between.
3. The new fields are displayed along with the corresponding placeholders enclosed in double chevrons. The new names will be displayed as placeholders in the window corresponding to the 'SelectTemplate' property. These placeholders will be replaced by the actual values during execution of the workflow.

Workflow Variable for capturing Document Path

Create Office Document stores p hysical path information of the document in a variable of type string with the following syntax:

'^'+CurrentActivity.Name+'DocumentPath'

For example, **^CreateFileDocumentPath**.

Activity Properties

The Create-Open-Office-Document activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Create-Open-Office-Document activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Document Name - This property specifies the name of the new document.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Destination Details - This property specifies the details of the destination in which the document should be created such as the file path, the user name and password to access the document among others.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Destination Details property

1. To specify the details of the destination in which to create the new document, click on the button for the 'Destination Details' property to open the property window.
2. In this window, specify the protocol for the file transfer, for example, FTP-HTTP in the **Protocol** field.
3. Then specify the location; the pattern to use for the transfer, for example, .DOC; the User Name, Password, Domain Name, and Port Number in the respective fields.
4. After specifying all the details, click **Submit** to save the settings.

See Create-Open-Office-Document: Destination Details for a detailed description of the Destination Details property window.

Select Template - This property specifies the template to be used for creating the document.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Select Template property

1. To select the template for creating the document, click on the button for the 'Select Template' property to open the property window.
2. In this window, specify the user name, password, and domain name in the respective fields.
3. After providing the credentials click **Browse** and browse to the location of the template.
4. Select the template file and click **Import**.
5. To specify the values to the placeholders, select the placeholder and select the type of value (Value, Variable, XML Variable or Content) from the drop down. Then assign the value by selecting the appropriate option from the second drop down or in case you had selected the "Value" option in the first drop down enter the value in the field that is displayed here.
6. After choosing the values for the placeholders, click **Submit**.

See Create-Open-Office-Document: Select Template for a detailed description of the Select Template property window.

Activity outputs

The Create-Open-Office-Document Activity does not return any value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Exception Behavior

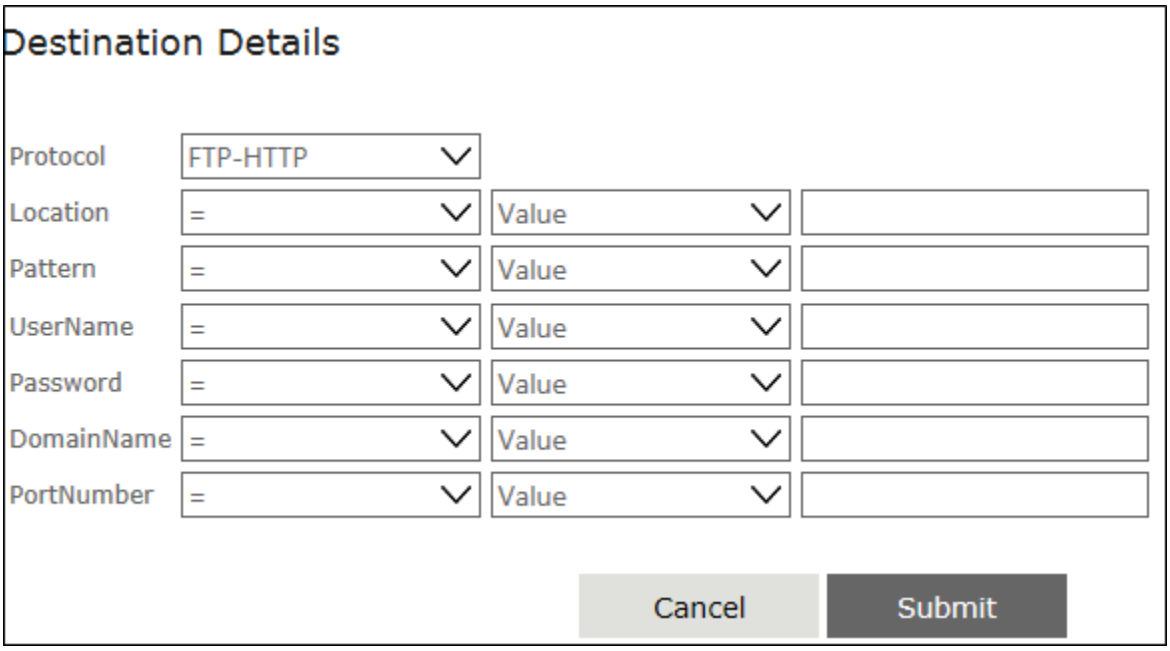
Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Create-Open-Office-Document: Destination Details

Destination Details Property window

You can specify the details of the file destination in this property window. To open this window, click the button for the 'Destination Details' property.



This window has the **Protocol** drop down where you can select the file transfer protocol. Below this drop down, it has controls to specify the following details.

- **Protocol:** If the selected protocol is FTP-HTTP, you can use FTP or HTTP protocol.

  For the HTTP protocol, you must enable the WebDAV Publishing feature for the web server. For more information about enabling the WebDAV Publishing feature, refer the Microsoft documentation.

  In the **Authentication** Feature in Internet Information Services (IIS) Manager, ensure that only **Windows Authentication** is enabled for the selected HTTP site.
- **Location**: You can specify the URI/UNC of the file or directory here.
- **Pattern**: You can specify the pattern to decide the file types to be transferred, such as *.doc,*.xls, in this field.
- **UserName**: You can specify the user name for the destination here.
- **Password**: You can specify the password for the destination here.
- **DomainName**: You can specify the domain or computer name of the destination here.
- **PortNumber**: You can specify the port number of the destination server.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

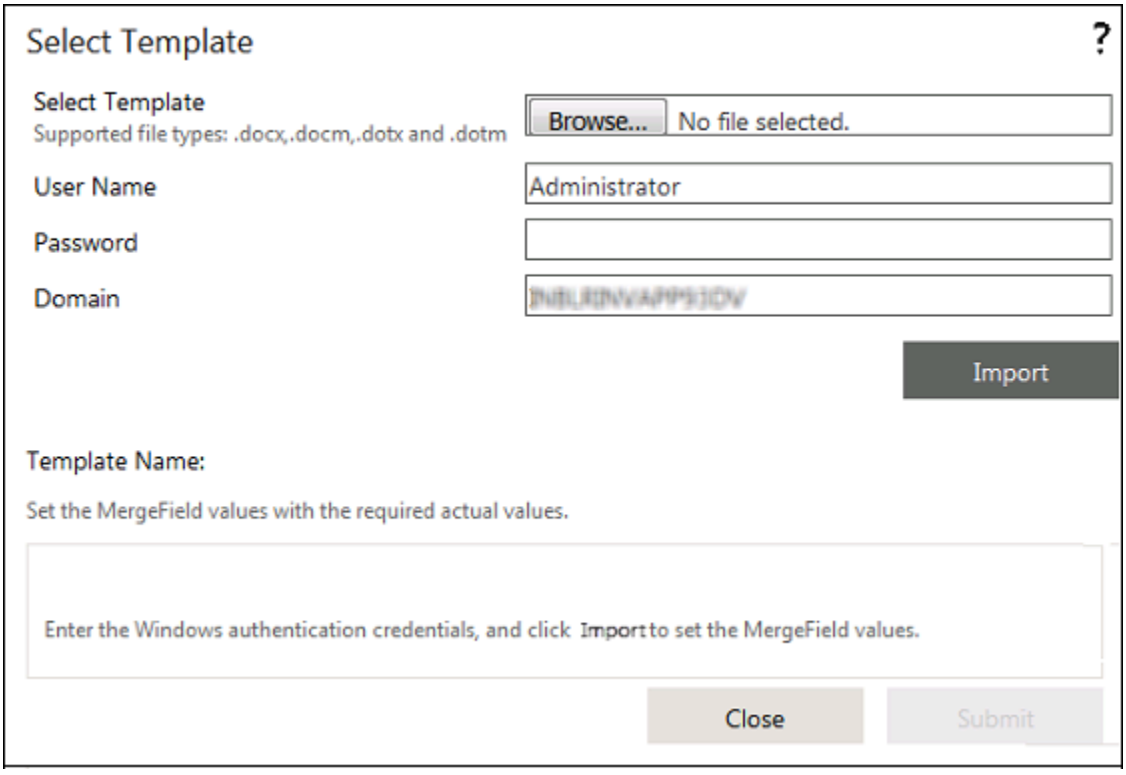
Each of the above parameter fields has the following controls.

- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the parameter in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the parameter. The control displayed depends on the option selected in the preceding drop down. If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.

Create-Open-Office-Document: Select Template

Select Template Property window

You can select the templates in this property window. To open this window, click the edit button for the **Select Template** property.

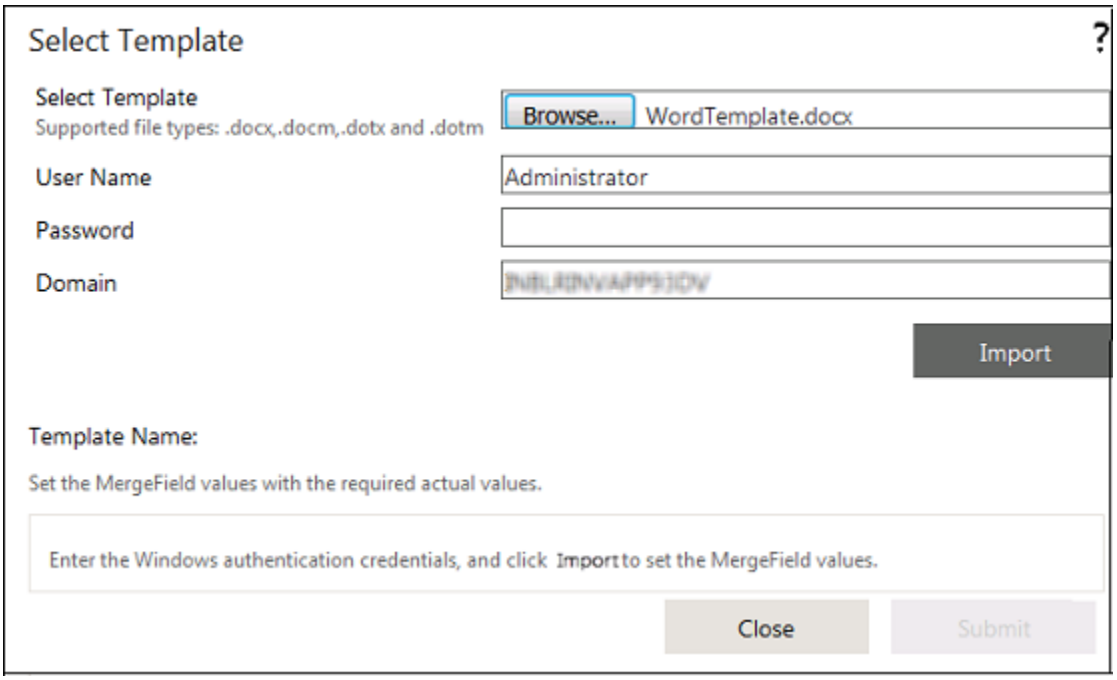


The top section of the window has the following fields to specify the login credentials and select the template.

- **Select Template:** You can enter the full path to the template here. Alternatively, click **Browse** to browse to the location of the template and select it.
- **User Name:** You can specify the user name here.
- **Password:** You can specify the password here.
- **Domain:** You can specify the domain name here.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Click **Import** to import the selected template. The imported template appears in the lower section of the dialog box.



The **Template Name** displays the name of the selected template. The place holders in the template are displayed in a grid with the following controls:

- The Placeholder name
- The Value type drop down. You can select the type of value to be assigned to the placeholder from this drop down. You can select Value, Variable, XML Variables, or Content here.
- The Value text box or drop down list. You can specify the actual value to be assigned to the placeholder here. If you select Value from the previous drop down list, a text box is displayed here, and you can enter the value directly. Otherwise, a drop down list is displayed, where you can select from the corresponding collection (Variable, XML Variable, or Content).

InvokeWebService Activity

Activity Description

The Invoke-Web-Service Activity can be used to invoke a web service from the workflow. The output of the activity can be used to determine the path of workflow execution.

Variables for capturing exception messages

The exceptions occurring in the Invoke Web Service activity can be captured by creating a system variable of type 'string'.

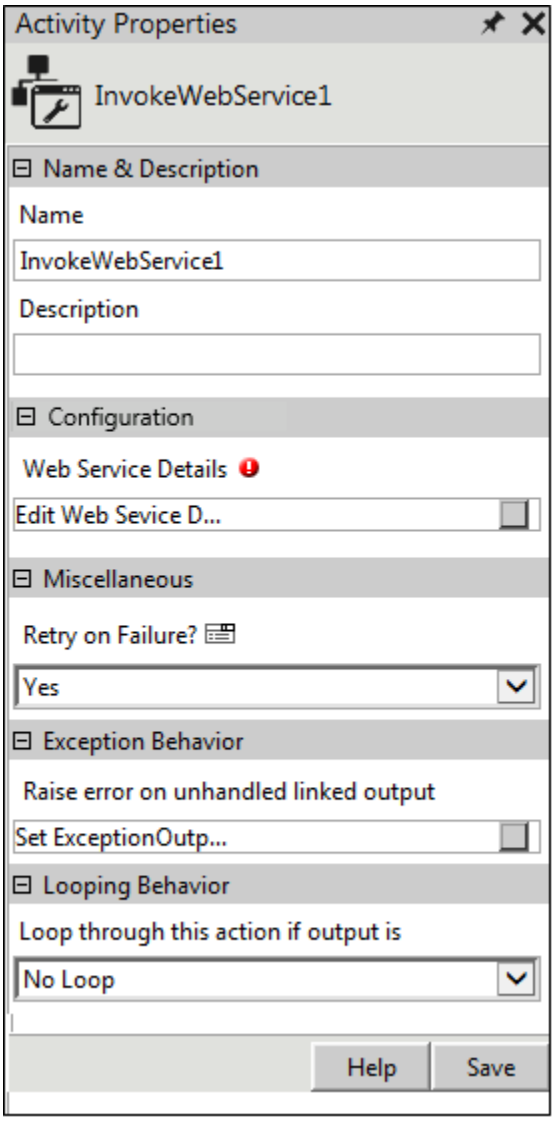
The variable name should be created in the following format:

'^'+CurrentActivity.Name+'Error'

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

The Invoke-Web-Service activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Invoke-Web-Service activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.



Name & Description

You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration

The property in the Configuration group has been described below. You can use this property to specify the web service details.

Web Service Details - The Web Service Details property allows the user to select a web service to invoke and specify the input and output details for it.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Web Service Details property

1. To add and configure Web Services, click on the button for the 'Web Service Details' property to open the Web Service List, which is the main property window.
2. This is the first window of the Web Service configuration user-interface and lists all the Web Services that have already been configured.
3. You can click the **Preview Methods** check box to preview the methods for the selected Web Service in this window.
4. Click the **Add** button in the web service listing window to add a new Web Service. This opens the window to add new Web Services. Enter the Web Service name and the WSDL URL, e.g., localhost/SkeltaWebService/WorkfllowClient.asmx?wsdl, in the respective fields in this window. Click on the **Save Web Service** button after specifying the required values here if the new Web Service does not need authentication.
5. Click on the **Authenticated** check box to select the option, if authentication is required for the web service being added. This displays authentication fields. Enter the login name, password, and domain name in the corresponding fields.
6. To bind a Web Service to the database right-click on the Web Service listing in the Web Service List window and select the Bind option from the context menu. This will display the Methods List window. If there are no public fields in the Web Service you are binding, go directly to step 8.
7. If there are any public fields in the selected web service, selecting the **Bind** option in the previous step will open the Public Fields List window listing all the public fields in the Web Service. You can edit field values by selecting the **Edit** option in the context menu for the fields list. In the web service parameter editing window that opens, click on the Serialize check box to enable serialization. Specify the value in the Value field by selecting the value type (Value, Variable, Content, XMLVariables) in the first drop down and providing the value. After specifying the value, click on the **Update Parameter Value** button to save it. Then click on the **Update** button at the bottom of the Public Fields List page to update the parameter value. After updating, click on the **Link** button to open the Web Service List window and right-click on the Web Service listing and then select Bind from the context menu. This will open the Methods List window mentioned in step 6.
8. In the Methods List window select the required method parameter(s) and bind them by clicking **Bind Parameters**.
9. In the Method Parameter listing window select parameter(s) to be updated and select the **Edit** option from the context menu. This opens the web service parameter editing window similar to the one in step 7. In this window specify the parameter values in the same way as described in step 7.
10. If you wish to specify authentication details for the web service, enter the Login Name, Password and Domain Name in the authentication fields in this window. If specified, these details will need to be used while executing the web service.
11. Select a Variable or Content Handler or XMLVariable to save the parameter values in from the **Save Output** in drop down.
12. Click on **Update** button after entering all the parameters to bind the Web Service.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

See WebService - Web Service Details for a detailed description of the Web Service Details property windows.

Miscellaneous

The property in the Miscellaneous group has been described below.

Retry on Failure? - This property allows the user to enable the activity to retry up to three times if the first attempt to connect to the web service fails. By default, it is set to **Yes**. This is useful to improve the chances of connecting to Webservices. This property can be set to **No** during development to avoid automatic retries.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

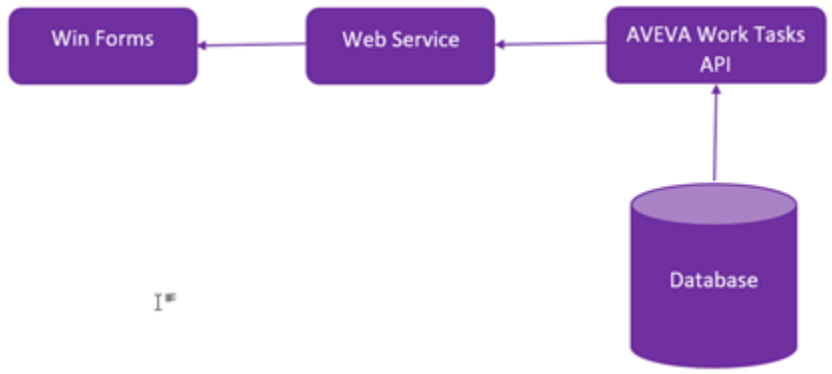
The Invoke-Web-Service Activity returns the following values -

Webservice Execution Failed: Displayed when the Web service execution has failed.

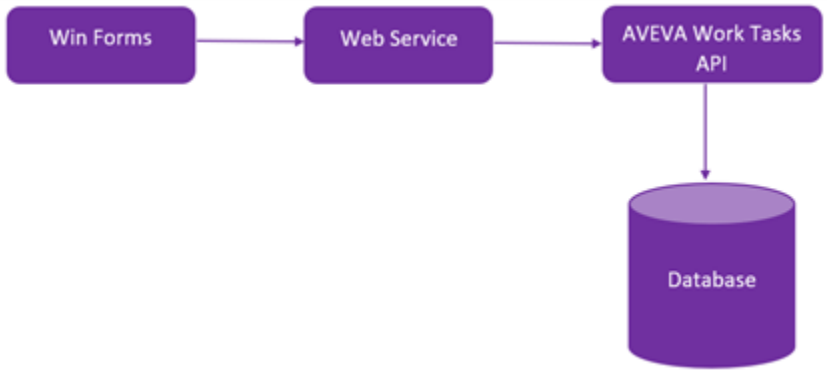
In addition to this standard return value it also returns the value specified by the user in the Return Values property as output.

The Web Service block diagram is as shown below:

Getting Work Item/Task details for an actor and rendering the same in Windows Forms



Submitting an Activity/Task via the web service and alerting the engine to execute next step in the workflow



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

Mapped Error Outputs for more details.

InvokeWebService: Web Service Details

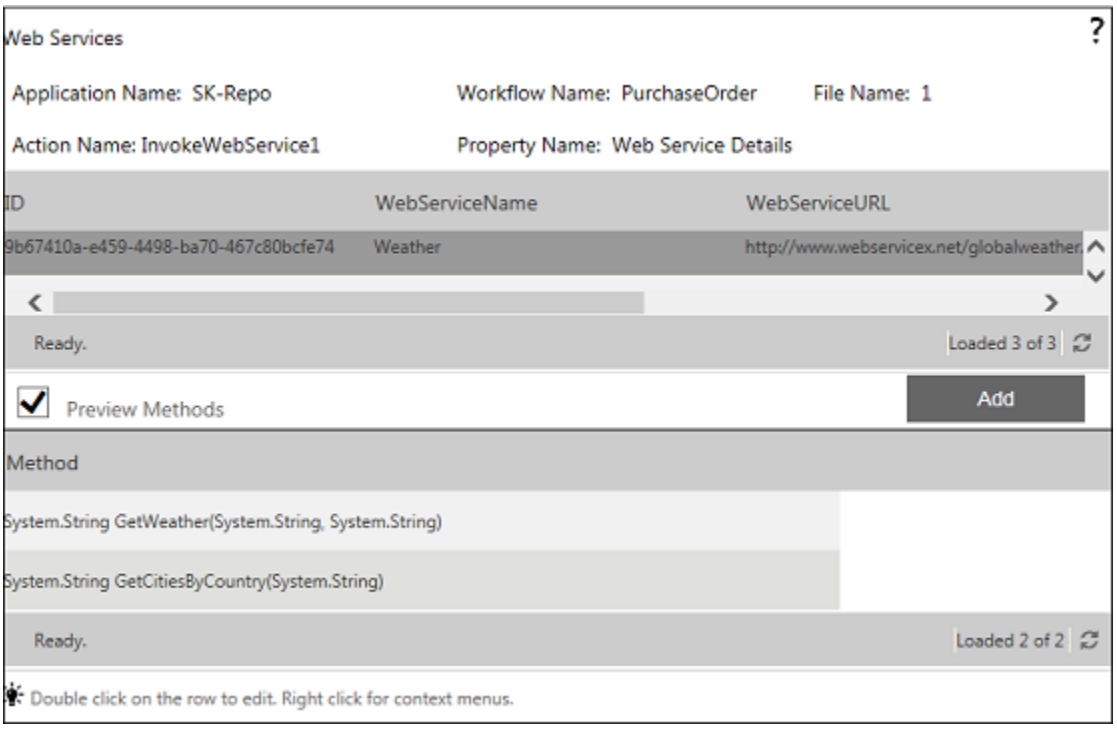
You can manage Web Services from the **Web Service Details** window.

The **Web Service Details** window lists all the web services configured for the application or workflow. The upper section of the window shows the web service details. The lower section of the window shows a preview of the methods of the selected web service if this provision is enabled.

The **Web Service Details** window has the following controls:

- **Preview Methods**: Enable the method preview display.
- **Add**: Add new web services.

The following image shows the Web Service Details window with the Preview Methods enabled:



To add a Web Service

1. Click **Add**.

   The window to add a web service appears.
2. Enter the following information:
   - **WebService Name**: Type a name for the Web Service.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **WebService URL**: Type the Web Service URL. You must enter a WSDL URL.

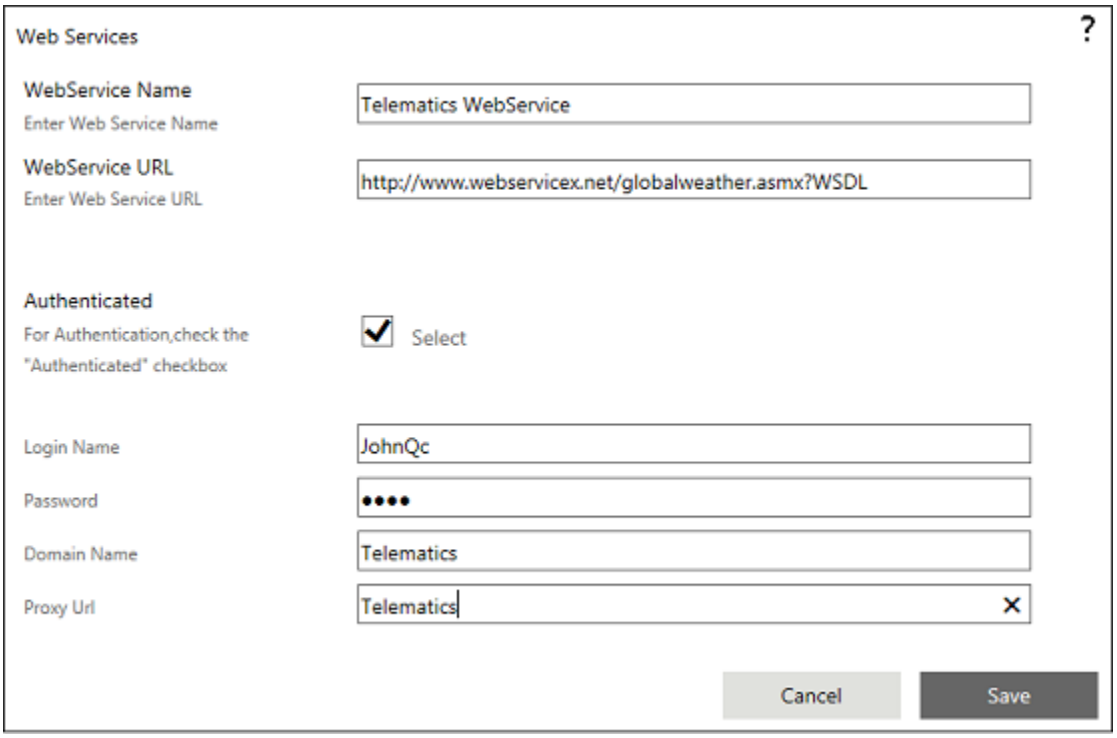
  For example: **localhost/SkeltaWebService/WorkfllowClient.asmx?wsdl**

  You can also enter an IPv6 address for the **WebService URL**. The IPv6 address must be enclosed within square brackets.
- **Authenticated**: Select if the web service requires authentication. When you select the **Authenticated** box, additional authentication fields appears.

  Enter the following information:
  - Login Name
  - Password
  - Domain Name
  - Proxy URL

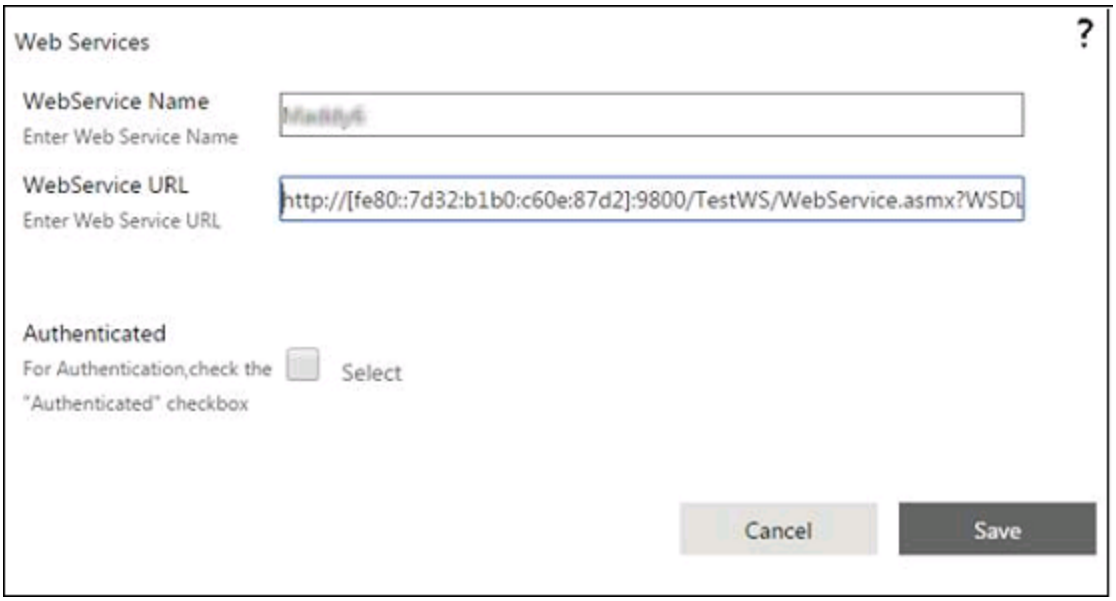
3. Click **Save**.

The following image shows a sample window for adding a web service:



The following image shows a sample window for adding a web service when the WebService URL is an IPv6 address:

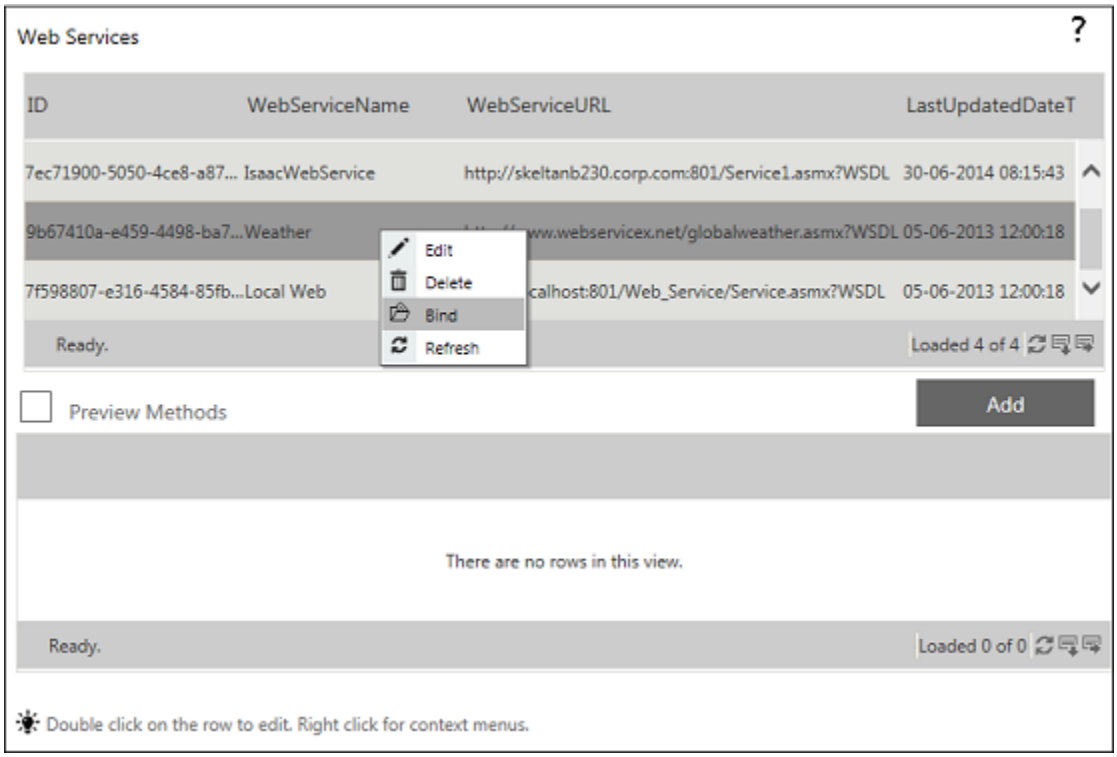
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Binding a Web Service

1. Right-click a web service.

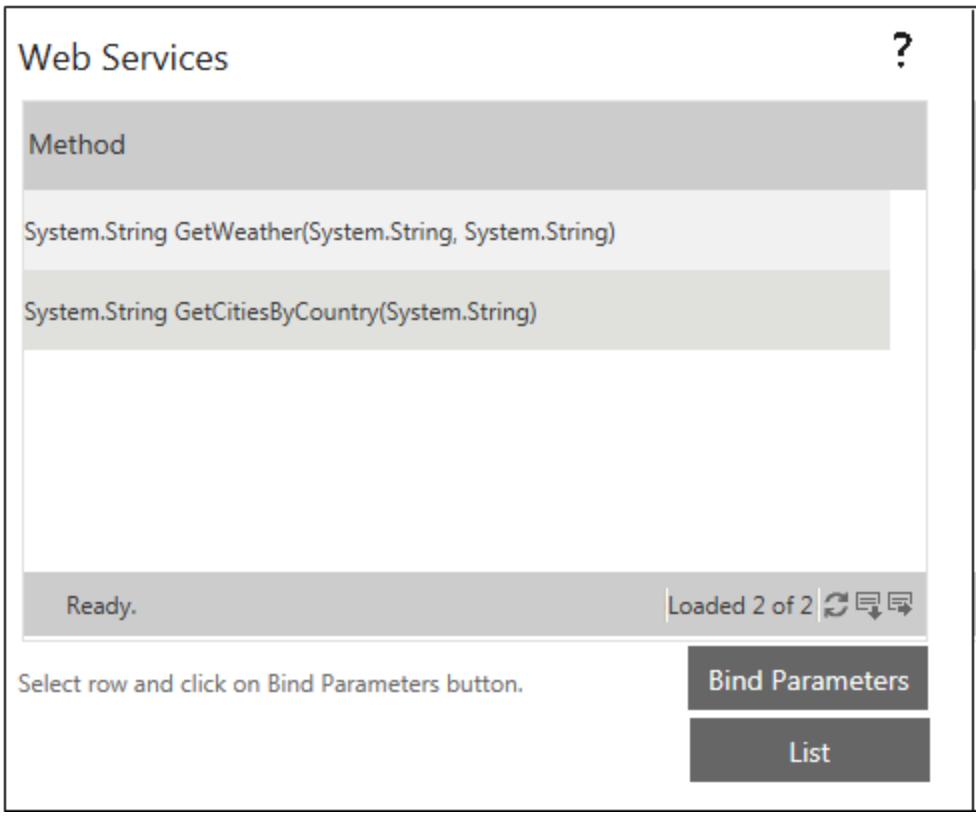
   The context menu shows **Edit**, **Delete**, **Bind**, and **Refresh** options.



2. Click **Bind** to bind the selected web service to the database.

   The **Methods** list of the web service appears in a new dialog box. This window also displays the Public Fields List window in case there are public fields in the selected web service.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Select the required method, and then click **Bind Parameters**.

   A dialog box appears that shows the details of the parameters of the selected method.

   You can click on the List button to return to the Web Services List page.
4. Select a Variable or an XML Variable to save the Web Service output in the **Save Output** in drop down.
5. If you select the **Save Output** in option as **XML Variable**, you can either use the available XML Variable or create your own XML Variable.
6. Click **Create XML Variable** to create an XML variable.

   The **Create XML Variable** dialog box appears.
7. Type the **XML Variable Name**, and then click **Create**.
8. If authentication is required, enter the Login Name, Password, and Domain Name.
9. Click Update.

Note: If the web method returns multi-level complex object, then create the XML variable in Start Activity using a sample XML schema returned by the web method. Then select this variable in the Invoke Web Service activity.

Invoke Web API Activity

Activity Description

You can use the Invoke Web API Activity to invoke a web API from the workflow. Use the output of the activity to determine the path of workflow execution.

The output is triggered based on the http Status Code returned by the activity. Alternatively, the output Status

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Code can be saved in the AVEVA Work Tasks System variable.
The variable has to be prefixed with "^" followed by "Activity Name _StatusCode". For example, if the activity name is Invoke Web API1, then variable name should be "^Invoke Web API1_StatusCode" and the data type should be "string".

The Error Message can be saved in the AVEVA Work Tasks system variable.
The variable has to be prefixed with "^" followed by "Activity Name _WebAPIErrorMessage". For example, if the activity name is Invoke Web API1, then variable name should be "^Invoke Web API1_WebAPIErrorMessage" and the data type should be "string".

The additional error information (i.e. http custom content message) can be saved in the AVEVA Work Tasks system variable.
The variable has to be prefixed with "^" followed by "Activity Name _WebAPIErrorDetails". For example, if the activity name is Invoke Web API1, then variable name should be "^Invoke Web API1_WebAPIErrorDetails" and the data type should be "string".

Note: During the execution of Web API, the error details are displayed in the "Step Details" page. These error details are fetched as-is from the "http custom content message". To display the clear and readable messages, make sure that the content for "http custom content message" is properly formatted in the Web APIs.

You can use any HTTP client to invoke your web API. In fact, you can invoke it directly from a web browser. When the Web API framework receives a request, it routes the request to an action. To determine which action to invoke, the framework uses a routing table.

Activity Properties

Specify correct values for the different properties in the Activity Properties area to configure the Invoke Web API activity.

To access the activity properties, click the Activity Properties tool in the Tool Bar or by select the appropriate option from the context menu for the Invoke Web API activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

Use the Name and Description properties to enter the the name and description for the activity. This property is optional, and can be set if required.

- **Name**: Specify a name for the activity.
- **Description**: Specifies a brief description to be displayed for the activity.

Configuration

Use the configuration properties to configure the functionality of the activity.

Configure Web API: Use this property to configure the Web API to be called. Web APIs are frameworks to build HTTP services for browsers and mobile devices.

For more information about this activity property, see Configure Web API.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Exception Behavior

Each activity has the **Raise Error on Unhandled Linked Output** property. Configure this property to raise an error if the output is not linked to the next activity in the workflow. You can also use this property to define and configure email templates and users for the exception mails. This property is optional, and can be set if required.

See Mapped Error Outputs for more details.

Looping Behavior

Use this property to specify the looping behavior of the activity. This property is optional, and can be set if required.

Loop through this action if output is: Specify the condition for looping the activity. Select any one of the following activities from the list:

- No Loop
- BadGateway(502)
- BadRequest(400)
- Error Encountered
- Forbidden(403)
- GatewayTimeout(504)
- HttpVersionNotSupported(505)
- InternalServerError(500)
- MethodNotAllowed(405)
- NotAcceptable(406)
- NotFound(404)
- NotImplemented(501)
- ProxyAuthenticationRequired(407)
- RequestTimeout(408)
- ServiceUnavailable(503)
- Successful
- Unauthorized(401)
- Unsuccessful
- UnsupportedMediaType(415)

Configure Web API

You can configure the Web APIs in the **Web API Activity Configuration** window. This window lists all the Web APIs configured for the application or workflow.

Http Method

The **Invoke Web API** activity supports the following HTTP request methods:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **GET**: The GET method retrieves information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.
- **POST**: A POST request sends data to the server, for example customer information, file upload, using HTML forms.
- **PUT**: A PUT request replaces all current representations of the target resource with the uploaded content.
- **DELETE**: A DELETE request removes all current representations of the target resource given by the URI.

Web API

Web APIs are frameworks to build HTTP services for browsers and mobile devices.

If the Web API is Windows authenticated, then the status of the Windows Authentication must be set to **Enabled** for the **EnterpriseConsole** site under **AVEVAWorkTasksSites** site in the **IIS Manager**.

If the Web API requires User Access Token authentication, provide the current logged in user's access token for authentication.

Note:
- If the repository is configured with AVEVA Identity Manager, Azure Active Directory, and AVEVA Connect, the User Access Token authentication does not work.
- If User Access Token is configured for the Web API and associated with Scheduler Event Association, the workflow fails.
- In an IDE environment, Web API with User Access Token does not work.

If the Web API requires Service to Service Access Token for authentication, you need to provide **Authority URL**, **Audience**, **Client ID**, **Client Secret**, and **Scope**.

In the **Authority URL**, provide the token end-point of the respective repository provider. For example,

- AVEVA Connect: `{authority url}/oauth/token`
- Azure Active Directory: `{authority url}/oauth2/v2.0/token`
- AVEVA Identity Manager: `{authority url}/connect/token`

Headers

Headers is used to view, add, and delete Header keys and values while configuring the Web API activity. Select the **Yes** option for **Show Header** to view, add, and delete Header keys and values. You can add Header value using Expressing Editor.

The **Headers** tab is displayed, only if the Header Key and Value is configured in the Web API resource file. You cannot add, edit, or delete the Header Keys and Values.

Note: You cannot set the **Headers** as optional i.e. the **Optional** property of the **Headers** cannot be set to **YES**. For each Header, the Key and Value are the mandatory properties.

Request URI

Request URI is a Uniform Resource Identifier and identifies the resource upon which to apply the request.

Example:

- Request URI without parameters: **api/product/1** for **http://localhost/MySite/api/product/1**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Request URI with parameters: **api/{product}/{ID}** for **http://localhost/MySite/api**

Use System Authentication

Select the **Use System Authentication** check box to use the access token of the Workflow Engine Service.

POST Data

Passing data to POST and PUT Methods

You can pass data to POST and PUT methods either through parameters or through the request body. If you pass data through request body, use the adjacent text box for the body content. If web API method uses [FromBody] attribute, select the **Yes** option for the **FromBody Attribute** property.

Content Type

You can configure the content type of the Post Data. The available options are **JSON**, **XML**, and **URL ENCODED**.

This field is displayed only when the value of **Http Method** is set to **POST** or **PUT**.

Encoding

You can configure the encoding type while configuring the Web API activity. The supported encoding types are **UTF-7**, **UTF-8**, and **UTF-16**. The **Default** encoding type is used for backward compatibility.

Output Settings

Save output type

Use the options in this field to save the return value from the Web Method so that you can use it in subsequent activities. You can save the return value as either a **Variable** or as an **XML Variable**.

Save output in

To save the output as a **Variable**, select the variable that you created in the **Start** activity. To save the output as an XML Variable, select the **XML Variable** that you created in the **Start** activity. You can also create the XML Variable using the **Create New** option. For more information on creating XML Variable, see Create XML Variable.

When you create a new XML Variable for Web API, we suggest you give sample Web API with multiple child nodes. This will add **ChildNode** node to the XML Variable structure. If the sample Web API contains only one child node, it may not add the ChildNode node to the XML Variable structure, and the nodes may not get mapped properly in the workflow.

You can also overwrite the XML Variable using the **Update XML Variable** option after modifying the Web API configuration.

Invoke Web API: Configure OpenAPI

AVEVA Work Tasks allows you to configure OpenAPI, and it is compliant with Swagger 2.0 and Swagger 3.0.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To configure the OpenAPI

1. Right-click the **Web API Activity**, and then click **Activity Properties** option. In the Activity Properties window, click **Configure Web API**.

   The **Web API Activity Configuration** window opens.
2. In the **Web API** drop-down list, select the Web API list item, which is an OpenAPI.

   This loads the OpenAPI data, which gets stored when OpenAPI list item is created using OpenAPI.
3. In the **Operation Group** dropdown list, select the operation group provided in the JSON file of the selected OpenAPI. These groups are the tags or the logical groupings of the selected OpenAPI.

   Tags are used as Operation Group. Though Tag is an optional element in the OpenAPI Specification, if available, it will be considered as Operation Group item.

   If Tags is empty/not available, then it will not list out in Operation Group, but the operation will be directly listed under All (by default) Operation Group.
4. In the **Operation** drop-down list, select the operation ID of the OpenAPI defined for the selected Operation Group in the JSON file of the selected OpenAPI.

   The OperationId is considered as Operation and will be listed out in the drop-down.

   The OperationId is an optional, unique element as per the OpenAPI Specifications. But if any of the OperationId is duplicate, then the same will be logged in the Logger.

   If the OperationId is not present, then Path_HttpMethod will be displayed as Operation under the Operation dropdown.

   Based on the Operation Group selection, the Operation will be listed in the Operation dropdown.

   If there are no Tags available for some operations, then they will appear in All Operation Group along with all other operations.

   When the Operation is selected, the HTTP methods (GET, PUT, POST, DELETE, or PATCH) and the associated parameters are retrieved from the JSON file of the selected OpenAPI.

   Note: If Operation ID is not available, the Operations Group tag is considered.
5. In the **HTTP Method**, the HTTP method is displayed when you select Operation, and this is a disabled field.
6. The **Headers** tab is displayed only if the Header Keys and Values are configured in the OpenAPI resource file. You cannot add or delete the Header Keys and Values for the OpenAPI.

   Note:
   - The **Show Header** option is available to hide/unhide the **Headers** tab in the **Web API Activity Configuration** page. Select **Yes** if you want to view, add, and delete the Header Keys and the Values, else select **No**. You can add the Value using Expression Editor.
   - You cannot set the **Optional** property of the **Header**. You cannot provide the Key, but you need to provide the Value. If the **Optional** property is **YES** and the Value is empty, the value will not be sent to the Web API and will be considered null. If the **Optional** property is **NO**, you need to provide the Value.
7. In the **Parameters** and **Body** tabs, enter the details given in the respective tables in this topic.
8. In the **Output settings**, configure the **Save output type** and **Save output in**.

   To save the output in a **Variable**, select the variable that you created in the **Start** activity. To save the output in an **XML Variable**, select the XML variable that you created in the **Start** activity. You can also create the XML variable using the **Create New** option from the **Save output in** dropdown.

   You can overwrite the XML Variable using the **Update XML Variable** option after modifying the OpenAPI configuration.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: If OpenAPI specification JSON file does not have a response for the selected Operation, then it will prompt the user as "The selected operation does not return any value."

9. Click **Save**.

   The OpenAPI gets configured.

Parameters Tab

This tab is displayed if the selected Operation has parameters.

In the Parameter's grid, the following parameters are listed:

- Path parameters: the parameters' value is part of the operation’s URL
- Query parameters: the parameters that are appended to the URL

Note: Header and Cookie parameters are not supported.

The table describes the fields available under **Parameters** tab.

Fields	Description
No	It displays the serial number of the parameter and represents its sequence.
Optional	It displays **YES** or **NO** values. **YES**, denotes that you can either enter the value for the parameter in the **Value** field or leave it blank. **NO**, denotes that you must enter the value for the parameter in the **Value** field. **Note**: For the path parameter, the Optional field is always NO, but for query parameter it can be either YES or NO.
Name	It displays the name of the parameter.
Data type	It displays the data type of the parameter. The supported data types are boolean, decimal, numeric, and string. **Note**: The OpenAPI-specific datatype formats are not validated.
Value	Enter the value to be passed for the parameter. The value is validated against the data type. The value can be hard-coded or can be passed through the Expression Editor. **Note**: The value passed using Expression Editor is not validated for the data type.

Body Tab

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

This tab is displayed if the selected Operation is POST, PUT, or PATCH.

It is used to pass PostData to the OpenAPI, and the PostData can be passed as raw format or in the form of parameters, hence two options - **Parameters** and **Raw** are available with the Radio buttons.

Both Parameters and Raw are validated based on the selection of Content Type.

Note: In the **Body** tab, it is mandatory to provide data either in the **Parameters** or the **Raw** option.

Parameters:

The request body is populated in the form of the name, data type, and the value to map individual parameters with a hard-coded value or to a dynamic value that is build-up through the Expression Editor.

The table describes the fields available under **Body** tab for **Parameters**:

Fields	Description
No	It displays the serial number of the parameter and represents its sequence.
Optional	It displays **YES** or **NO** values. **YES**, denotes that you can either enter the value for the parameter in the **Value** field or leave it blank. **NO**, denotes that you must enter the value for the parameter in the **Value** field. **Note**: For the path parameter, the Optional field is always NO, but for query parameter it can be either YES or NO.
Name	It displays the name of the parameter.
Data type	It displays the data type of the parameter. The supported data types are boolean, decimal, numeric, and string. **Note**: The OpenAPI-specific datatype formats are not validated.
Value	Enter the value to be passed for the parameter. The value is validated against the data type. The value can be hard-coded or can be passed through the Expression Editor. **Note:** The value passed using Expression Editor is not validated for the data type.

On the Body parameters, constitute the PostData, and pass it to the OpenAPI during execution.

Raw:

In this tab, you can provide raw data as PostData, either a hard-coded value or dynamic value that is built up through the Expression Editor.

The table describes the fields available under **Body** tab for **Raw** data:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Fields	Description
Post Data	Enter the value you want to pass to POST, PUT, and PATCH methods through the body. The value should be in the format based on the content type selected in the Content Type dropdown.

Note:
- For more information on PATCH method, see PATCH Method.
- Passing empty/null parameter in RAW JSON/XML format as PostData is not supported. The user needs to provide correct JSON/XML data in RAW format without null/empty values.
- During design, the value provided as raw data to pass to the POST or PUT method through the body is not retained when you change the operation (PUT or POST).

PATCH Method

The PATCH method is used to update an existing resource. It specifies only the changes and does not replace the entire resource.

The supported operations are add, remove, replace, move, and copy.

For more information on operations and the examples, see https://docs.microsoft.com/en-us/aspnet/core/web-api/jsonpatch?view=aspnetcore-3.1

You can use the PATCH method to update the configuration in Web API and Open API. For Web API, you can define the parameters only in the **Parameters** tab available based on the Request URL value. For Open API, you can define the parameters in the **Parameters** tab and **Body** tab.

Parameters Tab

Under this tab, if the Patch has any parameters, then those will be listed.

The table describes the fields available under **Parameters** tab.

Fields	Description
No	It displays the serial number of the parameter and represents its sequence.
Optional	It displays **YES** or **NO** values. **YES**, denotes that you can either enter the value for the parameter in the **Value** field or leave it blank. **NO**, denotes that you must enter the value for the parameter in the **Value** field. **Note**: For the path parameter, the Optional field is always NO, but for query parameter it can be either YES or NO.
Name	It displays the name of the parameter.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Fields	Description
Data type	It displays the data type of the parameter. The supported data types are boolean, decimal, numeric, and string. **Note**: The OpenAPI-specific datatype formats are not validated.
Value	Enter the value to be passed for the parameter. The value is validated against the data type. The value can be hard-coded or can be passed through the Expression Editor. **Note**: The value passed using Expression Editor is not validated for the data type.

Body Tab

This tab is used to pass PostData to the Web API, and the PostData can be passed as raw format or in the form of parameters, hence two options - **Parameters** and **Raw** are available with the Radio buttons.

Both **Parameters** and **Raw** are validated based on the selection of Content Type.

Note: It is mandatory to provide data either in the **Parameters** or the **Raw** option.

Parameters:

In this tab, you can specify only one PATCH operation.

The request body is populated in the form of the value, path, and op to map individual parameters with a hard-coded value or to a dynamic value that is build-up through the Expression Editor.

The table describes the fields available under **Body** tab for **Parameters**:

Fields	Description
value	Enter the property value or an object value. **Note:** The value passed using Expression Editor is not validated for the data type.
operationType	The OpenAPI specifications has this property, but is not applicable.
path	Enter the path of an array element or a property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Fields	Description
op	Enter the operation like add, remove, replace, move, or copy that you want to perform.

Raw:

In this tab, you can provide raw data as PostData, either a hard-coded value or dynamic value that is built up through the Expression Editor.

The table describes the fields available to provide **Raw** data:

Fields	Description
Post Data	Enter the operations with the property value or an object value you want to pass through the body using the PATCH method. You can specify multiple Patch operations in the JSON format.

Create XML Variable

In the Web API Activity Configuration window, use the **Create New** option in the dropdown list of **Save output in** to create a new XML Variable.

Note: The value returned from a Web API should have a valid and non-localized JSON. In the case of localized JSON, store the returned value in a String Variable instead of an XML Variable.

To create XML Variable for Generic Web API

In the Web API Activity Configuration window,

1. Enter the required data i.e. HTTP Method, Parameters (if not optional), Content Type, and Post Data.
2. Select the **XML Variable** in the **Save output type** dropdown list.
3. Select the **Create New** option in the **Save output in** dropdown list.

   The **Create Web API XML Variable** window opens.
4. Enter the name for the XML Variable. The XML Variable name with a single character is not supported.
5. Select the **Yes** check box for **Always include ChildNode in the XML schema** to add **ChildNode** under Root node to hold multiple items.

   Note: If the Web API selected for creating XML Variable returns multiple nodes, it will add the ChildNode node by default. But if you select the **Yes** checkbox and if the Web API returns even a single node, it will add the ChildNode node. This is applicable only for JSON returning Web APIs and not for XML returning Web APIs.
6. Under **Parameters** tab, enter the values to be passed against the listed parameters.
7. Click **Save**.

   Note: The Web API will be executed in the Web API server during this operation. Based on the Web API

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

functionality, there is a possibility of data update or delete.

To create XML Variable for OpenAPI:

In the Web API Activity Configuration window,

1. Select the Operation to be performed.
2. Select the **XML Variable** in the **Save output type** dropdown list.
3. Select the **Create New** option in the **Save output in** dropdown list.

   The **Create Web API XML Variable** window opens.

   Note: If the selected Operation does not have any response in the specification, it displays the message "the Operation does not return any value" and does not open the **Create Web API XML Variable** window.
4. Enter the name for the XML Variable. The XML Variable name with a single character is not supported.
5. Select the **Yes** check box for **Always include ChildNode in the XML schema** to add **ChildNode** under Root node to hold multiple items.

   Note: If the Web API selected for creating XML Variable returns multiple nodes, it will add the ChildNode node by default. But if you select the **Yes** checkbox and if the Web API returns even a single node, it will add the ChildNode node. This is applicable only for JSON returning Web APIs and not for XML returning Web APIs.
6. Click **Save**.

Note:
- The supported data types are boolean, decimal, numeric, and string.
- XML Variables without root elements are created with a root node named **root**. In earlier versions, it was **Root**. In case of upgrade and if XML Variables are updated, then correct this change in respective activities and variables. Mainly affected in Xml Variables of type Iterator, Activities like Script, Update Variable, and properties using Expression Editor.

Script Activity

Activity Description

The Script Activity enables the execution of script written in languages, such as Jscript and VBScript, from the workflow. When the Script activity is included in the workflow, the activity calls the Run method in the script. The references to the assemblies in the script are taken from the activity properties. The string value returned from the Run method of the script is assigned to the Output parameter. The Script activity executes the script and gives the return values specified in the properties. The link matching this value is picked up to continue workflow execution.

Features of Script Activity

- Compilation of the script code is made at design time to enable discovery of syntax or other errors during design.
- Loading all scripts defined through script activities as a single assembly. All script activities are compiled as a single assembly and only one assembly is loaded during workflow execution. This significantly improves performance.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- When a workflow is published, the script details are stored in the SWScriptCode table to enable identification of source code and assembly references.

  Note: Framework 4.0 has a library named System.Action and results in an ambiguity between the System.Action and Workflow.NET.Action. To avoid this ambiguity, it is recommended to name the Action class with complete namespace as 'Workflow.NET.Action' if it is used in the scripts inside script activity, while running BPM.NET 2011 SP2 in Framework 4.0.

Activity Properties

The Script activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Script activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

The following image shows the Name & Description properties of Script Activity:

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Programming

The properties in the Programming group have been described below. You can use these properties to program the script activity.

The following image shows the Programming Properties of the Script Activity:

Language - This property specifies the language of the script to be run. This property can be set to 'C#.NET', 'Jscript' or 'VBScript' depending on the scripting language required.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Code - This property specifies the code of the script to be run.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the Code property

1. To enter the code, click on the button for the 'Code' property to open the property window.
2. In this window enter the code directly in the text editor.
3. If you want to assign the code dynamically during run-time, you can do so using the **Field** controls. For example, you can add a variable or content pointing to the code from the two drop downs for **Field** and click on the **Append** button. This will add the placeholder for the variable or content in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the code.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. After adding the code, click the **Update** button to save it
7. Click the **Close** button to close the window without saving the code.

See Script - Code for a detailed description of the Code property window.

Reference - This property specifies the global objects that can be referenced in the script.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the References property

1. To specify the References, click on the button for the 'References' property to open the property window
2. In this window, enter the reference object in the **References** field
3. Next, enter the corresponding assembly path in the **Assembly Path** field
4. After entering the return value details, click the **Add Row** button to add this reference and display a new row for entering another reference
5. Remove rows if necessary by selecting them and clicking the **Remove Selected** button
6. After specifying the references, click the **Update** button.

See Script - References for a detailed description of the References property window.

Output Values

The properties in the Output Values group have been described below. You can use these properties to specify the output values for the activity.

The following image shows the Output Values Properties of the Script Activity:

Return Values - This property allows the user to specify the return values for this action. These values become the conditions for the workflow engine to pick up the next action to execute.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the Return Values property

1. To specify the Return Values, click on the button for the 'Return Values' property to open the property window
2. In this window, enter the return value in the **Return Values** field
3. After entering the return value details, click the **Add Row** button to add this return value and display a new row for entering another return value
4. Remove rows if necessary by selecting them and clicking the **Remove Selected** button
5. After specifying all the return values, click the **Update** button.

See Script: Return Values for a detailed description of the Return Values property window.

Action outputs

The Script Action does not have any standard return value. It will output the value specified by the user in the Return Values property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

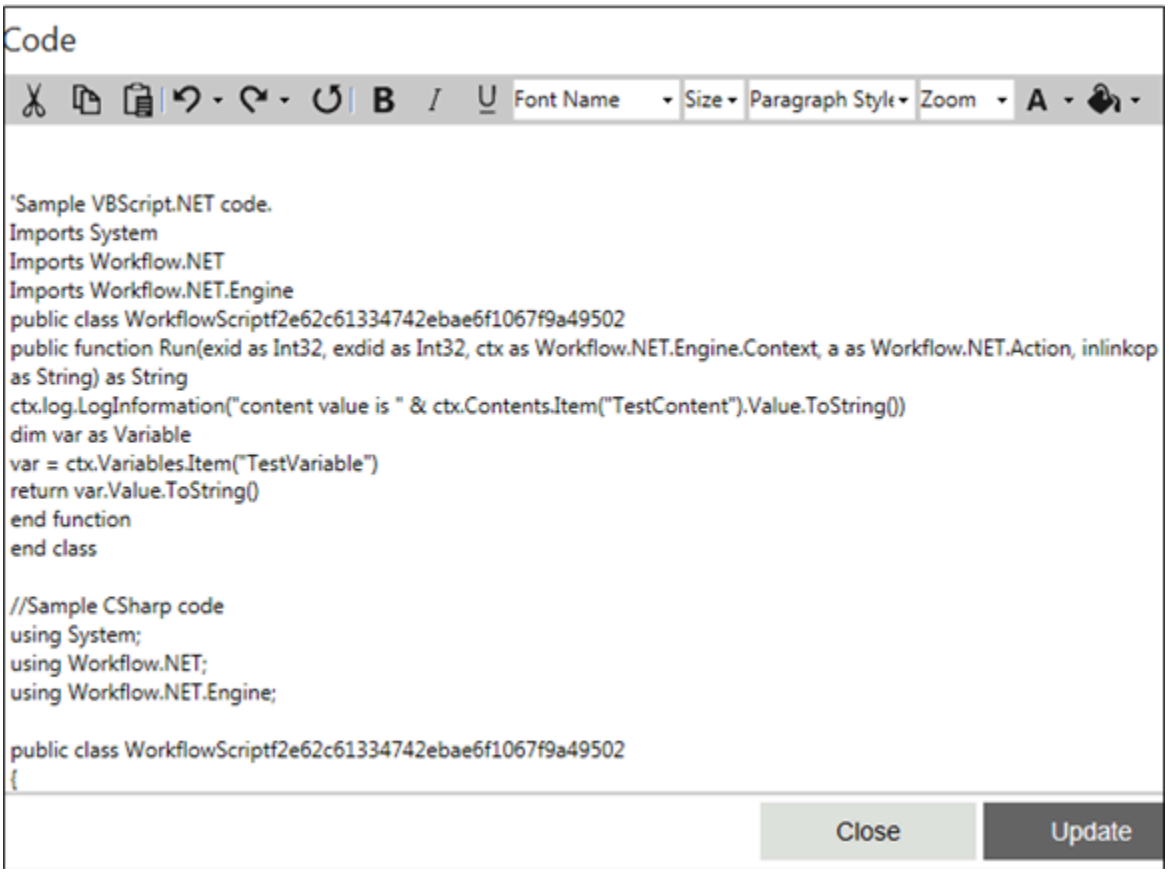
Note: In case Workflow containing Script activity is converted to Quickflow and the Script activity contains reference to any System binaries, user would need to change the assembly path.

Script: Code

Code Property Window

You can set the code of the script in the **Code** property window. To open this window, click the button for the "Code" property.

The following image shows the script **Code** property window:



The **Code** property window contains a text editor to enter the script code. You can do this either by entering a discrete value directly in the text editor. You can use the text tool bar in the text editor to manage and format the code.

The text tool bar contains the following options:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Options	Shortcut Key	Description
Cut	Ctrl+X	Cut text
Copy	Ctrl+C	Copy text
Paste	Ctrl+V	Paste copied text
Undo	Ctrl+Z	Undo the last action
Redo	Ctrl+Y	Redo an action that you undid
Repeat Last Command	F4	Repeat the last command that was executed
Bold	Ctrl+B	Make text bold
Italic	Ctrl+I	Italicize text
Underline	Ctrl+U	Underline text
Font Name	-	Select the font for the parameter
Size	-	Enter the font size
Paragraph Style	-	Select a paragraph style
Zoom	-	Zoom in and zoom out the view
Foreground Color	-	Set text color
Background Color	-	Set the background color
Print	-	Print the text

The **Code** property window also contains the following tabs:

- **Update**: Use this tab to save the parameter.
- **Close**: Use this tab to close this window.

Script: References

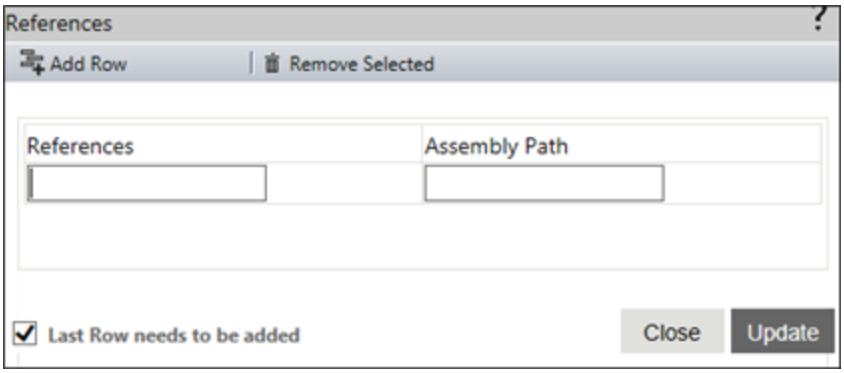
References Property Window

You can set the references for the script in the **References** property window.

To open this window, click the button for the "References" property. It displays the references in a grid. The grid contains a blank row where you can enter the parameters for new references.

The following image shows the **References** property window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

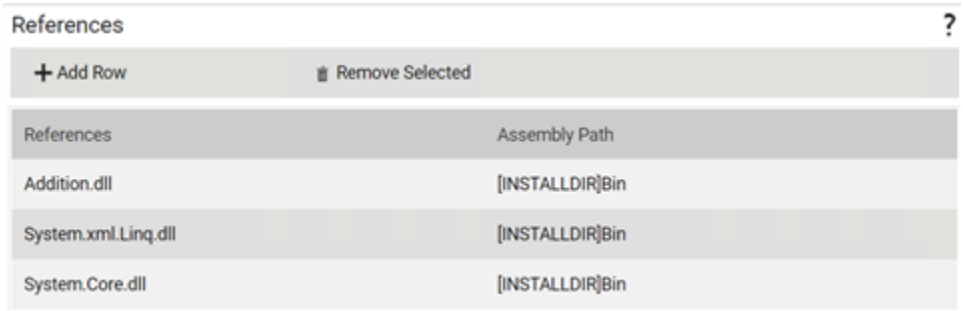


The **References** property window contains the following fields and tabs:

- **Add Row**: Use this tab to add the newly entered reference, and display a new row to the grid where you can add another reference.
- **Remove Selected**: Use this tab to remove the reference selected in the grid.
- **References**: Use this field to enter a new reference object.
- **Assembly Path**: Use this field to enter the assembly path for the new reference.
- **Update**: Use this tab to save the newly added reference(s).
- **Close**: Use this tab to close the property window without saving the newly added reference(s).
- **Last Row needs to be added**: Select this check box to save the last row added.

Enter the relative path, **[INSTALLDIR]Bin** as the assembly path to refer to the external assemblies that are referenced in script action and are available in the AVEVA Work Tasks installed Bin folder.

The following image shows the external assembly relative path:



Script: Return Values

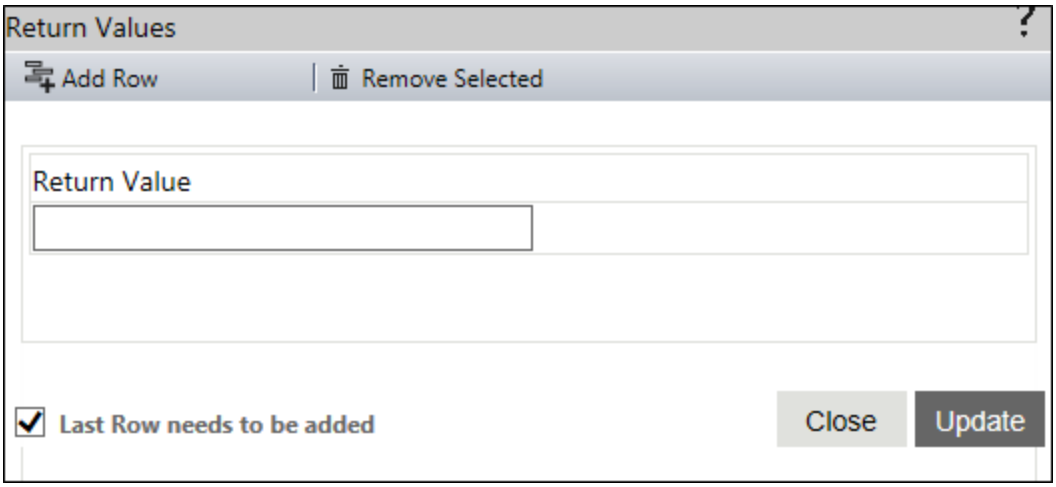
Return Values Property Window

You can set the return values for the Script in the **Return Values** property window.

To open this window, click the button for the "Return Values" property. It displays the values in a grid. The grid contains a blank row where you can enter a new return value.

The following image shows the **Return Values** property window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The **Return Values** property window contains the following fields and tabs:

- **Add Row**: Use this tab to add the newly entered return value, and display a new row to the grid where you can add another value.
- **Remove Selected**: Use this tab to remove the return value selected in the grid.
- **Return Values**: Use this field to enter the return value.
- **Update**: Use this tab to save the newly added return value(s).
- **Close**: Use this tab to close the property window without saving the newly added return value(s).
- **Last Row needs to be added**: Select this check box to save the last row added.

File Operations Activity

Activity Description

The File Operation Activity can be used to perform file transfers from within the workflow. It can be used to transfer files within local networks as well as using HTTP and FTP protocols.

File Transfers Supported

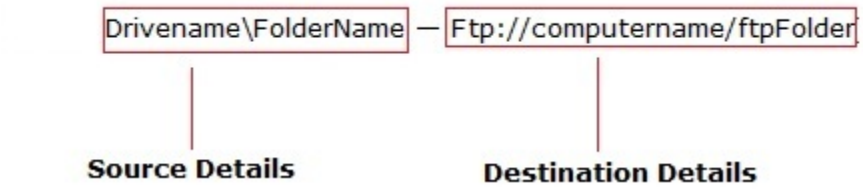
The list of supported file transfers:

- UNC — HTTP - Example: \\computername\sharename\filename — Http://computername/ WebFolder /filename
- File — HTTP - Example: \\DriveName\foldername\filename — Http://computername/ WebFolder /filename
- File — FTP - Example: \\Drivename\ftpFolder\FileName — Ftp://computername/Sharename/filename
- FTP — Filen - Example: Ftp://computername/Sharename/filename — Drivename\FolderName\
- HTTP — UNC - Example: Http://computername/WebFolder /filename — \\computername\sharename\filename
- UNC — UNC - Example: \\computername\sharename\filename — \\computername\sharename\filename
- File — File Example: \\DriveName\foldername\filename — DriveName\foldername\filename

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- UNC — HTTP (Directory Transfer) Example: \\Computername\sharename\filename — Http://ComputerName/WebFolder
- File — HTTP (Directory Transfer) Example: \\Drivename\FolderName — Http://computername/WebFolder
- File — FTP (Directory Transfer) Example: Drivename\FolderName — Ftp://computername/ftpFolder

  Note: The above examples are in the following format: Source Details — Destination Details



The File Operation activity is not supported if the Source and Destination paths are in the combinations given below:

- FTP and HTTP
- FTP and FTP
- HTTP and FTP
- HTTP and HTTP

Variable for getting the list of files that were transferred

The list of files that were transferred using the File Operations activity can be retrieved using a variable of type array, with the following syntax:

^'+CurrentActivity.Name+ 'FilesUploaded'

For example, ^TransferFileOperationFilesUploaded

Variable for getting the list of files that were not transferred

The list of files that were not transferred using the File Operations activity can be retrieved using a variable of type array, with the following syntax:

'^'+CurrentActivity.Name+ 'FilesNotUploaded'

For example, ^TransferFileOperationFilesNotUploaded

Activity Properties

The File Operation activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the File Operation activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

SourceDetails - This property allows the user to specify the details of the file source, like its location, the transfer protocol, etc.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the SourceDetails property

1. To specify the details of the file source, click on the button for the 'SourceDetails' property to open the property window.
2. In this window specify the file operation (transfer) protocol by selecting an option from the drop down for the **Protocol** field.
3. Next assign the URI/UNC of the source file or directory as the value for the **Location** field. Build an expression to assign the location details by selecting the operator, type of value (Value, Variable, Content, XMLVariables), and the value for the property from the corresponding drop downs.
4. In the same way, you can assign a search pattern to filter the type of file to be copied or transferred as the value for the **Pattern** field. For example, you can specify "*.doc" as the filter pattern. This will select only files with the extension .doc for copying or transferring.
5. You can also assign the user name and password of the source file/directory in the User Name and Password fields respectively, by following the same procedure that is used in step 3.
6. In addition, you can assign the domain name and port number of the source in the **DomainName** and **PortNumber** fields respectively, by following the same procedure that is used in step 3.
7. After adding the parameter, click **Submit** to save it.

See File Operations: Source Details for a detailed description of the Source Details property window.

DestinationDetails - This property allows the user to specify the details of the file destination.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the Destination Details property

1. To specify the details of the file destination, click the button for the 'DestinationDetails' property to open the property window.
2. In this window specify the file operation (transfer) protocol by selecting an option from the drop down for the **Protocol** field.
3. Next assign the URI/UNC of the destination file or directory as the value for the **Location** field. Build an

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

expression to assign the location details by selecting the operator, type of value (Value, Variable, Content, XMLVariables), and the value for the property from the corresponding drop downs.

4. In the same way, you can assign a search pattern to filter the type of file to be copied or transferred as the value for the **Pattern** field. For example, you can specify "*.doc" as the filter pattern. This will select only files with the extension .doc for copying or transferring.
5. You can also assign the user name and password of the source file/directory in the **User Name** and **Password** fields respectively, by following the same procedure that is used in step 3.
6. In addition, you can assign the domain name and port number of the source in the **DomainName** and **PortNumber** fields respectively, by following the same procedure that is used in step 3.
7. After adding the parameter, click **Submit** to save it.

See File Operations: Destination Details for a detailed description of the Destination Details property window.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Action outputs

The File Operations Activity returns the following values -

Successful: Displayed when the file transfer is successful.

Failure: Displayed when the file transfer is unsuccessful.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

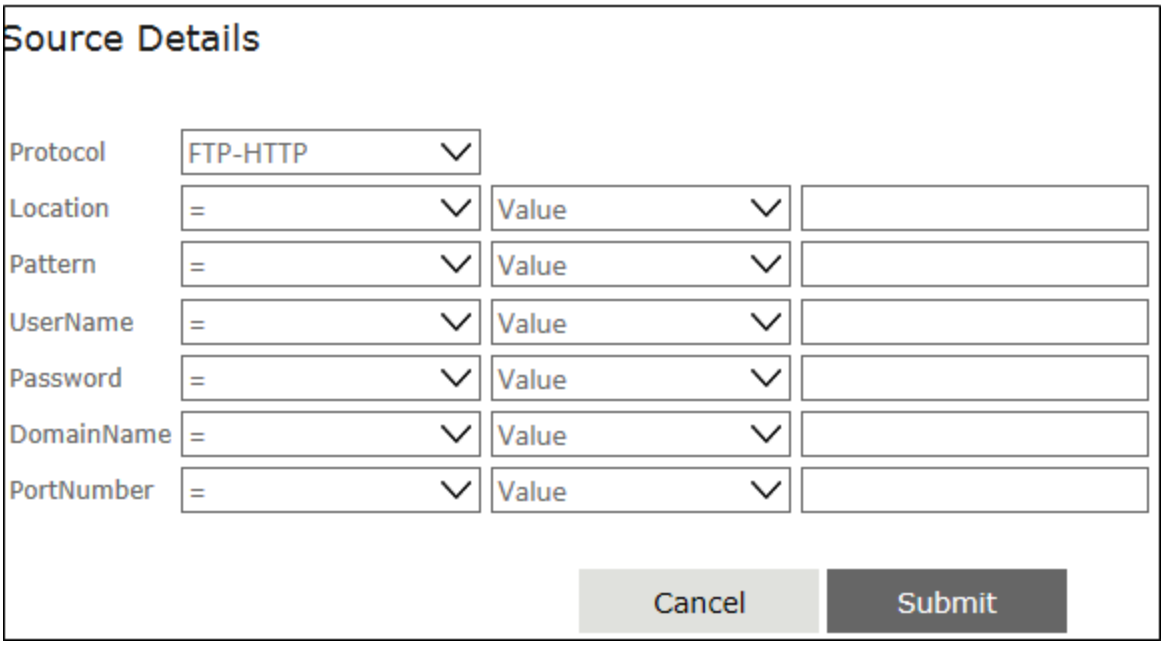
See Mapped Error Outputs for more details.

File Operations: Source Details

Source Details Property Window

You can specify the details of the file source in this property window. To open this window, click **Source Details** property. The **Source Details** property window appears as shown in the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The following table shows the fields of the **Source Details** property window:

Parameter	**Description**
Protocol	The required file transfer protocol, such as FTP-HTTP.
Location	The URI or UNC of the file or directory.
Pattern	Pattern to decide the file types to be transferred, such as *.doc,*.xls.
UserName	User name for the source.
Password	Password for the source.
Domain	Domain or machine of the source name.
PortNumber	Port number of the source server.

Each of the above parameter fields has the following controls.

- **Operator**: Select the operator for the expression from the drop-down menu. For example, "="
- **Value**: Select the type of value to assign to the parameter in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- **Value text box or drop-down menu**: Specify the actual value to assign to the parameter. The control displayed depends on the option selected in the Value column. If you select "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression. If you select "Variable", "Content", or "XMLVariables", there will be a drop-down menu instead of the field. This drop-down menu contains the variable collection, or content collection, or XMLVariables collection from which you can select an appropriate option.
- **Submit**: Click **Submit** to save the specified parameters.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

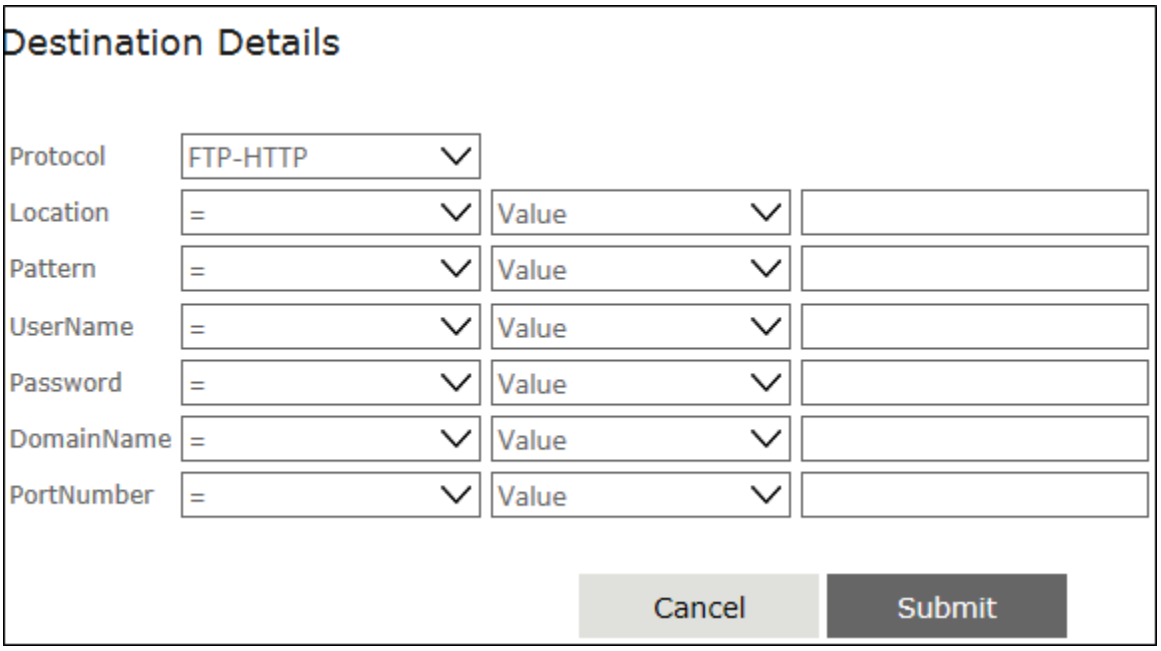
- **Cancel**: Click **Cancel** to cancel the settings and close the property window.

Note: In AVEVA Work Tasks, the validation for user credentials for FTP-HTTP is not provided. However, if a particular FTP needs authentication, then the user has to provide appropriate credentials.

File Operations: Destination Details

Destination Details Property Window

You can specify the details of the file destination in this property window. To open this window, click **Destination Details** property. The **Destination Details** property window appears as shown in the following image:



This window has the **Protocol** drop down where you can select the file transfer protocol. Below this drop down, it has controls to specify the following details.

- **Protocol** - If the selected protocol is FTP-HTTP, you can use FTP or HTTP protocol.

  For the HTTP protocol, you must enable the WebDAV Publishing feature for the web server. For more information about enabling the WebDAV Publishing feature, refer the Microsoft documentation.

  In the **Authentication** Feature in Internet Information Services (IIS) Manager, ensure that only **Windows Authentication** is enabled for the selected HTTP site.
- **Location** - You can specify the URI/UNC of the file or directory here.
- **Pattern** - You can specify the pattern to decide the file types to be transferred like *.doc,*.xls in this field.
- **UserName** - You can specify the user name for the destination here.
- **Password** - You can specify the password for the destination here.
- **Domain** - You can specify the domain/machine name of the destination here.
- **PortNumber** - You can specify the port number of the destination server.

Each of the above parameter fields has the following controls.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The Operator drop down - You can select the operator for the expression here. For example, "="
- The Value type drop down - You can select the type of value to assign to the parameter in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the parameter. The control displayed depends on the option selected in the preceding drop down. If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.

At the bottom of the window you will find the following buttons:

- **Submit** - You can click this button to save the parameters that have been specified.
- **Cancel** - You can click this button to cancel the settings and close this window.

Change Workflow Status Activity

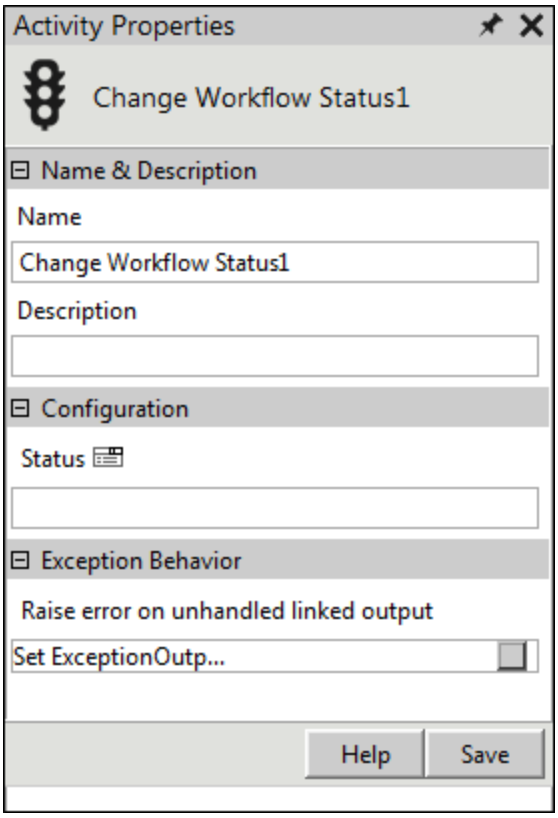
Activity Description

The Change Workflow Status Activity updates the status of the workflow in the database.

Activity Properties

The Change Workflow Status activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right-click menu for the Change Workflow Status activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Change Workflow Status Activity Properties

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Status - This property specifies the new status to be updated.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Activity outputs

The Change Workflow Status Activity does not return any value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

XML Action Activity

Activity Description

The XML Action Activity can be used to assign XML data for activity. This was possible earlier only by using Contents and Variables defined in the Start activity. The activity has been enhanced to accept XML data containing namespaces. The namespace manager that has been added to the activity handles namespaces within the XML data internally.

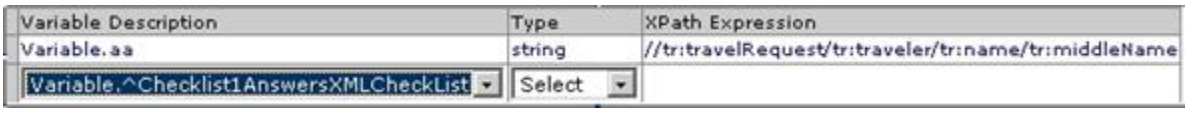
View an example of assigning XML data with namespaces to a variable

The namespaces in the following example used to illustrate the extraction of a particular data are highlighted in red.

```
<tr:travelRequest xmlns:xhtml="http://www.w3.org/1999/xhtml" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance" <tr:travelRequest xmlns:xhtml="http://www.w3.org/1999/xhtml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<tr:description></tr:description>
<tr:date xsi:nil="true"></tr:date>
<tr:traveler>
<tr:name>
<tr:prefix></tr:prefix>
<tr:givenName>guru</tr:givenName>
<tr:middleName>nayak</tr:middleName>
<tr:surname></tr:surname>
<tr:suffix></tr:suffix>
<tr:singleName></tr:singleName>
</tr:name>
</tr:traveler>
</tr:travelRequest>
```

The XPath expression to retrieve the middle name and assign it to a workflow variable 'aa' is given below. The image shown below illustrates the actual setting of the variable with this XML data.

// tr:travelRequest/ tr:traveler/ tr:name/ tr:middleName



Application Scenarios

The XML Action can be used with other activities to retrieve required XML data by defining XPath expressions. For example, it can be used with the Checklist activity to retrieve the Answers to Checklist Questions.

The XPath expressions for retrieving answers to Checklist Questions

The XML Activity can be used with the Checklist Activity to retrieve the Answers to the Checklist Questions. The XML data format for the Checklist Activity to be declared in the Start activity is '^'+ activity name +

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

'AnswersXMLCheckList' , e.g. '^Checklist1AnswersXMLCheckList'.

The XPath expressions to retrieve the Answers from the Checklist XML string are given below:

XPath with attribute:

```
//checklist/question[@name="RADIO"]/answer
//checklist/question[@name="DROPDOWN"]/answer
//checklist/question[@name="TEXTBOX"]/answer
//checklist/question[@name="MEMO"]/answer
//checklist/question[@name="CHECKBOX"]/answer
```

Activity Properties

The XML Activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the tool bar or by selecting the appropriate option from the right click menu for the XML Activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Input Values

The properties in the Input Values group have been described below. You can use these properties to specify the input values for the activity.

Get Variable - This property allows the user to specify the variable that holds the XML string. This variable is defined in the Start activity.

Property Type: Optional (This property supports backward compatibility.)

Get Data - This property allows the user to pass the XML data. This data can be passed in one of two forms: (1) XML String, or (2) XML File (where the complete path of the XML file is specified).

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Get Data property

1. To specify the XML data, click the button for the 'Get Data' property to open the property window.
2. In this window enter the XML data directly in the text editor. You can either specify the file path to an XML file here or enter the XML data directly.
3. If you want to assign the parameter dynamically during run-time, you can do so using the **Field** controls. For example, you can add a variable or content pointing to the XML data from the two drop downs for **Field** and click on the **Append** button. This will add the placeholder for the variable or content in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the XML data.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

undo and redo changes, zoom in and zoom out the view as well as print the text.

5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the XML data, click **Update** to save it.
7. Click to close the window without saving the XML data.

See XML Action: Get Data for a detailed description of the Get Data property window.

Output Values

The properties in the Output Values group have been described below. You can use these properties to specify output values for the activity.

Set Variables - This property allows the user to set the XPath to retrieve the required XML attribute from the XML string that is set in the Get Variable property.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Set Variable property

1. To set the XML data to selected variables, click the button for the 'Set Variables' property to open the property window.
2. In this window, select the variable from the drop down for the **Variable Description** field.
3. Next select the variable type from the **Type** drop down.
4. Specify the XPath Expression for the XML data to be assigned to the selected variable in the **XPath Expression** field.
5. After entering the details, click **Add Row** to add this variable setting and display a new row for assigning XML data to another variable.
6. Remove rows if necessary by selecting them and clicking **Remove Selected**.
7. After setting the XML data to the variables, click **Update**.

See XML Action: Set Variables for a detailed description of the Set Variables property window.

Configuration

The property Assign Empty Node Values in the Configuration group has been described below. You can use this property to configure the functionality of the activity.

Assign Empty Node Values

This property allows the user to choose whether the XML node value can be updated with an empty value or not.

If this property is set as "Yes", then the variables can be updated with the "Empty" node values and the activity will complete with a "Successful" output.

If this property is set as "No", then the variables will not be updated with the "Empty" node values and the activity will complete with an "UnSuccessful" output.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The XML Action Activity returns the following values -

Successful: Displayed when the activity is successful in retrieving the attribute from the XPath expression.

Unsuccessful: Displayed when the activity is unsuccessful in retrieving the attribute from the XPath expression.

Exception Behavior

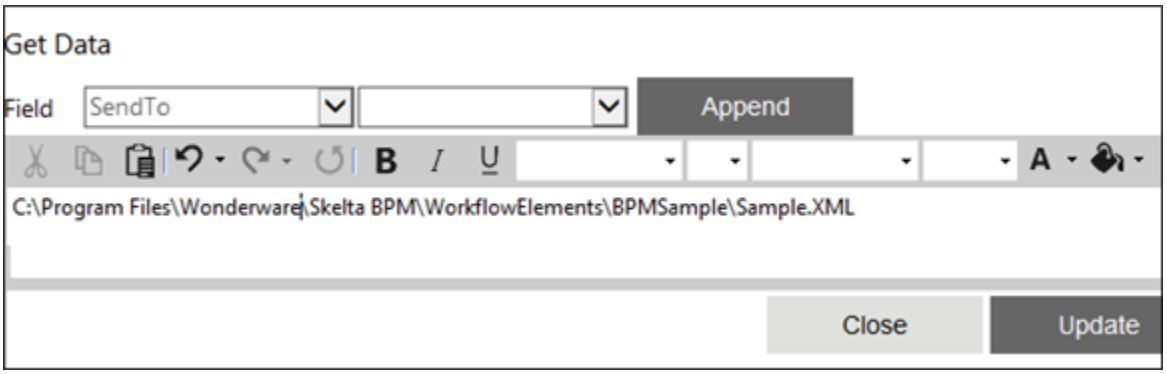
Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

XML Action: Get Data

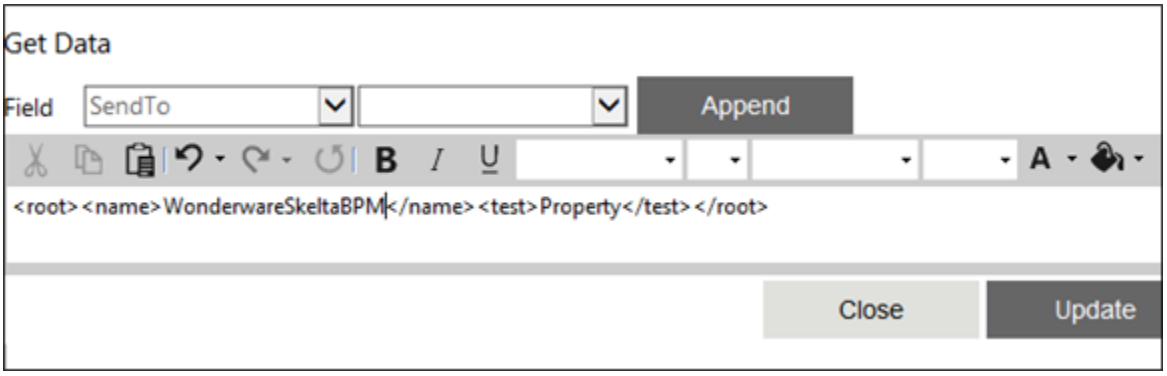
Get Data Property window

You can set the XML data in this property window. To open this window, click the button for the 'Get Data' property.

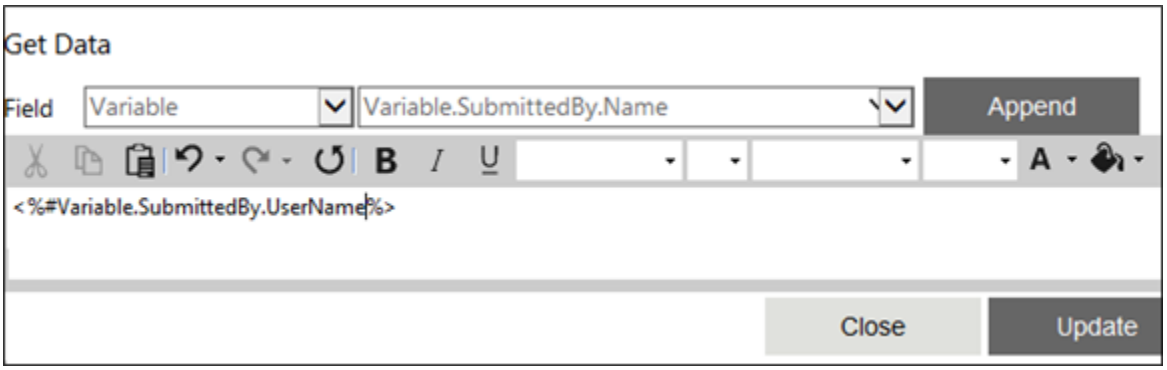


© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Get Data property - XML file path



Get Data property - XML data



Get Data property - XML data through variable

The window has a text editor to enter the XML data. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value or XML Variable here.
- The Value text box/drop down - You can specify the actual XML data. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the XML data in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the XML data.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the XML data details. It has the following text tools –

- **Cut (Ctrl+X)** - to cut text
- **Copy (Ctrl+C)** - to copy text

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Paste (Ctrl+V)** - to paste text
- **Undo (Ctrl+Z)** - to undo the last action
- **Redo (Ctrl+Y)** - to undo the last action
- **Repeat Last Command (F4)** - to repeat the last command that was executed

It also has the following formatting & other tools –

- **Bold (Ctrl+B)** - to make text bold
- **Italic (Ctrl+I)** - to italicize text
- **Underline (Ctrl+U)** - to underline text
- **Font Name** - to select the font for the XML data
- **Size** - to specify the font size
- **Paragraph Style** - to select a paragraph style
- **Zoom** - to zoom in and zoom out the view
- **Foreground Color** - to set text color
- **Background Color** - to set the background color
- **Print** - to print the text

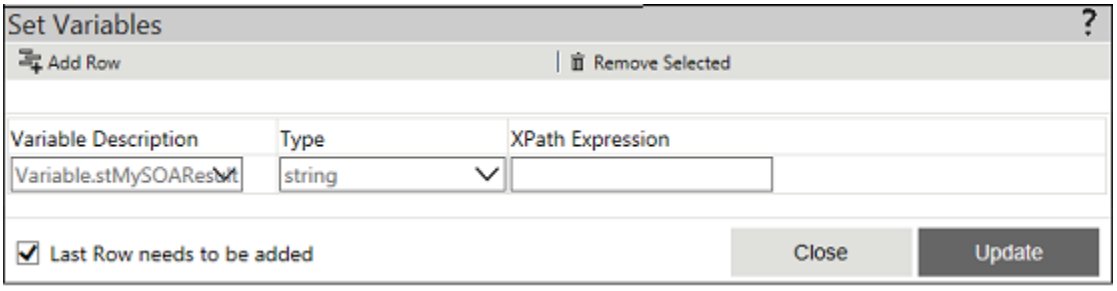
At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the XML data.
- **Close** - You can click this button to close this window.

XML Action: Set Variables

Set Variables Property window

You can set Xpath expressions to workflow variables in this property window. To open this window, click the button for the 'Set Variables' property. It displays the settings for the variables in a grid, which also has a blank row where you can enter the settings for a new variable.



The window has two tabs at the top to manage the rows of variable settings in the grid.

- **Add Row** - You can click this tab to add the newly entered variable settings and display a new row to the grid where you can add another variable settings.
- **Remove Selected** - You can click this tab to remove the variable selected in the grid.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The grid has the following parameters that you need to specify for the new variable.

- **Variable Description** - You can select the variable to which to assign the XML data from the drop down for this field.
- **Type** - You can select the variable type from this drop down.
- **XPath Expression** - You can use this field to enter the XPath Expression for the XML data to be assigned to the selected variable.

There are two buttons and a check box at the bottom of the screen.

- **Update** - You can click this button to save the newly added variable settings.
- **Close** - You can click this to close the property window without saving the newly added variable settings.
- **Last Row needs to be added** - You need to select this check box to save the last row added.

Engine Activities

The following activities are found under Engine Activities:

- **End Activity Execution Activity**: This activity enables the user to end or terminate the execution of an activity in a workflow.
- **End Child Workflow Execution Activity**: This activity enables the user to end a child workflow from the parent workflow.
- **BlockingOutput Activity**: This activity facilitates synchronous execution of workflow activities other than human activities with an active form.
- **Database Activity**: The database activity enables execution of queries, stored procedures, and functions from the workflow.
- **Decision Activity**: The Decision Activity can be used to insert a decision in the workflow.
- **ForEach-Loop Activity**: The ForEach Loop Activity is used to loop an activity through a collection within an XMLIterator Variable.
- **For-Loop Activity**: The For-Loop Activity can be used to start a loop within a workflow.
- **Logger Activity**: The Logger Activity can be used to log messages like information, warnings, and errors to the logger console from the workflow.
- **Loop Execution Step Entry Activity**: The user can use this activity if an instance level association between the End Activity Execution activity and the activity to be ended is required.
- **Next Loop Activity**: The Next Loop activity can be used as the terminal activity in a loop initiated using a For-Loop Activity.
- **Resource Variable Activity**: The Resource Variable Activity can be used to identify an actor to perform a workflow activity, through a variable.
- **Rule Activity**: The Rule Activity allows complex rules or expressions to be built into the workflow.
- **Workflow Activity**: The Workflow Activity enables the execution of a second workflow from a workflow.
- **Update Variable Activity**: The Update Variable Activity can be used to update the value of Variables and XmlVariable that can be used in the subsequent activities.
- **Update Parent Variables Activity**: The Update Parent variables activity can be used to update a workflow

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

variable from a second workflow, provided both workflows belong in the same repository.

- **Synchronize Activity**: The Synchronize Activity synchronizes the execution of different activities.
- **Set Workflow Output Variable**: The Set Workflow Output Variable activity can be used to set the output of the parent workflow from the child workflow.
- **XML Navigator**: The XML Navigator is used to either locate or navigate to a specific record in a collection within an XMLIterator Variable.

End Activity Execution Activity

Activity Description

This activity enables the user to end or terminate the execution of an activity in a workflow. The user can end an activity present either in the same (parent) workflow or the child workflow which is associated to the parent workflow.

Application Scenario

Consider a workflow with a 'Wait For Event' activity and an 'Approval' activity. The Wait For Event activity is executed only when a new list item is added, otherwise the activity will be in a waiting state. This scenario, however, requires that if the user executes the approval activity, the workflow need not wait for the execution of the Wait For Event activity and hence end the activity. In such scenarios, the End Activity Execution activity can be used. This activity helps the user to ensure that a workflow completes without any activity in waiting state.

Refer to the User Scenarios given in the *User Guide* for a detailed understanding of this activity.

Activity Properties

The End Activity Execution activity has to be configured by specifying appropriate values for different properties in the **Activity Properties** pane. This pane can be accessed by clicking on the Activity Properties tool in the **Tool Bar** or by selecting the Activity Properties option from the context menu of End Activity Execution activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the **Name & Description** group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property is to be used to specify a name for the activity.

Property Type: Mandatory (This property must be set.)

Description - This property can be used to specify a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in this group have been described below. You can use these properties to configure the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

functionality of the activity.

Activity Details - This property is used to specify the activity that is to ended and the workflow in which the activity is used.

Property Type: Mandatory (This property must be set to end the activity in a workflow.)

See End Activity Execution: Activity Details for a more detailed understanding of its properties.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Looping Behavior

The property in the **Looping Behavior** group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop-down field.

Property Type: Optional (This property needs to be set only if necessary.)

End Activity Execution: Activity Details

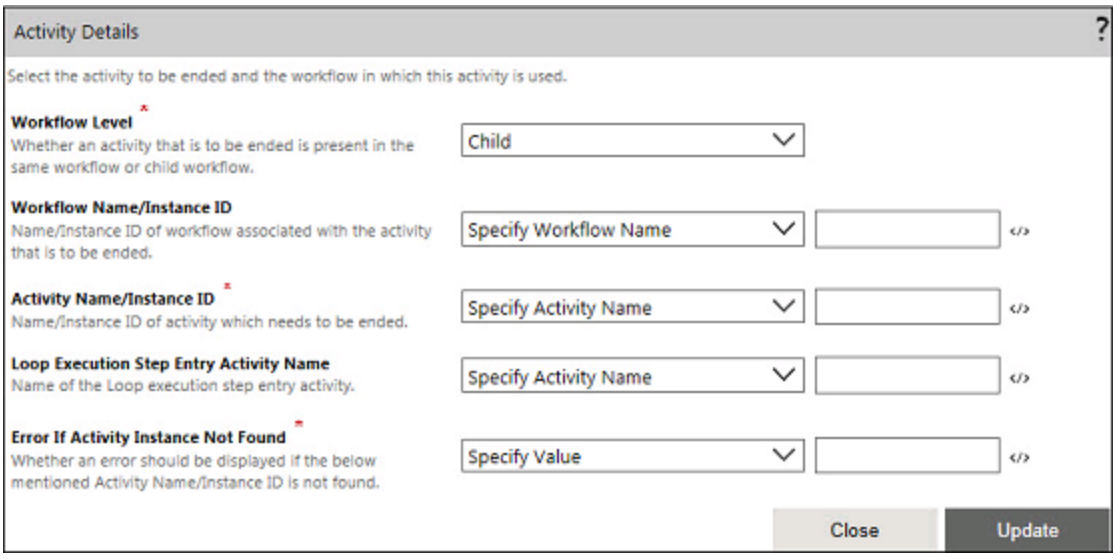
Activity Details property window

This topic gives you a detailed understanding of the **Activity Details** properties. The Activity Details properties are used to specify

- the workflow in which the activity to be ended is used.
- the activity which needs to be ended.
- the error that needs to be displayed if the activity that is to be ended is not found.
- the loop execution step entry activity to monitor the instances that are to be ended.

To open the **Activity Details** window, click the Activity Details button.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



This window has the following properties:

- **Workflow Level**: This property helps you to select the workflow level in which the activity to be ended is present. You can select either the parent workflow level or the child workflow level in which the activity to be ended is present.
- **Self:** You can select this option if the activity to be ended is present in the parent/same workflow.
- **Child**: You can select this option if the activity to be ended is present in a child workflow.
- **Specify Workflow Level**: If you do not know the workflow level, then you can use this option to specify the workflow level and type the workflow level in the adjacent field. Alternatively, you can use the expression editor, by clicking the icon , to select a variable assigned to the required workflow level. During runtime, this variable value is used to know the workflow level. You can assign the 'Self' or 'Child' values to a variable which will be used to know the value of this property.
- To know more about using expression editor, see Using Expression Editor to Build Expressions for Properties topic in *User Guide* section.
- **Workflow Name/Instance ID**: This property helps you to select the workflow name or its instance ID in which the activity to be ended is present. This option is enabled only if you have selected the 'Child' workflow level option.

You can select the activity name from the drop-down field. Alternatively, you can either select the **Specify Workflow Name** or the **Specify Workflow Execution ID** option.

- **Specify Workflow Name**: Select this option to specify the workflow name and type the workflow name in the adjacent field. Alternatively, you can click the expression editor icon and select a variable referring to the required workflow name in the expression editor window.
- **Specify Workflow Execution Instance ID**: If you do not know the workflow name, then select this option to specify the workflow execution ID and type the execution ID in the adjacent field. Alternatively, you can click the expression editor icon and select a variable referring to the required workflow execution instance ID in the expression editor window.
- **Activity Name/Instance ID**: This property helps the user to select the activity name or the instance ID of the activity to be ended. Alternatively, you can either select the Specify Activity Name or Specify Activity Execution ID option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Specify Activity Name**: Select this option to specify the activity name and type the activity name in the adjacent field. Alternatively, you can click the expression editor icon and select a variable assigned to the required activity in the expression editor window.

  Note: If the parent activity is associated with multiple versions of the child workflows that have the same name, all the activities added to these child workflows are listed in this field.

- **Specify Instance ID**: Select this option to specify the activity instance ID and type the ID in the adjacent field. Alternatively, you can click the expression editor icon and select the instance ID assigned to the required activity in the expression editor window.
- **Error If Activity Instance Not Found**: This property helps you to set the output of this activity as "Error encountered" or "Successful" when the workflow to be ended has already been executed or is yet to be executed.

Property Type: Mandatory (This property must be set to end the workflow.)

Yes: If this value is assigned, then the End Activity Execution activity gives an output as "Error encountered" whenever the instance of the activity that is to be executed is not found. In other words, if the activity execution is not yet started or is already completed, then this error is encountered.

No: If this value is assigned, then the End Activity Execution activity will give the output as "Successful".

Specify Value: This option can be used during the runtime execution of the workflow. You can assign the 'Yes' or 'No' values to a variable which will be used to know the value of this property. Alternatively, you can click the expression editor icon and select a variable assigned to the workflow in the expression editor window.

Loop Execution Step Entry Activity: You can use this property to specify the name of the Loop Execution Step Entry activity to which the activity to be ended is associated. This property is required if an instance level of association between the End Activity Execution activity and the activity to be ended is required. If this is not set, the End Activity Execution activity ends all the instances of the activity that is to be ended. This property is mainly used when the For Loop and For-Each-Loop activities are used in a workflow.

Specify Activity Name: Select this option and type the Loop Execution Step Entry activity name in the adjacent field. Alternatively, you can click the expression editor icon and select a variable referring to the required activity in the expression editor window.

End Child Workflow Execution Activity

Activity Description

This activity enables the user to end a child workflow from the parent workflow.

Application Scenario

Consider a scenario where a child workflow is associated to the parent workflow. The parent workflow has an "Approval Activity". On the other hand, the child workflow contains a "Wait for Event" activity which is set to execute when a list item is deleted. This scenario, however, requires that if the user executes the approval activity in the parent workflow, the workflow need not wait for the completion of the child workflow which has the wait for event activity waiting to be executed. In such a scenario, the End Child Workflow Execution activity can be used to end the child workflow. This activity helps the user ensure that the parent workflow completes without any child workflow waiting for execution.

Refer to User Scenarios given in the *User Guide* for a detailed understanding of this activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

The End Child Workflow Execution activity has to be configured by specifying appropriate values for different properties in the **Activity Properties** pane. This pane can be accessed by clicking on the Activity Properties tool in the **Tool Bar** or by selecting the Activity Properties option from the context menu of End Child Workflow Execution activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property is to be used to specify a name for the activity.

Property Type: Mandatory (This property must be set.)

Description - This property can be used to specify a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in this group have been described below. You can use these properties to configure the functionality of this activity.

Workflow Details - This property is used to specify the workflow which needs to be ended. The user has an option to either set this property or the Specify workflow execution ID property.

Property Type: Mandatory (This property must be set to end the workflow.)

See End Child Workflow: Workflow Details for a more detailed understanding of its properties.

Specify workflow execution ID - This property is used to specify the workflow execution ID of the child workflow. The user has an option to either set this property or the Workflow Details property.

Property Type: Mandatory (This property must be set to end the workflow.)

Steps to set Specify Workflow Execution ID

1. To specify this property, type the child workflow execution ID in the field. Alternatively, click the expression editor icon and select a variable assigned to the child workflow in the expression editor window.

Specify Loop Execution Step Entry - This property is used to specify the name of the loop execution step entry activity used in the parent workflow.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set Select loop execution step entry

1. To specify this property, type the name of the loop execution step entry activity used in the parent workflow. Alternatively, click the expression editor icon and select a variable assigned to the child workflow in the expression editor window.

Error If Activity Instance Not Found: This property helps you to set the output of this activity as "Error encountered" or "Successful" when the workflow to be ended has already been executed or is yet to be executed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Mandatory (This property must be set to end the workflow.)

Yes: If this value is assigned, then the End Child Workflow Execution activity gives an output as "Error encountered" whenever the instance of the child workflow that is to be executed is not found. In other words, if the child workflow execution is not yet started or is already completed, then this error is encountered.

No: If this value is assigned, then the End Child Workflow Execution activity will give the output as "Successful".

Specify Value: This option can be used during the runtime execution of the workflow. You can assign the 'Yes' or 'No' values to a variable which will be used to know the value of this property. Alternatively, you can click the expression editor icon and select a variable assigned to the workflow in the expression editor window.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Looping Behavior

The property in the **Looping Behavior** group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop-down field.

Property Type: Optional (This property needs to be set only if necessary.)

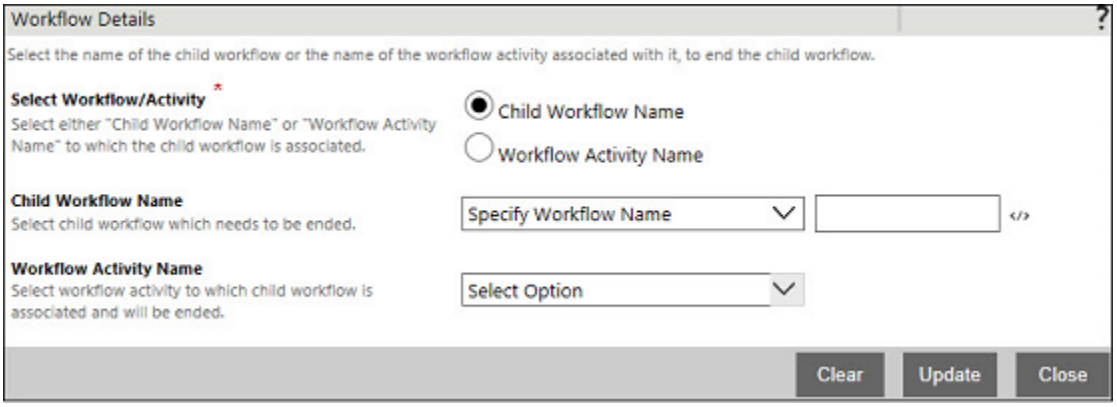
End Child Workflow: Workflow Details

Workflow Details property window

This topic gives you a detailed understanding of the **Workflow Details** properties. The Workflow Details properties are used to specify

- the child workflow to be ended.
- the workflow activity to which the child workflow that is to be ended is associated.

To open the **Workflow Details** window, click the Workflow Details edit button.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

This window has the following properties:

Select Workflow/Activity: Select the name of the child workflow that is to be ended or the name of the activity that is associated to the child workflow. Select the **Child Workflow Name** or **Workflow Activity Name** radio button.

Property Type: Mandatory (This property must be set.)

- **Child Workflow Name**: Select the name of the child workflow that needs to be ended from the drop-down list. This option is enabled when you have selected the **Child Workflow Name** radio button in the **Select Workflow/Activity** property.

  Note: You can view the child workflow in the drop-down list only if you have configured the child workflow in the Workflow activity.
- **Specify Workflow Name**: Select this option and type the name of the child workflow in the adjacent field. Alternatively, you can click the expression editor icon and select a variable assigned to the child workflow in the expression editor window.
- **Workflow Activity Name**: Select the name of the workflow activity used in the parent workflow, to which the child workflow is associated. The workflow activity name can be selected from the drop-down list. This option is enabled when you have selected the Workflow Activity Name radio button in the **Select Workflow/Activity** property.
- **Specify Activity Name**: Select this option and type the name of the workflow activity to which the child workflow is associated, in the adjacent field. Alternatively, you can click the expression editor icon and select a variable assigned to the workflow activity in the expression editor window.

BlockingOutput Activity

Activity Description

This activity facilitates synchronous execution of workflow activities other than human activities with an active form.

This is typically used in forms, whose fields need to be updated based on the output of multiple workflow activities of the same workflow, executed one after the other, without closing the form.

Activity Properties

The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

Specify the Output to be returned to the application that calls this workflow when this activity is encountered.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down. The looping option for this activity is Completed.

Property Type: Optional (This property needs to be set only if necessary.)

Note:
- When you use the BlockingOutput activity for returning the blocking output value, ensure that all the activities being executed before this activity, do not have sleeping behavior. For example: There should not be any activities, such as human or scheduler activities, which go to the sleep mode. However, Workflow activity with some specific settings can be included before this activity. For more information about the Workflow activity, see Workflow Activity.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Database Activity

Activity Description

The database activity enables execution of queries, stored procedures, and functions from the workflow. The database activity establishes connection with the database server and retrieves or updates data in the database. The activity gets the schema of the database table to an XML variable according to the query, stored procedures, or functions passed.

Application Scenario

Consider a scenario where a manager wants to send appraisal forms to all the employees. The manager can add the database activity with a query to select the email address column from the Employee Database. The Output XML will have the schema based on the query and this XML variable can used to assign employees to whom the appraisal forms are to be sent. When the workflow is initiated, the query in the database activity will be executed and all the employees will receive the appraisal form in their Inbox.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

The Database activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Database activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Note: Every time you input a value for a property, ensure to save the data. Click the **Save** button on the **Activity Properties** window. You need to save the property so that the subsequent properties in the activity can inherit the required information.

Name & Description

You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

You can use these properties to configure the functionality of the activity.

Database Connection

Database connection property is used to get the connection to the Database Server. You can use this property to select the Database connection (which has been configured in **Enterprise Console > Settings > Database Connections**).

Property Type: Mandatory (This property must be set if the activity has to establish connection with the database server.)

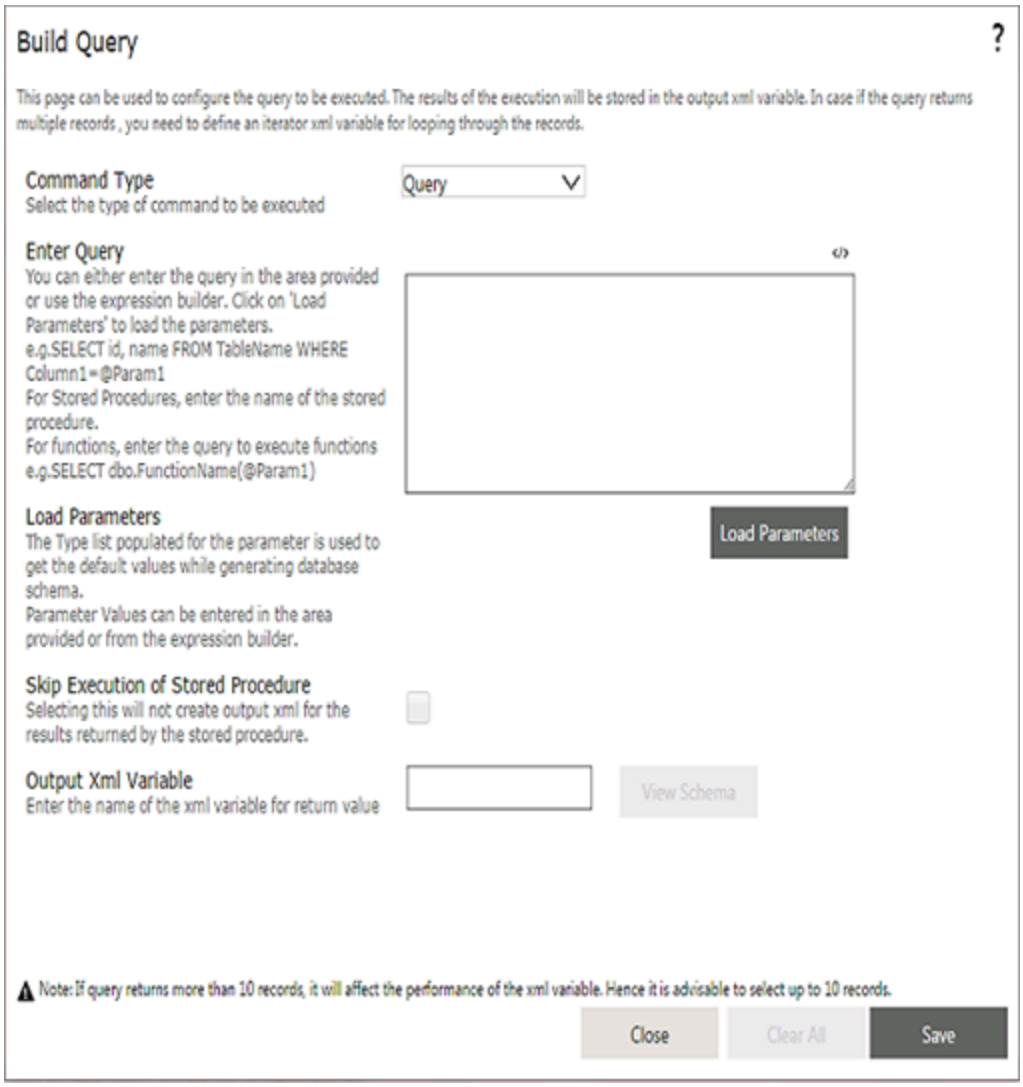
Query Builder

To enter the query, click the **Query Builder** property to open the property window.

In this window, set the following properties:

1. Command Type
2. Query
3. Output XML variable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Command Type: The command types that can be selected

- Query
- Stored Procedure
- Functions

Enter Query: As the Command Type is selected, you can enter a query, Stored Procedure, or Function.

Note: The **Expression Editor <>** icon is displayed only for the **Command Type "Query ",** as the expressions set cannot be evaluated in design time for stored procedures and functions.

Query: Query can be entered in the text box or from the Expression Builder.

For example, the query can be "**Select * from EmployeeTable**" and if the query is to be passed with parameters then the query will be **"Select * from EmployeeTable where EmployeeID=@Id**". Here @Id is the parameter. When you click on "Load Parameters", it will load the parameters within the query. In the Value field you can pass the value for the parameter.

Query with 'Order By', 'Group By' keywords are supported.

The following Parameter values can be saved with empty values:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

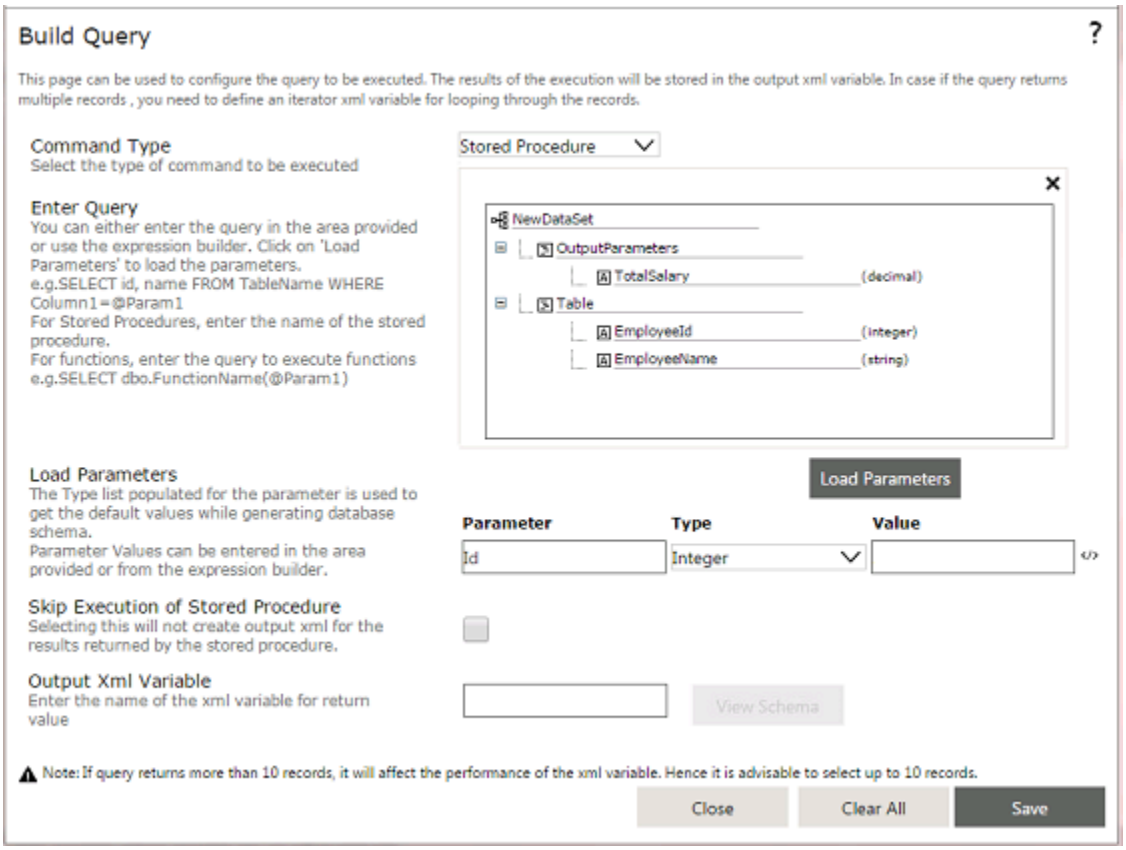
- Name: Displays the name of the parameter.
- Type: Select the Parameter type from the drop-down list. This type helps you to execute the query with internally defined default values only while generating the database schema. The appropriate Type needs to be selected for the Query and Functions. For stored procedures the Types will be automatically selected.
- Value: Type the value of the parameter.

Parameter helps you retrieving values from the database where the value is null or empty.

If the query is to retrieve values from the database we have to specify the Output XML variable name. For more information, see the table *Database Activity*.

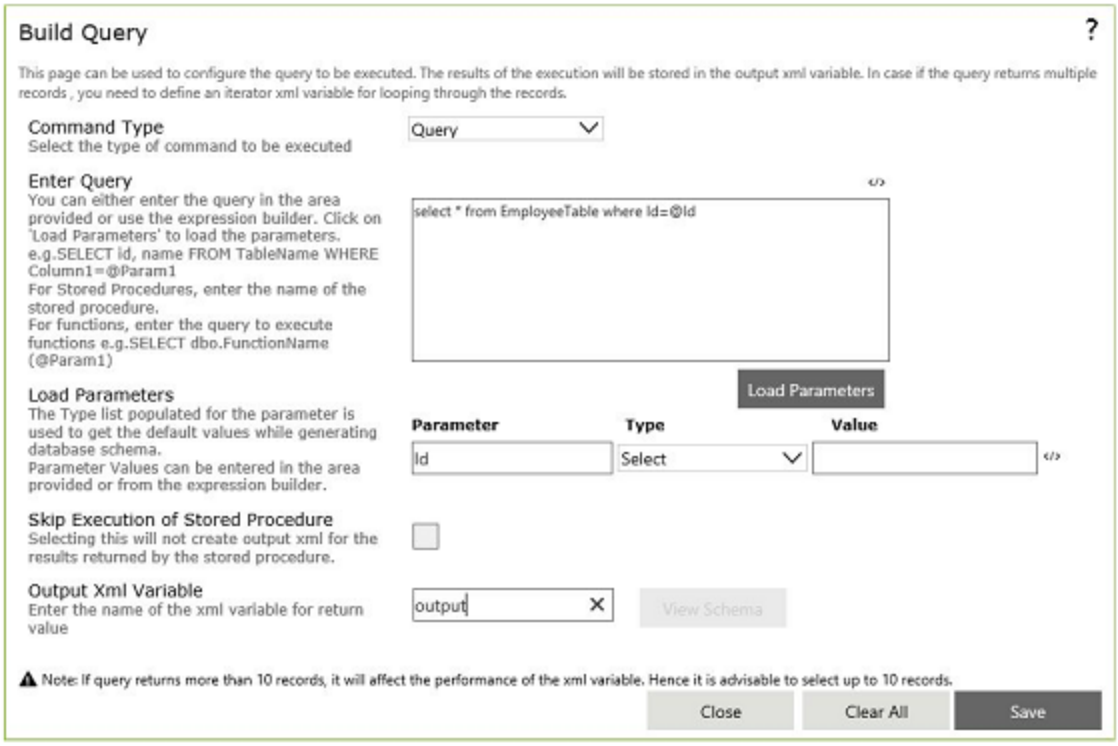
Multiple queries are not supported. If multiple queries are used, XML variable will carry the result of the last query written.

Click on **'View Schema'** to view the schema of the output XML variable "outXML"



XML variable used in one activity cannot be reused within other database activities.

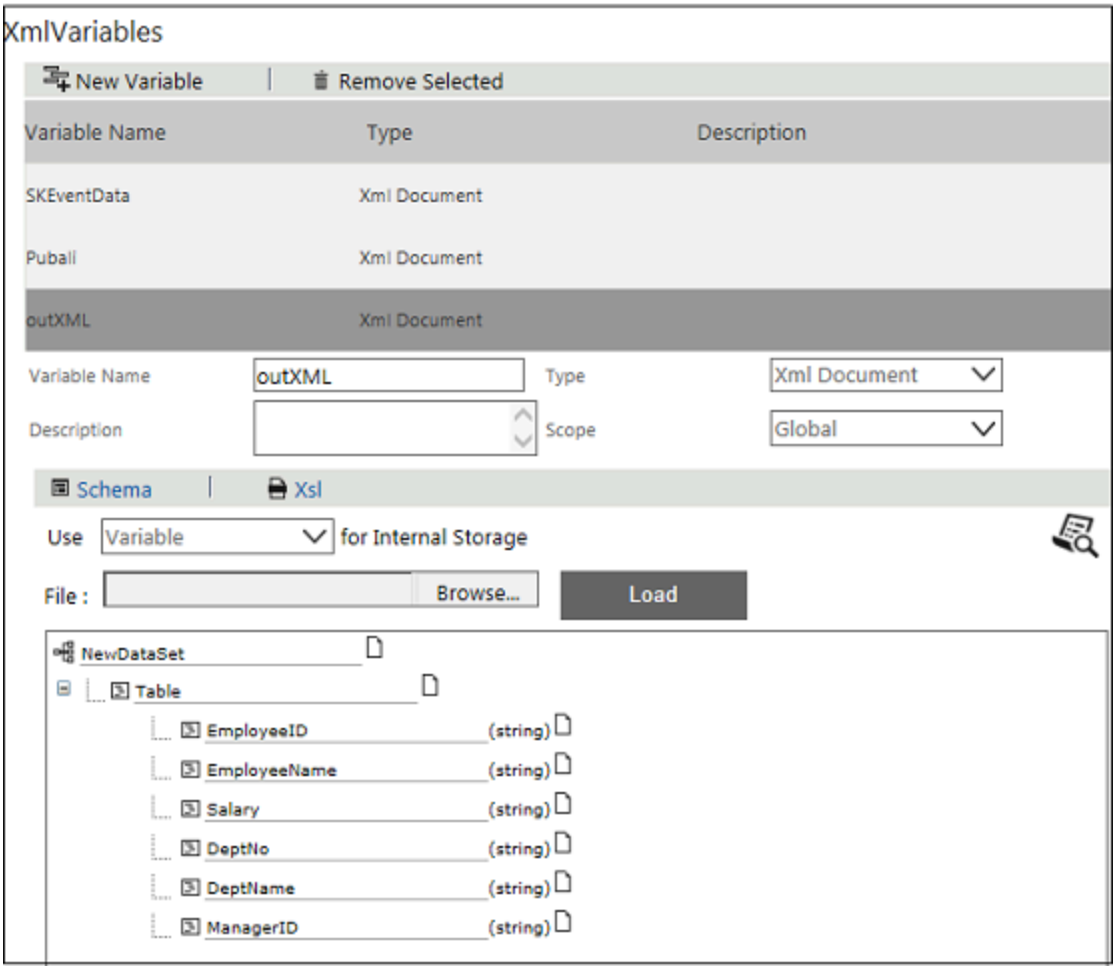
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



If the output schema containing return values have been modified after the property is saved, open the Query Builder property and click Save. The Schema will get updated with the latest values. Click **View Schema** to see the updated changes.

This XML Variable "outXML" will be added in the **Start > XML Variables** as shown below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Parameter values cannot update data types like 'image', 'blob' and 'unique identifier' from XML variables or variables.

Stored Procedure

Stored procedures can be executed by simply specifying the name of the Stored Procedure (provided there is a stored procedure created in your database server). On the click of "Load Parameters" it will load the parameters present in the Stored Procedure. If you want to retrieve values on executing the Stored Procedure you can specify the name of the XML variable. If return values from the stored procedure are stored in output parameters, the activity will load the XML schema with the output parameters.

To discover the output XML schema, system internally executes stored procedure with parameter values as empty string or null. If a stored procedure has validations on parameter or data integrity constraints, you cannot configure a stored procedure.

If a stored procedure returns the values for return Result Set and output parameters, then both the values will be available in the XML variable.

Note: If there are multiple Return Result Sets, then only the first return Result Set will be considered.

- **Skip Execution of Stored Procedure**: To avoid the execution of stored procedure to get the output xml schema, select this check box so that you can configure stored procedure with parameter validation or data integrity constraints.

Note:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- It is recommended to use proper parameter values/variables that has appropriate values at runtime to execute the stored procedure.
- To execute a stored procedure, which expects default values for execution, you can create a wrapper stored procedure with the same parameter. You can then call the actual stored procedure with default values passed to the stored procedure if the parameter values are empty or null.

Functions

Functions are executed in the same way as executed in the database server, for e.g. "select dbo.FunctionName". If the function contains parameters you should specify the exact parameters. For e.g. "select dbo.FunctionName(@param1, @param2)" (provided there is the function created in your database server). If you want to retrieve values on executing the function you can specify the name of the XML variable. If return values from the function are stored in output parameters, the activity will load the XML schema with the output parameters.

Note:
- Table valued functions are not supported.
- To execute a function in the Oracle environment, add **from dual** after the function name.
For example: select MYFUNCTION('SK-Fresh', 'FN') from dual

Output XML Variable for return value:

Output XML variable will have the schema of the table as per the query entered. If the query is "Select * from EmployeeData" then the XML variable will have all the columns as nodes in the schema which can be looped through and used in the subsequent activities. Output XML Variable name should be unique, it cannot be repeated.

Note: To create an XML Iterator of the output XML Variable the node selected should always be the "Table" node.

Capture runtime exceptions in a variable

You can create a variable with the name **^<ActivityName>Error** in the **Start** activity. The exception message will be available in the variable if an exception occurs.

For example: For an activity named as "DBAct1", you have to create a variable with name as **^DBAct1Error** in the **Start** activity.

For more information on creating variables, see Start - Variables.

Capture returned records count in a variable

You can create a variable with the name **"^"** + **CurrentAction.Name** + **"ResultRowsCount"** in the Start activity. The total number of records will be returned based on the query stored in the variable.

For example: For an activity named **DBActivity**, you have to create variable with name as **^DBActivityResultRowsCount** in the **Start** activity.

For more information on creating variables, see Start - Variables

Property Type: Mandatory (This property must be set if the activity is to retrieving any value from the database (i.e. if query is of "select" type, else this field will be optional because it will not load any schema for 'insert', 'update' or 'delete' type of query)

Retry Connections: This property allows setting the maximum retries if a Database goes Offline at run time. By default the value set will be 1.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if necessary.)

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down. The looping option for this activity is Completed.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Database Activity returns the following values:

Successful: Displayed when the query is executed.

Unsuccessful: Displayed when the query execution is unsuccessful.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details

The following table shows the SQL database types and their corresponding parameter types for a database activity:

SQL Database Type	Database Activity Parameter Type
bigint	Integer
binary	Byte
bit	Boolean
char	String
date	Date
datetime	DateTime
datetime2	DateTime
datetimeoffset	DateTime
decimal	Decimal
float	Double
int	Integer
money	Currency

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SQL Database Type	Database Activity Parameter Type
nchar	String
numeric	Decimal
nvarchar	String
real	Single
smalldatetime	DateTime
smallint	Integer
smallmoney	Currency
sql_variant	Object
time	Time
timestamp	Binary
tinyint	Byte
uniqueidentifier	Guid
varbinary	Binary
xml	Xml

Decision Activity

Activity Description

The Decision Activity can be used to insert a decision in the workflow. The output of the activity can be used to determine the path of workflow execution.

Activity Properties

The Decision activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking the Activity Properties tool in the tool bar or by selecting the appropriate option from the right click menu for the Decision activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The property in the Configuration group has been described below. You can use this property to specify the decisions.

Decisions - The Decisions property allows the user to create complex rules to control the execution path of the workflow. In contrast to the Rule action, which only allowed a True/False kind of decision, the Decision property can be used to rules with multiple outputs based on the condition specified.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Decisions property

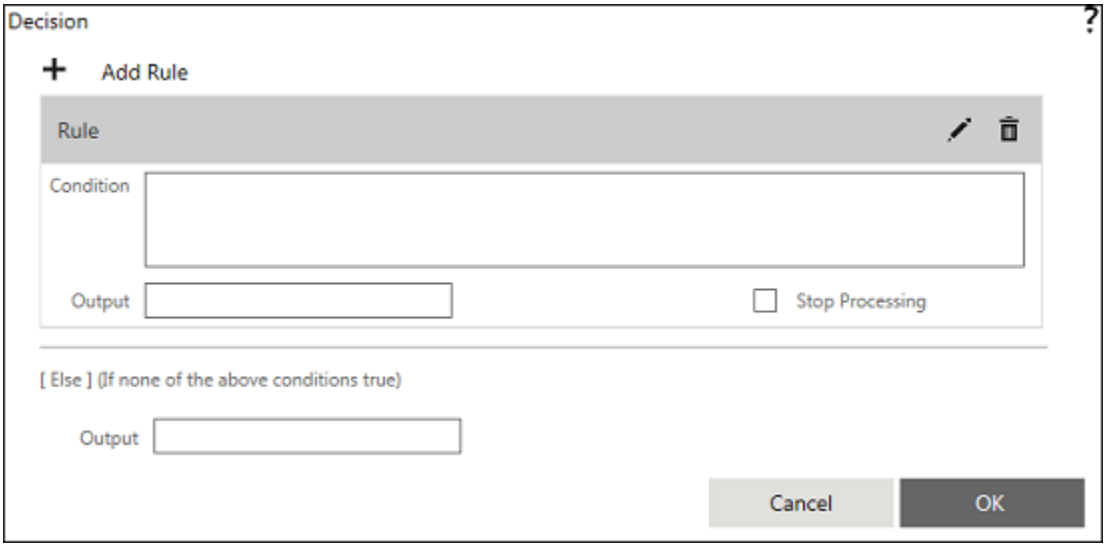
See Decision: Decisions for a detailed description of the Decision property window.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected. See Mapped Error Outputs for more details.

Decision: Decisions

You can create conditions for the decision action in this property window. To open this window, click the edit button for the 'Decisions' property. The following window shows the **Decision** property window:



This **Decision** window has the following controls to specify the condition for a rule:

- **Add Rule**: Click **Add Rule** to add a new set of the above controls for adding another rule.
- **Condition**: Specify the expression for the condition in this text area.

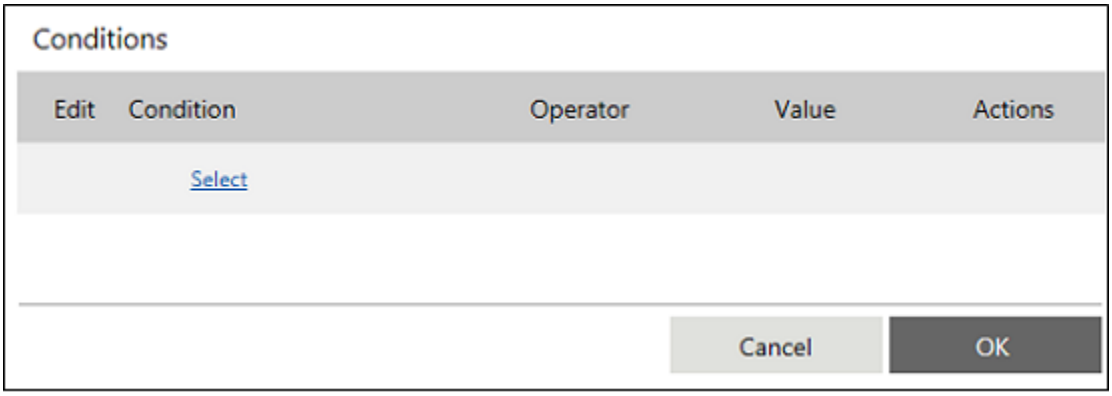
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Condition icon** : Click the condition icon to open the Condition window to build the condition expression.
- **Output**: Specify the output for the rule in this text box.
- **Stop Processing**: Select this check-box to enable the process to be stopped when this rule is "true".

The window also has the **Output** text box to specify the output for the exception, that is when none of the rule conditions are true, in the [**Else**] (**If none of the above conditions true**) field.

Condition window

You can build expressions for the rule condition in this window, which is opened when you click the Condition icon in the Decisions property window. This window has the Select link, which you can use to specify the left hand side of the expression. On mouse-over this link opens a combo, with a collection of objects that you can specify for the left hand side of the expression. The window has the **OK** button at the bottom to save the new expression. The **Cancel** option allows you to cancel the expression and return to the property window.



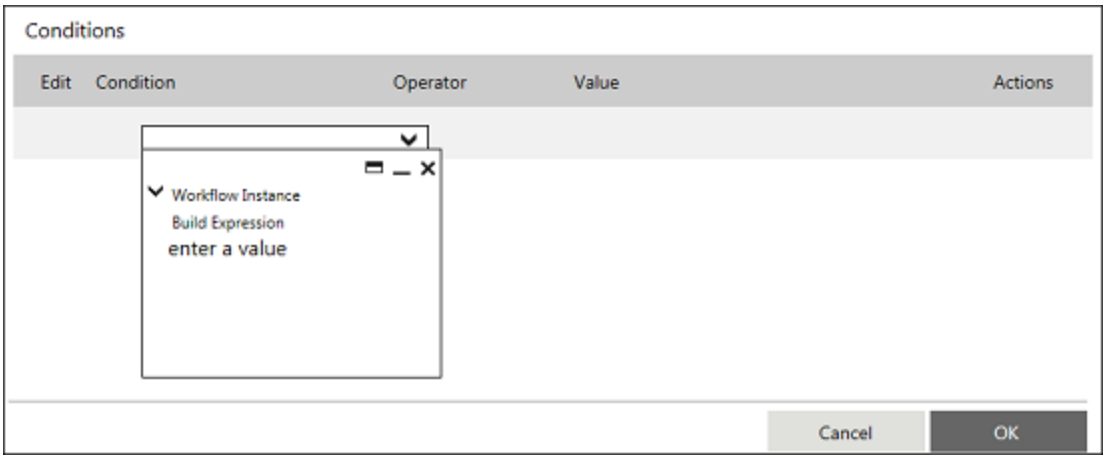
Building an Expression for the Decision Condition

To illustrate the process of selecting an object for the right hand side of the expression, let us consider an XML Variable with "Employee" as a node and "FirstName" as an element under it. Specify this as the object for the left hand side of the expression. From the mouse over drop down, scroll down till you see the XML Variables node. Expand this node and the "Employee" node under it. Select the "FirstName" option.

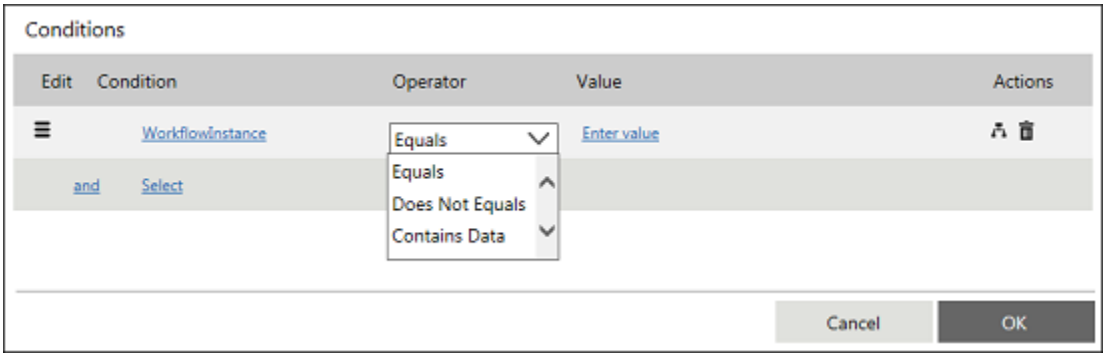
Note: The C# code is not directly supported in the Expression editor for the Decision activity. If you want to return any value from the C# code, refer to the Returning Value from the C# Code topic.

The following window shows how to set an object for the left hand side of the expression:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

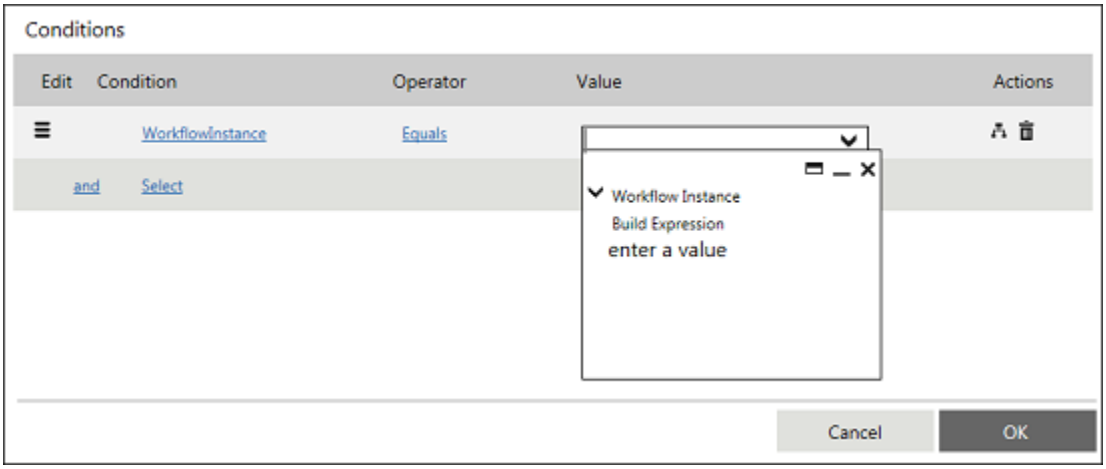


This will display the operator combo with operators corresponding to the data type of the selected object. In this case, the data type being string, the operators are string operators, shown in the image below. Select "Equals" as the operator.



Operators combo with string operators

The default control to specify the object for the right hand side of the expression, which is a text box, is displayed at the right hand side of the row, as shown in the image below. You can specify a matching value in this text box. As the object selected is First Name, specify "John" here.



Default control for specifying the right hand side of the expression

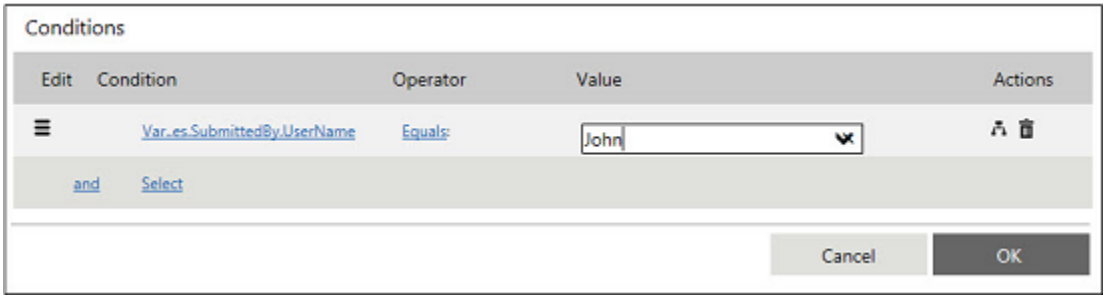
This will complete the first expression and the settings will be displayed as shown below. You can mouse over any link to edit the value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using the Expression mode icon

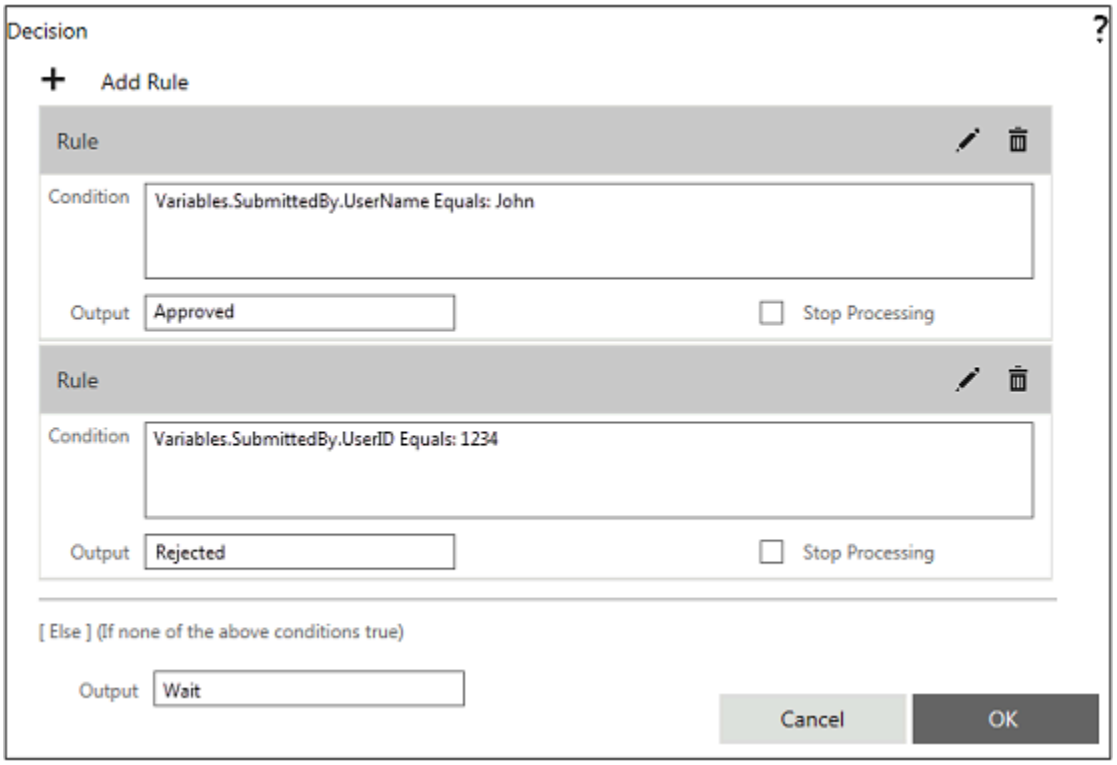
Click the Expression mode icon, , to change the default text box to the Expression mode. In the Expression mode, you can select an object from the combo for the right hand side of the expression.

Consider that the left hand side of the expression is VirtualUserID. You can assign the Variables.SubmittedBy.UserId as the value for the right hand side.



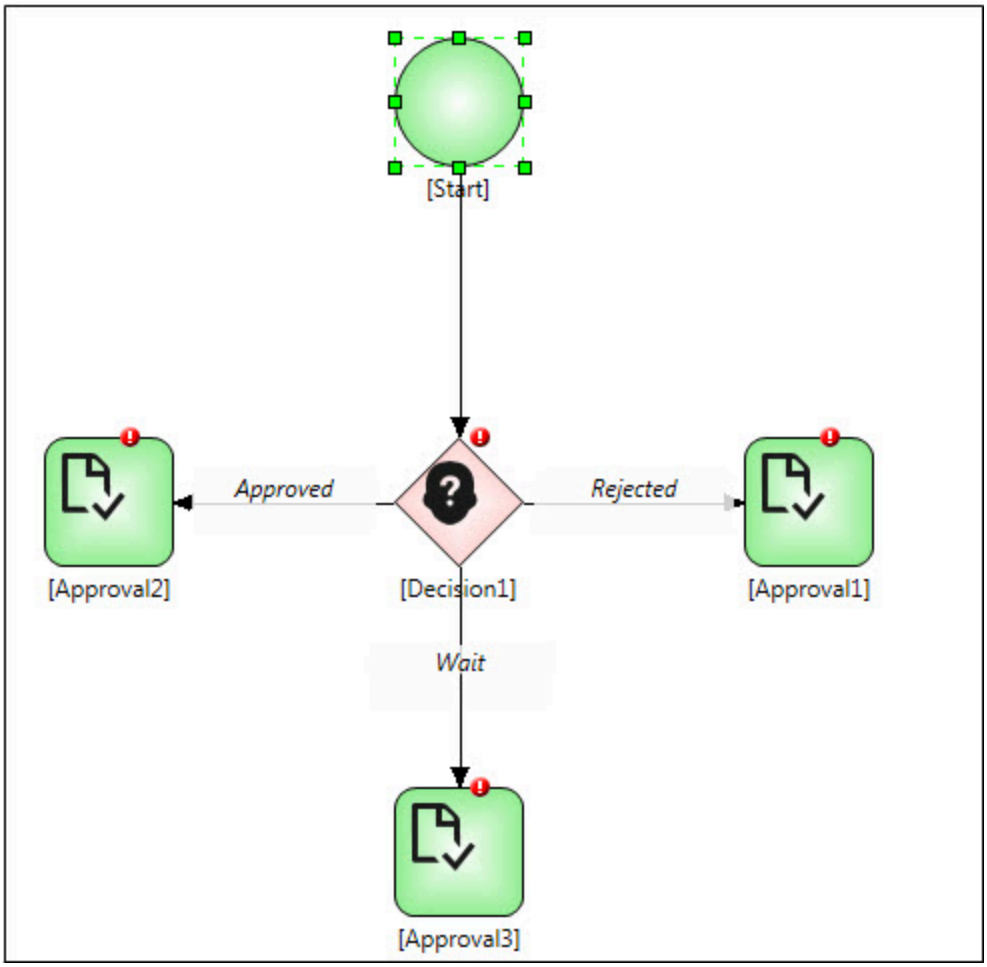
You can select "Build Expression" in the Expression mode combo to open the Expression Editor and build an expression instead of specifying an object on either side of the expression. You can select "Enter a value" in the right hand side combo to return to the default mode, that is, the text box.

Click **OK** to save the expression and return to the **Decision** property window. The following image shows the window with Decision conditions added:



The Decision activity would be typically used in a workflow as shown below, with each decision output determining a different path for workflow execution.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Decision activity in a typical workflow

Returning Value from the C# Code

You must follow the following steps to return any value from the C# code:

1. Declare a variable in the start activity.
2. Initialize the variable in Start activity or using update variable using the C# code.
3. This variable can be used in the Decision activity.

Note: In Decision Activity, when you specify a value for comparing numeric, date, time, or date, time type of variables, you must specify the value in Invariant format.

ForEach-Loop Activity

Activity Description

The ForEach Loop Activity is used to loop an activity through a collection within an XMLIterator Variable. It allows repeated execution of an activity or group of activities. It must always be used along with the Next Loop Activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

The properties of the ForEach Loop activity can be configured in the Activity Properties window for that activity.

To access Activity Properties, do one of the following:

- Click **Activity Properties** in the **Process Designer Tool Bar**
- Right-click and select **Activity Properties**

Activity Properties for this activity are organized under:

Name & Description

Specify the name and description for the activity in this group.

Name - Enter the name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - Enter a brief description for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The Configuration group has a single property for selection of the XMLIterator Variable to be looped through.

XMLIterator - Select the XML Iterator variable to be looped through. All XMLIterator variables as declared in Start activity are available for selection.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Execution - Select **Serial** if the activity has to be looped through each item in a collection in sequence. An activity will be created for an item only after completion of the activity for the preceding item. Select **Parallel** to simultaneously create activities involving each item. This field defaults to **Serial**. Also see Example of Serial Execution.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down. The looping option for this activity is Completed.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Exception Behavior

These options of Wonderware Skelta BPM 2009 have now been integrated with Raise Error on Unhandled Linked Output in the latest version of the product.

Property Type: By default is set as **NO**

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Activity outputs:

The ForEach Loop Activity returns the following values -

Step: Displayed to indicate the loop step

Loop End: Displayed when the end of the loop is reached

Example of XMLIterator Variable in ForEach Loop

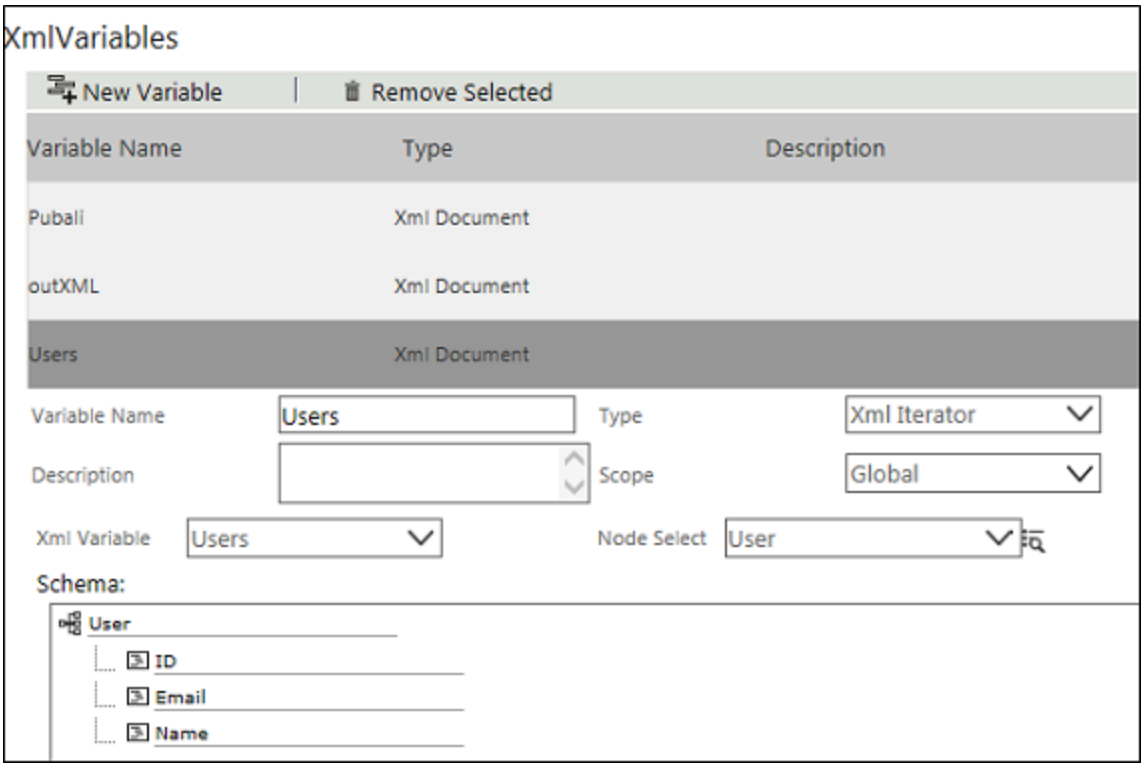
In this example, the XMLIterator Variable is used to get approval from all managers in the Managers.XML file containing the list of all users who are managers.

Managers.XML file

```
<Managers>
 <Users>
<User id=1>
 <Name>John</name>
 </User>
<User id=2>
 <Name>Jim</name>
</User>
<User id=3>
 <Name>Victor</name>
</User>
<User id=4>
 <Name>Andy</name>
</User>
 </Users>
</ Managers >
```

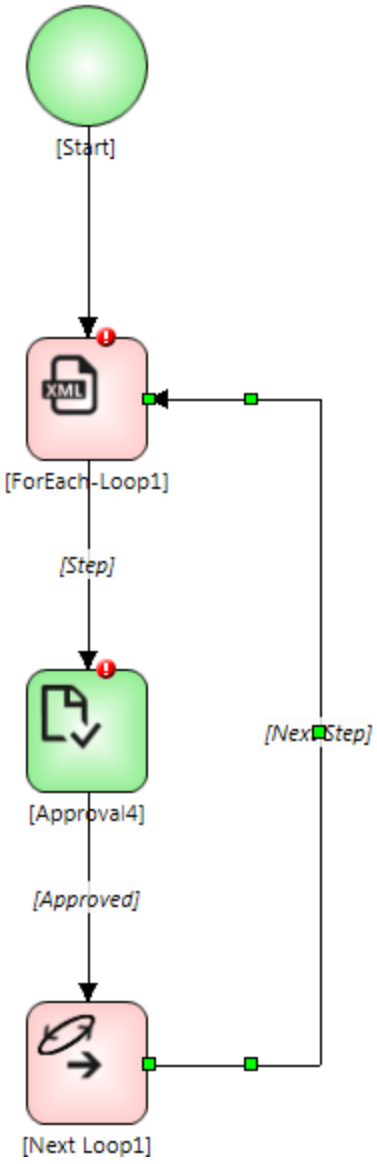
Step 1: XMLIterator Variable is defined for users. The node Users holds the collection of managers.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

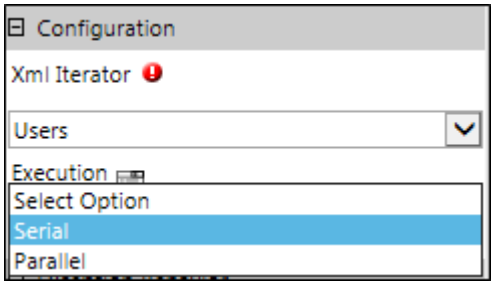


Step 2: The Approval activity is set within a ForEach Loop activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

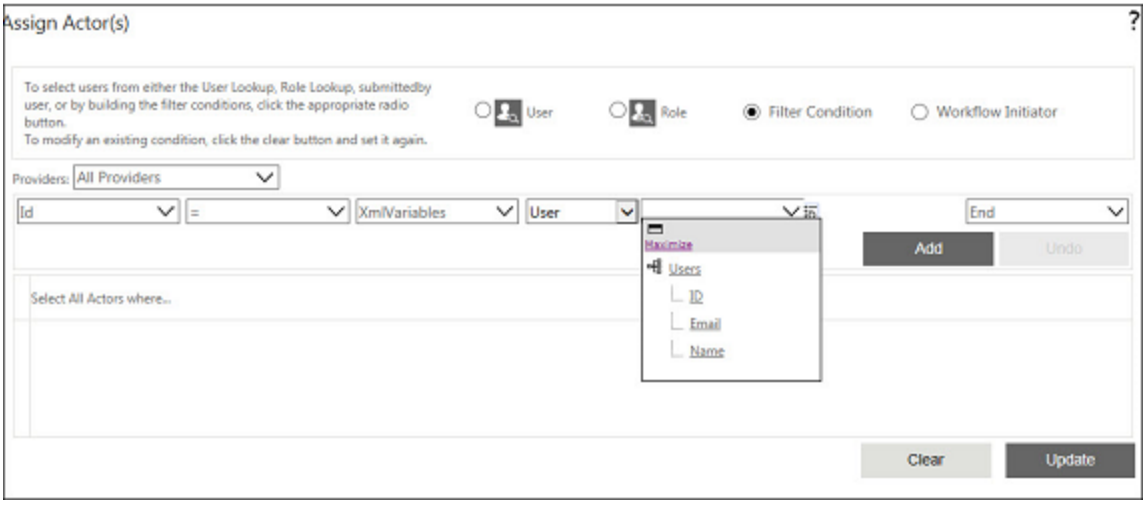


Step 3: The XMLIterator Variable Users is defined in XMLIterator field of ForEach Loop properties. The Execution type is set to Parallel as the approval is to be processed simultaneously by all approvers.



Step 4: The XMLIterator Variable Users is specified in the condition for selecting actors for the approval.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Actor Selection for Approval Activity

Output: When the above workflow is invoked, Workflow Engine resolves "XMLVariables.user.users.ID". It gets the ID of each manager and assigns the task to that manager.

Capture looping count in a variable

You can create a variable with the name **"^" + CurrentAction.Name + "LoopCount"** in the **Start** activity. The total number of times the ForEach-Loop is executed will be returned in the variable.

For example: For an activity named **LoopActivity**, you have to create variable with name as **^LoopActivityLoopCount** in the **Start** activity.

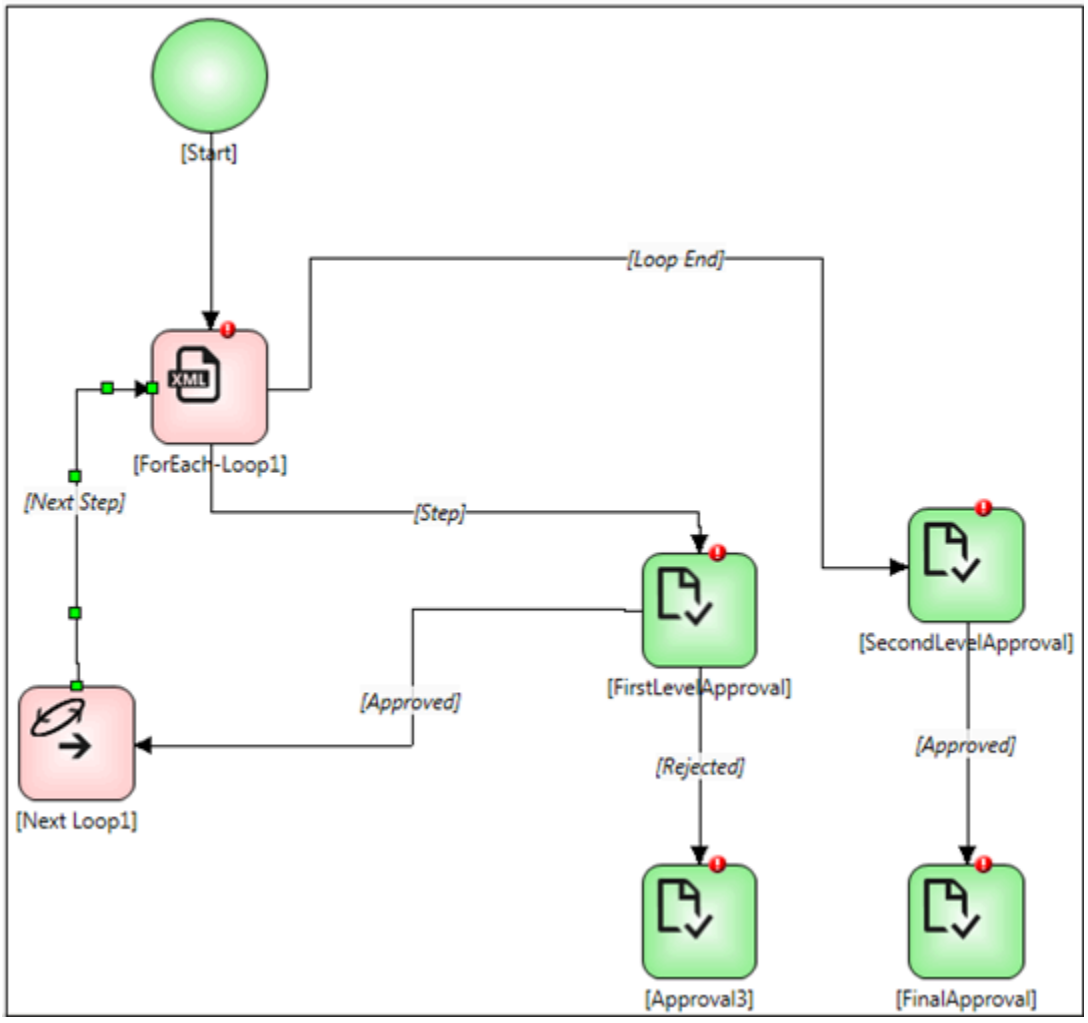
For more information on creating variables, see Start - Variables

Example of Serial Execution

To expedite processing of purchase orders, every line item in an order is forwarded to a different user for first level verification of item availability, price et cetera. The complete order can be processed only if all line items are approved. The line items are forwarded for approval sequentially. If any item is not approved, then the document is forwarded to the manager for approval. This process is depicted in the following figure.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Process Example for Serial Execution



For-Loop Activity

Action Description

The For-Loop Activity can be used to start a loop within a workflow. It is useful for repeated execution of an activity or group of activities. It must always be used along with the Next Loop Activity.

Activity Properties

The For-Loop activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the For-Loop activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Loop Variable - This property allows the user to specify the variable to use as the counter for the loop. It can be selected from the drop down for this property. These variables should have been declared in the Start activity.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Loop Count - This property allows the user to specify the loop count, which will determine the number of iterations of the loop.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the Loop Count property

1. To specify the loop count, click on the button for the 'Loop Count' property to open the property window.
2. In this window, enter the count value directly in the text editor.
3. If you want to assign the loop count dynamically during run-time, you can do so using the **Field** controls. For example, you can add a variable or content pointing to the loop count from the two drop downs for **Field** and click on the **Append** button. This will add the placeholder for the variable or content in the text editor. At run-time this placeholder will be replaced by the corresponding value, that is the loop count.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After specifying the count, click the Update button.
7. Click the Close button to close the window without saving the parameter.

See For - Loop Count for a detailed description of the Loop Count property window.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity outputs:

The For-Loop Activity returns the following values -

Step: Displayed to indicate the loop step

Loop End: Displayed when the end of the loop is reached

Variable Not Found: Displayed when the specified variable is not found

Invalid Count: Displayed when there is an invalid loop count

Exception Behavior

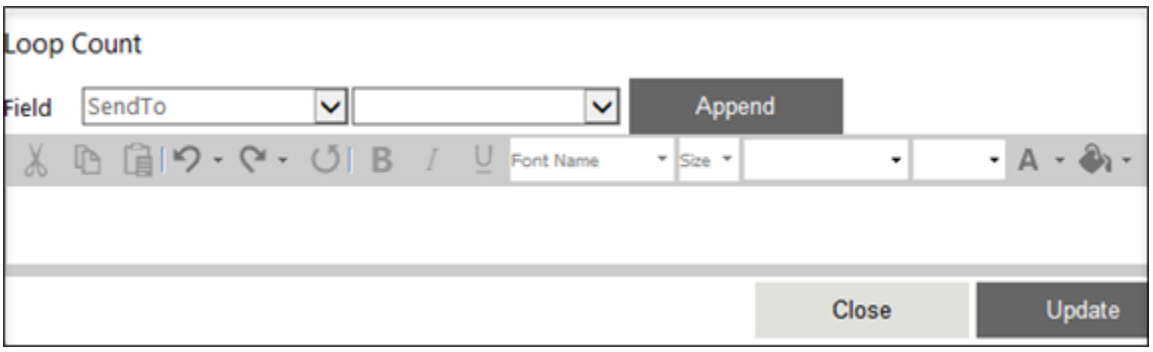
Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

For - Loop Count

Loop Count Property window

You can set the loop count that will be used to determine the iterations of the loop in this property window. To open this window, click the button for the 'Loop Count' property.



The window has a text editor to enter the loop count. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value or XML Variable here.
- The Value text box/drop down - You can specify the actual value of the loop count. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the loop count in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the loop count.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the loop count details. It has the following text tools –

- Cut (Ctrl+X) - to cut text

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the loop count
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the loop count
- **Close** - You can click this button to close this window

Logger Activity

Activity description

The Logger Activity can be used to log messages like information, warnings, and errors to the logger console from the workflow.

Activity Properties

The Logger activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Logger activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

LogType - This property allows the user to select the type of message to be logged. The drop down lists the following options: Information, Warning, and Error.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

LogMessage - This property allows the user to specify the message to be logged.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Log Message property

See Log - Log Message for a detailed description of the Log Message property window.

Activity outputs:

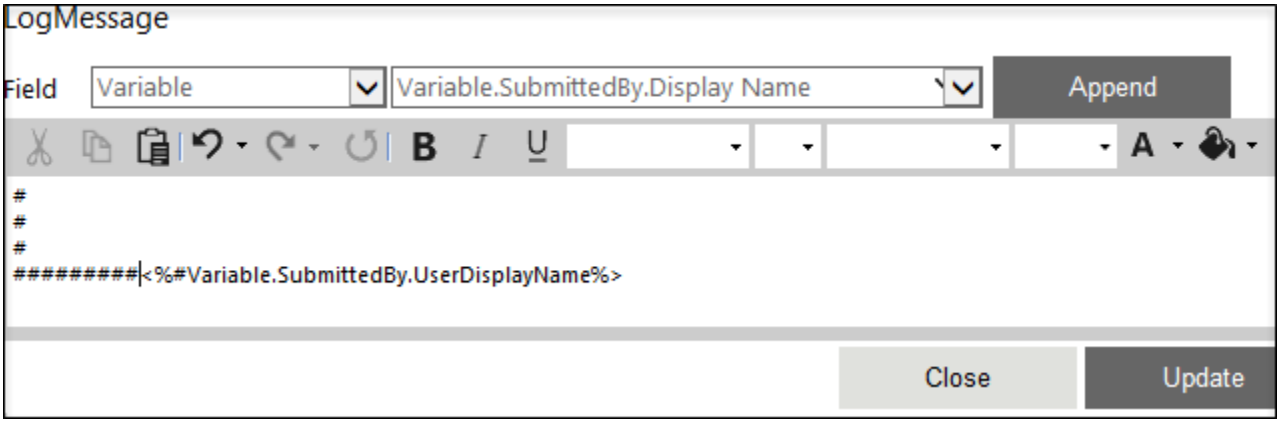
The Logger Activity does not return any value.

Example for a Logger:

Step 1: In the Activity Properties of the Logger1, select LogType as Information, and click Save button of Activity Properties.

Step 2: Log Message - Select the field as **Variable** in the first drop-down. In the second drop-down, select the Variable.SubmittedBy.DisplayName, and then click **Append**.

Step 3: The entry would be as this: <%#Variable.SubmittedBy.DisplayName%>. The '#' Symbol is just for quick identification of data in the Logger.



Note: Before publishing the workflow, start the AVEVA Work Tasks Logger Console.

Output

In the logger, you can see the contents of the variable Variable.SubmittedBy.UserName with the '#' symbols.

In the below figure you can see the name Administrator at the end of the # symbols.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Exception Behavior

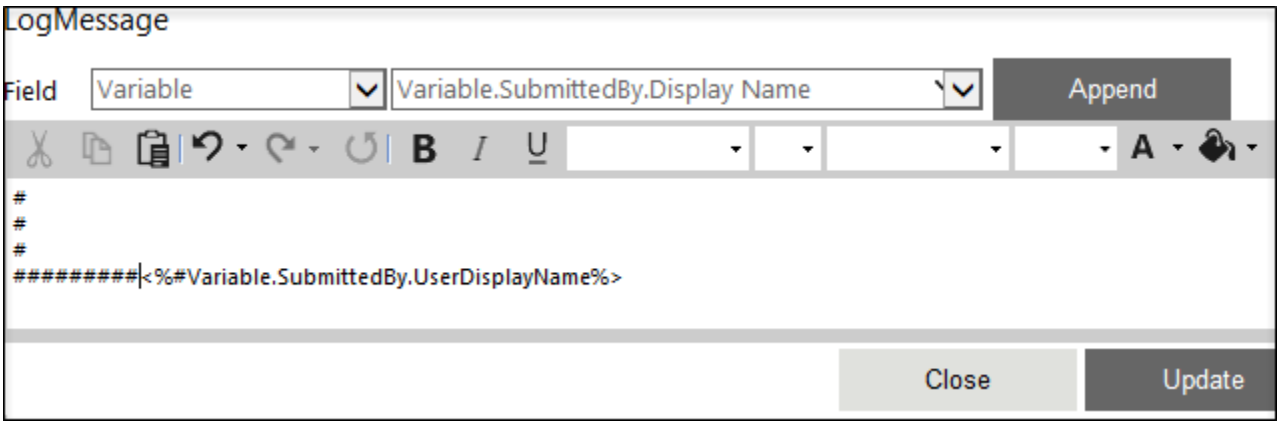
Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Log - Log Message

Log Message Property window

You can specify the Log Message, which should be logged to the Log Console from the workflow, in this property window. To open this window, click the button for the 'Log Message' property.



The window has a text editor to enter the Log Message. You can do this either by entering a discrete value directly in the text editor or by using the Field controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value or XML Variable here.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The Value text box/drop down - You can specify the actual value of the Log Message. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Log Message in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Log Message.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Log Message details. It has the following text tools –

- **Cut (Ctrl+X)** - to cut text
- **Copy (Ctrl+C)** - to copy text
- **Paste (Ctrl+V)** - to paste text
- **Undo (Ctrl+Z)** - to undo the last action
- **Redo (Ctrl+Y)** - to undo the last action
- **Repeat Last Command (F4)** - to repeat the last command that was executed

It also has the following formatting & other tools –

- **Bold (Ctrl+B)** - to make text bold
- **Italic (Ctrl+I)** - to italicize text
- **Underline (Ctrl+U)** - to underline text
- **Font Name** - to select the font for the Log Message
- **Size** - to specify the font size
- **Paragraph Style** - to select a paragraph style
- **Zoom** - to zoom in and zoom out the view
- **Foreground Color** - to set text color
- **Background Color** - to set the background color
- **Print** - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the Log Message
- **Close** - You can click this button to close this window

Loop Execution Step Entry Activity

Activity Description

The user can use this activity if an instance level association between the End Activity Execution activity and the activity to be ended is required. If this activity is not used, the End Activity Execution activity ends all the instances of the activity that is to be ended. This activity has to be used when the For Loop and For-Each-Loop activities are used in a workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Scenario

Consider a scenario where a ForEach-Loop activity is linked to a Wait-For-Event activity and an Approval activity. There is a parallel execution of each instance of Wait-For-Event activity that is associated to an instance of Approval activity. With the ForEach-Loop activity running in parallel execution mode, multiple instances of Wait-For-Event activity and Approval activity are created. In such a scenario, the Loop Execution Step Entry activity can be used to establish an instance level association between the End Activity Execution activity and the activity to be ended, thereby controlling the instances to be ended. Loop Execution Step Entry activity has to be placed in such a way that it is the parent activity for Wait-for-event and approval activity.

See the User Scenarios of End Activity Execution activity and End Child Workflow Execution activity for a detailed understanding of this activity.

Activity Properties

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property is to be used to specify a name for the activity.

Property Type: Mandatory (This property must be set.)

Description - This property can be used to specify a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop-down field.

Property Type: Optional (This property needs to be set only if necessary.)

Next Loop Activity

Activity Description

The Next Loop activity can be used as the terminal activity in a loop initiated using a For-Loop Activity. It will check if the count value has been reached. If the value has not been reached it will increment the 'Loop Variable' set in the For-Loop Activity and repeat the execution of the activities in the loop.

Activity Properties

The Next Loop activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

tool bar or by selecting the appropriate option from the right click menu for the Next Loop activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop-down.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Next Loop activity returns the following value -

Next Step: Displayed to indicate the next step in the loop

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Resource Variable Activity

Activity description

The Resource Variable Activity can be used to identify an actor to perform a workflow activity, through a variable. It sets the resource properties to a variable. If the variable type is set as 'resource' then the resource property is set to the variable. If variable is set as 'string' then the actor IDs are assigned to variable. The activity gives the output "Activity Executed" which is the output condition to be set to link this activity to the next. This activity can also be used to assign the next activity based on actor workload to the person with the lightest load, in order to properly balance workload.

Activity Properties

The Resource Variable activity has to be configured by specifying appropriate values for the different properties

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Resource Variable activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Assign Resource - This property allows the user to select the variable to which the resource properties will be set. The variables will have to be created in the properties for the Start activity. The variables can be one of the three types - String, Array or Resource. String variables take one UserID at a time, while Arrays take in a number of UserIDs pointing to different resources. Resource variables will take the complete resource properties.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Enable Workload Balancing - This property allows the user to set the Enable Workload Balancing property by selecting from the drop down. Setting this property value to Yes, enables workload balancing. This ensures that the task assignment will be done taking workload balancing into consideration.

Property Type: Optional (This property needs to be set only if required.)

MaxUsers Count - This property allows the user to specify the maximum number of actors who will be considered for task assignment. This property is related to the previous property. If workload balancing is set to Yes, then this property needs to be set to a value greater than 1, in order for workload balancing to be performed.

Property Type: Optional (This property needs to be set only if required.)

Actors

The property in the Actors group has been described below. You can use this property to specify the actors who will work on the activity.

Actors - This property specifies the actor(s) who are to be assigned to the variable using the ResourceVariable activity.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Actors property

1. To specify the actors, click on the button for the 'Actors' property to open the property window.
2. In this window, build an expression to select the actors using the actors properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.

3. Add the condition to the expression by clicking **Add**. This will display the expression in the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Resource Variable - Actors for a detailed description of the Actors property window.

See the Using XML Variables section in the About the XML Variable User Interface topic for details on how to use the XMLVariables to build the expression.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Resource Variable Activity returns the following values -

Not Enough Resources: Displayed when filter Condition did not yield enough resources

Activity Executed: Displayed when the Resource variable is set successfully

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

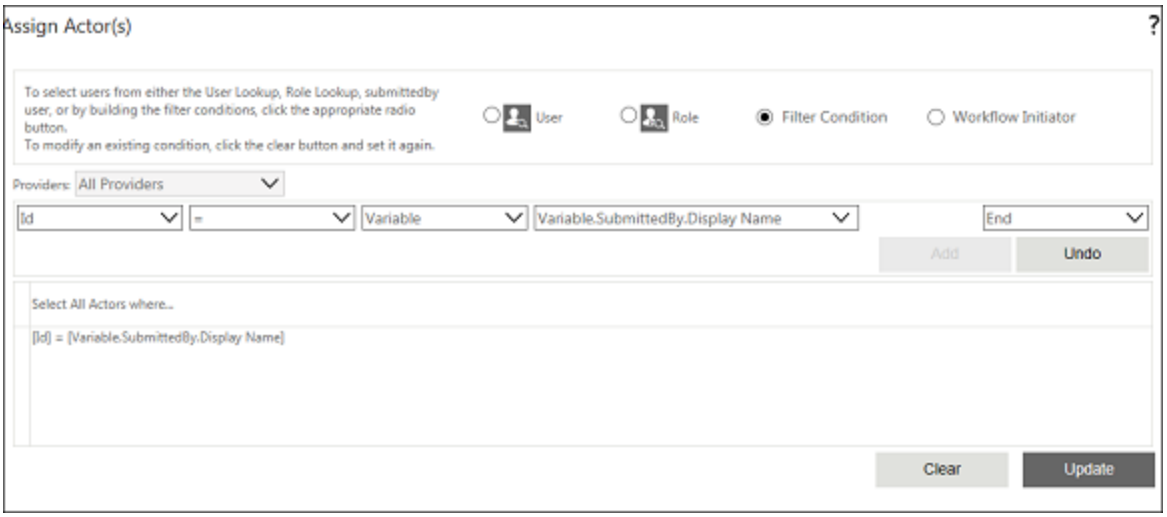
See Mapped Error Outputs for more details.

Resource Variable - Actors

Actors Property window

You can set the actors to whom the Resource Variable work item is to be sent in this property window. To open this window, click the button for the 'Actors' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

Different Methods of Assigning an Actor

- Assigning an Actor by building an expression:

Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

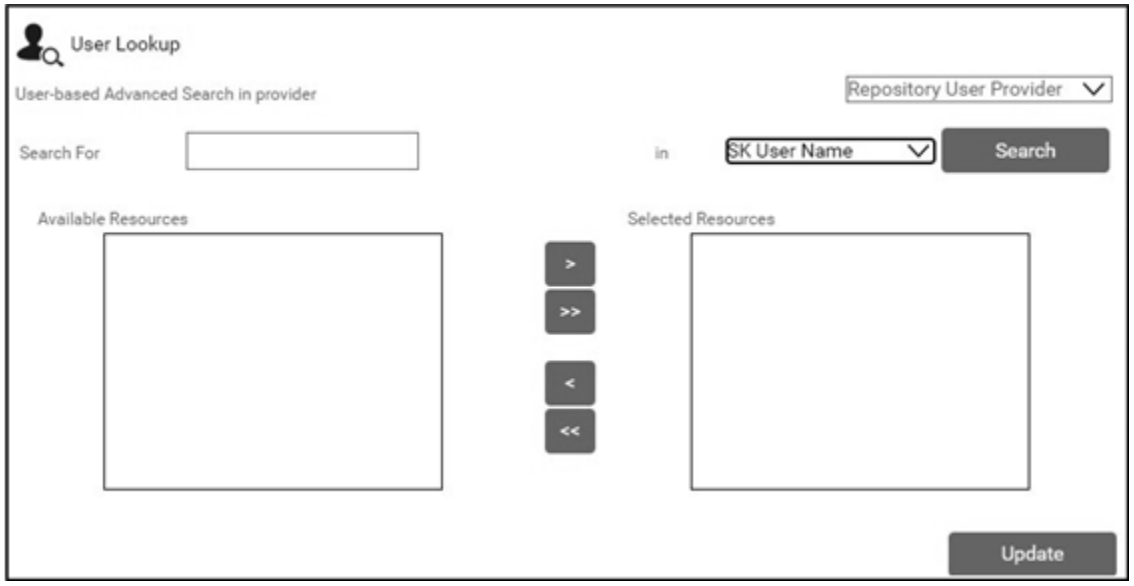
At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

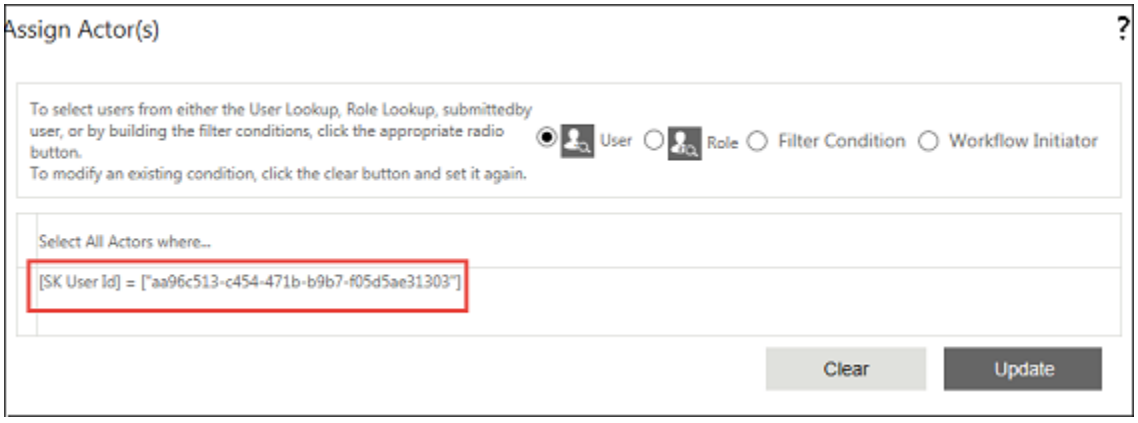
- Assigning an Actor through User Lookup

Through the **User Lookup** you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.
2. The **User Lookup** dialog is displayed, search and select the required actor.



3. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window. In the **Assign Actor** window, you will see that an expression is displayed, see the following image:



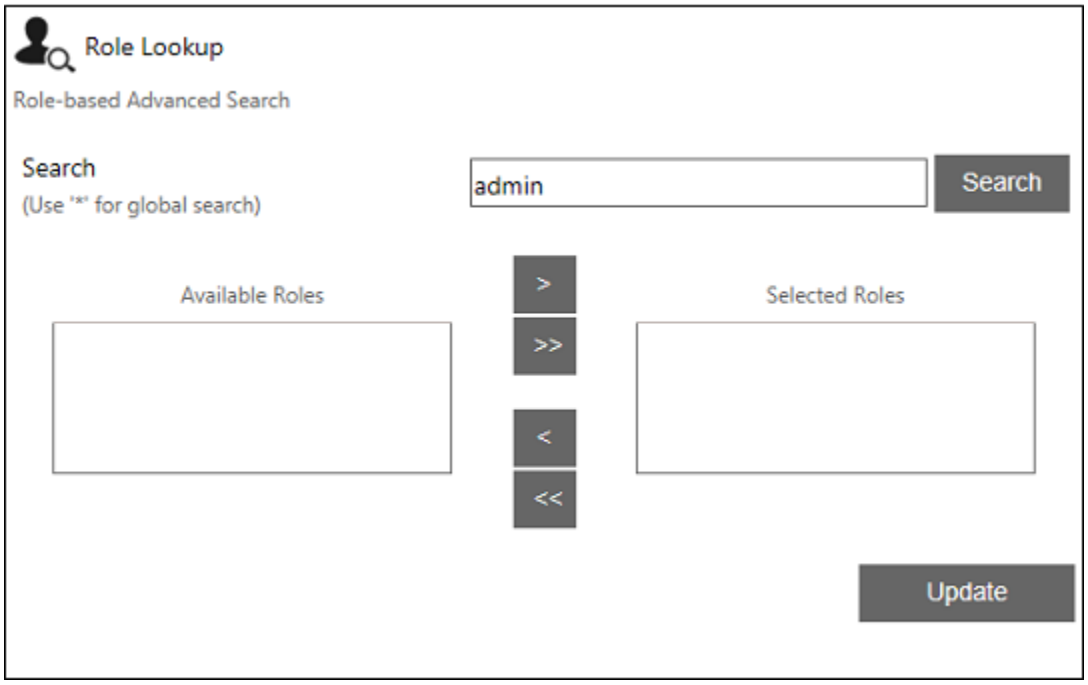
4. Click **Update** in the Assign Actor(s) window to update the selected actor(s) in the **Assign Actor** property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

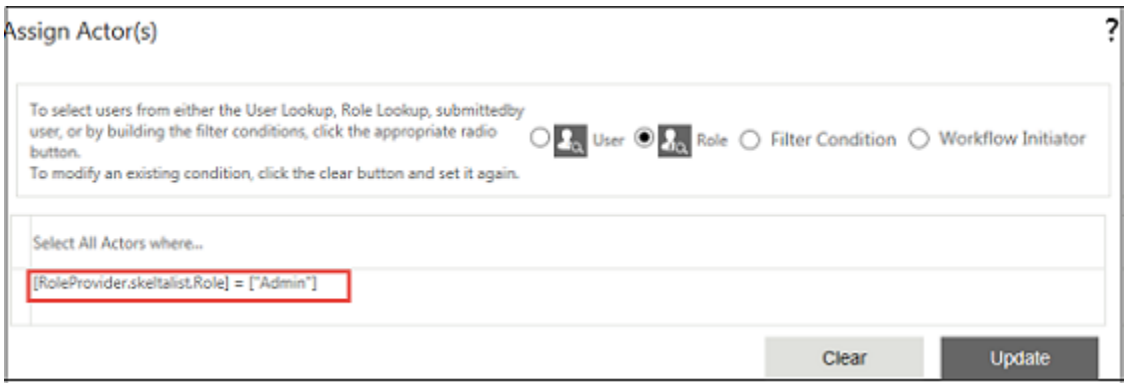
- Assigning an Actor through Role Lookup

Through the Role Lookup you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

1. Select the **Role Lookup** option.
   The **Role Lookup** dialog is displayed, search and select for the required role.



2. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, see the following image:
3.

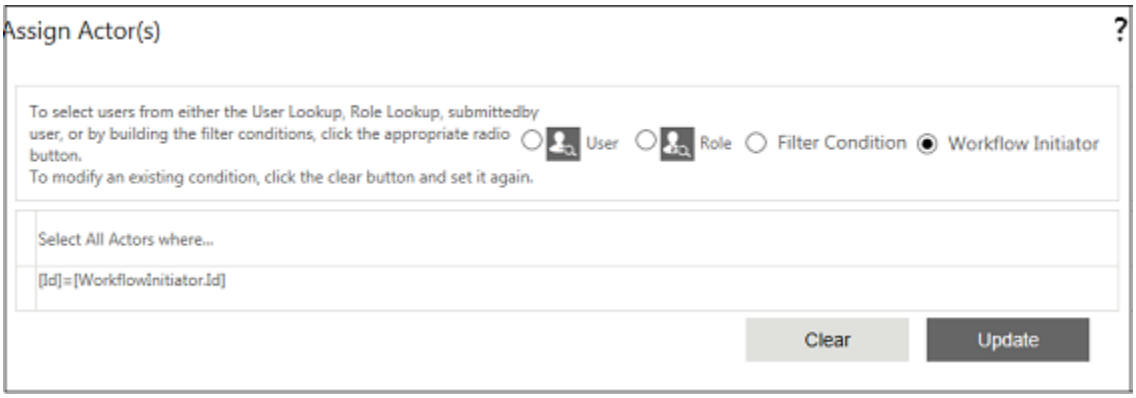


4. Click **Update** in the Assign Actor(s) window to update the Assign Actor property.

- Assigning an Actor through Workflow Initiator

Selecting the **Workflow** Initiator allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, see the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Rule Activity

Activity description

The Rule Activity allows complex rules or expressions to be built into the workflow. The expression will either result in 'true' or 'false'. The workflow engine evaluates the expression and assigns the result (true/false) to the output. The engine determines the path of the workflow from the output.

Activity Properties

The Rule activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the tool bar or by selecting the appropriate option from the right click menu for the Rule activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The property in the Configuration group has been described below. You can use this property to specify the decisions.

Condition - This property allows the user to frame complex expressions for the workflow execution. The expression results either in 'true' or 'false'. The workflow engine, after evaluating the expression and getting the result, decides on the path for the activity, as per the process design.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the Condition property

1. To specify the condition for the Rule activity, click the button for the 'Condition' property to open the property window.
2. In this window, build a complex expression for the rule using the fields provided. Select a type of value (Variable, Content, XMLVariables) from the first drop down list, then select the corresponding value from the second drop down. This forms the left-hand side of the rule condition expression. Next, select the operator for the condition from the third drop down. Finally, select the type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the respective drop downs to add the right-hand side of the rule condition expression. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression in the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Rule - Condition for a detailed description of the Condition property window.

See the Using XML Variables section in the About the XML Variable User Interface topic for details on how to use the XMLVariables to build the expression.

Activity outputs

The Rule Activity returns the following values -

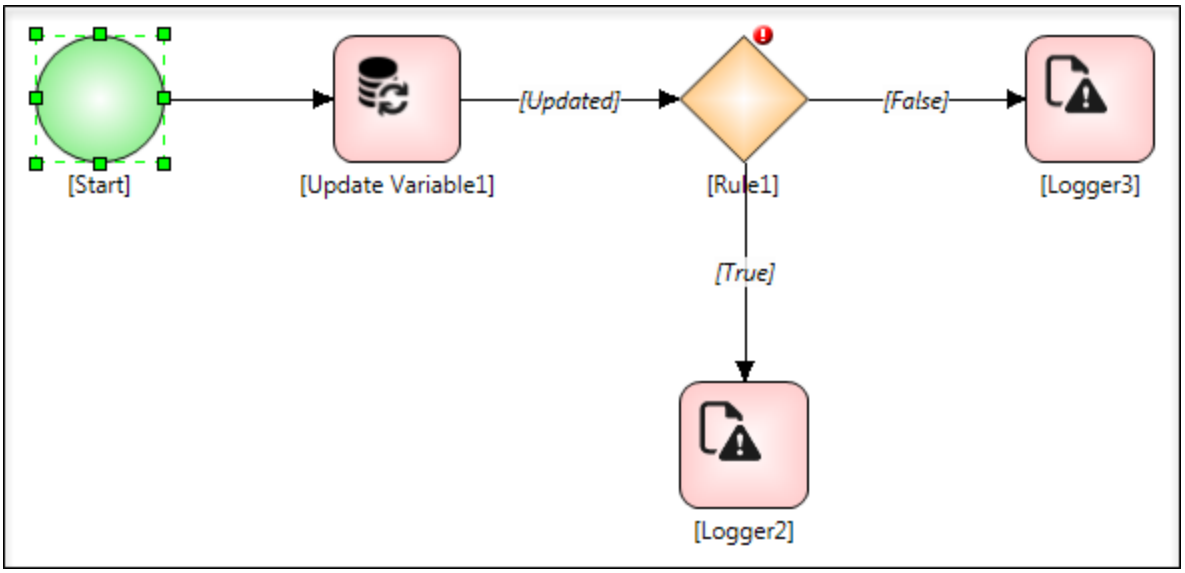
True: Displayed when the rule condition is true

False: Displayed when the rule condition is false

Example for a Rule Activity:

Step 1: Design a workflow as shown in the following figure.

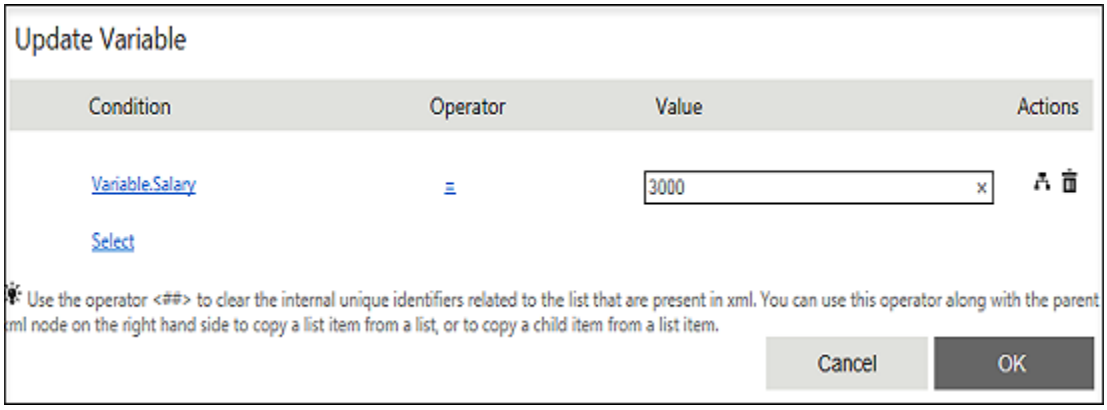
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Step 2: In the Start Activity declare a variable Salary.

Step 3: In the 'Set Variable' property of Update variable activity, perform the following actions:

- In the first drop-down, select Variable.Salary.
- Type a value. For Example, 3000.
- Click **OK**.



Step 4: In the Condition property of Rule Activity, perform the following actions:

- In the first drop-down, select Variable.Salary.
- Second drop-down, select '>' .
- Third drop-down, select 'Value'.
- In the adjacent text box, enter a variable. For Example, 2000.
- In the last drop-down, select AND.
- Click **Add**.
- Once again, in the first drop-down, select Variable.Salary.
- Second drop-down, select '<'.
- Third drop-down, select 'Value'.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- In the adjacent text box, enter a variable. For Example, 6000.
- In the last drop-down, select END.
- Click **Add**.
- Click **Update**. The entry would look as the following:

[Variable.Salary]>2000 And Variable.Salary<6000

Step 5: On the link connecting the right logger, select False and display an error message.

Step 6: On the link connecting the left logger, select True and display the value of the variable.Salary.

Output: The Variable.Salary will be updated to 3000, and in the rule condition the condition will be checked. The rule activity will return true and execute the logger at the bottom of the rule activity displaying the value of the variable Salary in the Logger Console.

Exception Behavior

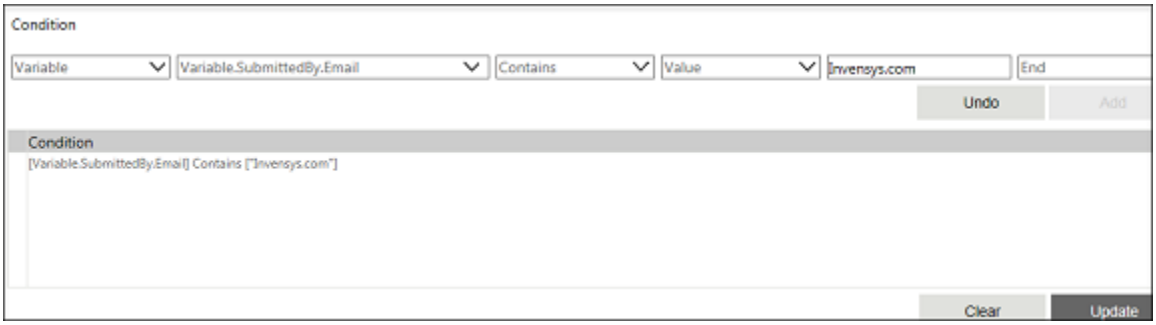
Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Rule - Condition

Condition Property window

You can set the condition corresponding to the Rule activity in this property window. To open this window, click the button for the **'Condition'** property.



The window has a set of drop downs at the top to select the different parameters required to build an expression for the condition. You can also find related buttons next to the drop downs.

- The first Value type drop down - You can select the type of value that will form the left-hand side the rule condition expression, in this drop down. You can select "Variable", "Content", or "XMLVariables" from this

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

drop down.

- The first Value drop down - You can specify the actual value corresponding to the value type selected in the preceding drop down. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Operator drop down - You can select the operator for the expression here. E.g., "=", "<=", "IN", "Contains".
- The second Value type drop down - You can select the type of value to be matched with the value specified in the left-hand side of the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/ drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down. If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

Workflow Activity

Activity description

In AVEVA Work Tasks, a workflow can be called from another workflow provided both workflows belong to the same repository. In this case, the calling workflow is termed as **Parent Workflow** and the workflow to be called is termed as **Child Workflow**.

The Workflow Activity enables the execution of a second (child) workflow from a (parent) workflow. The Workflow Activity can be used to call a child workflow from a parent workflow. The **Variables for the child workflow** property and **XMLVariables for the child workflow** property enable parent workflow to pass variable and XML variable information to the child workflow respectively.

Activity Properties

The Workflow Activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the workflow activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Repository - This property specifies the application or repository name that has the workflow to be run. The child workflow should belong to the same repository of the parent workflow. The repository property of a child workflow is exposed for backward compatibility.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Workflow - This property specifies the workflow to execute. This is an editable combo, which drops down a list with the names of all available workflows for the specified application. You can select one of the workflows as the child workflow. Alternatively, you can enter a name in the field to create a new child workflow. Now when, you right-click the workflow activity icon and select the "Open Sub Process" option, a new Process Designer window will be opened with the Application name and Workflow name specified here. You can create the new sub-process in this window.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Version - This property specifies the version only if any specific version of the (child) workflow to be executed. Leave this as blank, if the latest published version of (child) workflow has to be executed.

Property Type: Optional (This property needs to be set only if required.)

Contents for ChildWorkflow - This property specifies the content to be passed on to the workflow. Build the content to be passed to the child workflow, if required.

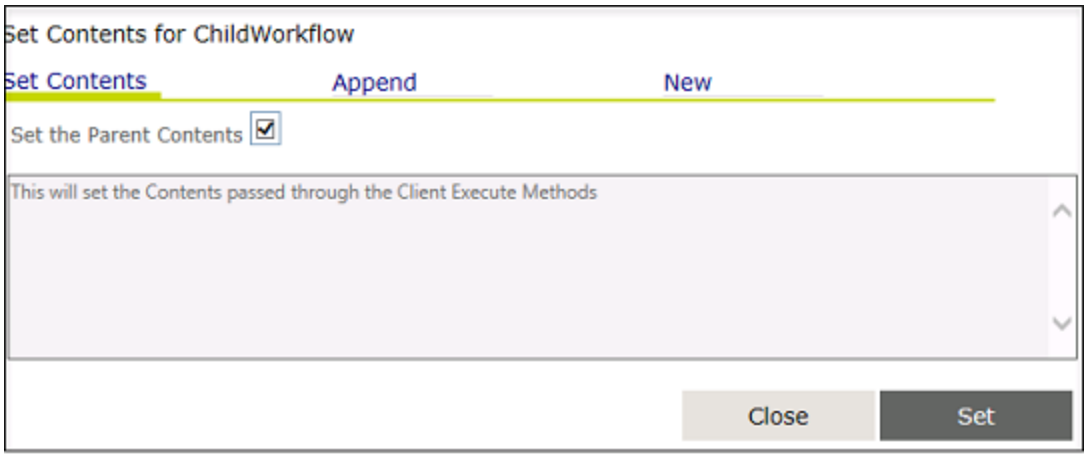
Property Type: Optional (This property needs to be set only if required.)Use one of the following options to specify the content to be passed:

- Set the contents to be passed as same as Parent Contents
- Append some more contents to the parent workflow contents
- New workflow contents

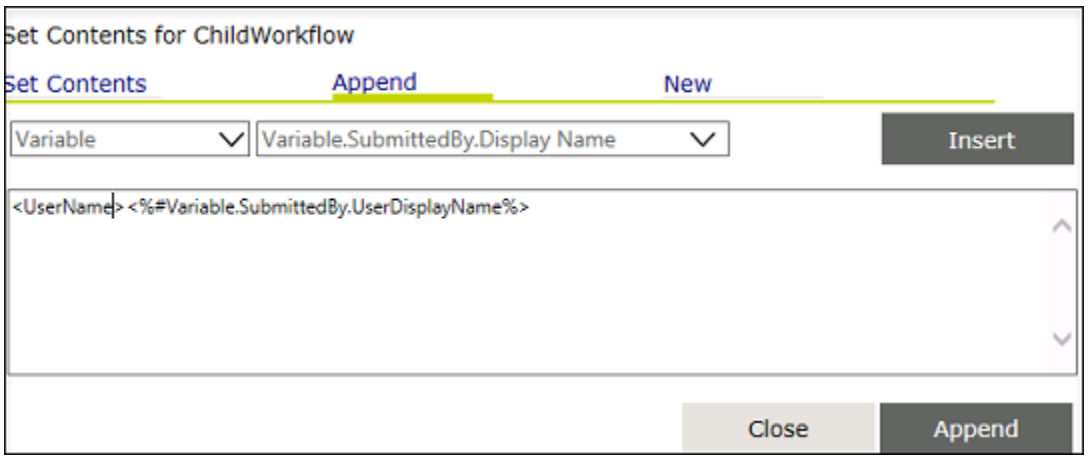
Steps to set the Contents for ChildWorkflow property

1. To specify the contents to be passed to the child workflow, click the button for the 'Contents for ChildWorkflow' property to open the property window.
2. In the Set Contents for ChildWorkflow dialog box, the Set Contents tab is selected by default and the corresponding window appears.
3. Click the Set the Parent Contents check box to pass the parent contents XML to the child workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

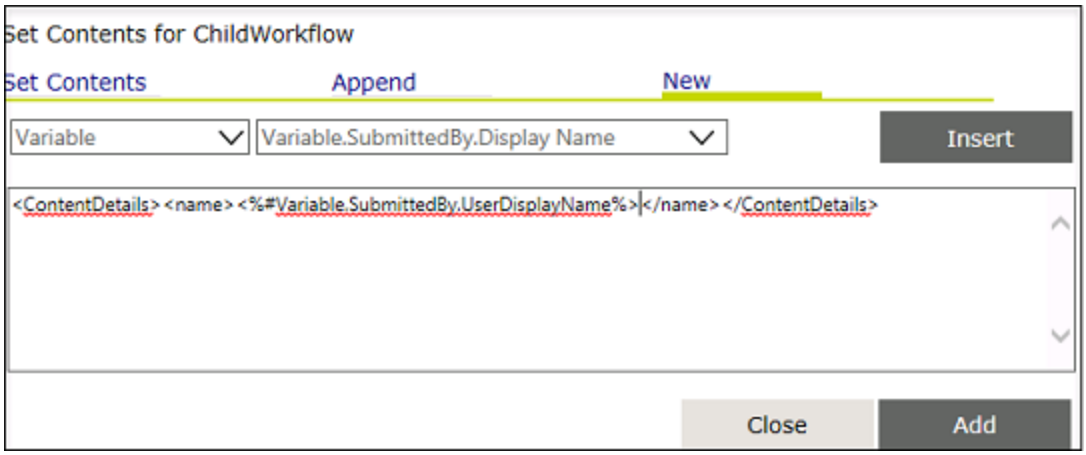


4. Click the Set button to set this property.
5. Alternatively, you can append XML content to the parent workflow contents and pass them to the child workflow. To do this, click Append to open the corresponding window.
6. Select a type of value (Variable or Content) and the corresponding value collection from the respective drop-down and then click Insert to add the selection to the text area.



7. Add more XML contents if necessary. After adding all the values, click **Append** to save the setting.
8. Third options to build complete XML content and can pass these contents to child workflow. To do this, click the New tab to open the corresponding window.
9. In this window, select a type of value (Variable or Content) and the corresponding value collection from the respective drop-down and then click Insert to add the selection to the text area.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



10. After specifying the new content, click **Add** to save the settings.

See Workflow - Contents for ChildWorkflow for a detailed description of the Contents for ChildWorkflow property window.

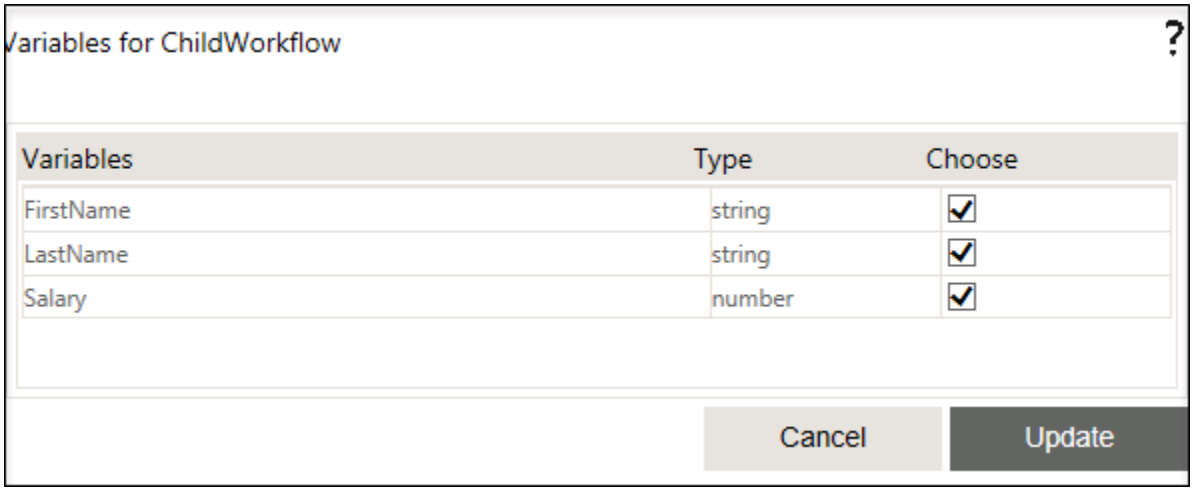
See the Using XML Variables section in the About the XML Variable User Interface topic for details on how to use the XMLVariables to build the expression.

Variables for ChildWorkflow - This property specifies the variables to be passed on to the workflow. This window shows the list of variables defined for the current (parent) workflow. Select the variables that need to be passed to the child workflow.

Property Type: Optional (This property needs to be set only if required.)

Steps to set the Variables for ChildWorkflow property

1. To specify the variables to be passed to the child workflow, click the button for the 'Variables for ChildWorkflow' property to open the property window.
2. In the Variables for ChildWorkflow window, select the required variables to be passed to the child workflow.



3. After specifying the data click the Update button.

Note: These variables are expected to exist in the child workflow. Make sure to define variables with same name and type as defined in parent workflow.

See Workflow - Variables for ChildWorkflow for a detailed description of the Variables for ChildWorkflow property window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

XMLVariables for ChildWorkflow - This property specifies the XML variables to be passed on to the child workflow. This window shows the list of XML variables defined for the current (parent) workflow. Select the XML variables that need to be passed to the child workflow.

Property Type: Optional (This property needs to be set only if required.)

Steps to set the XMLVariables for ChildWorkflow property

1. To specify the XML variables to be passed to the child workflow, click the button for the 'XMLVariables for ChildWorkflow' property to open the property window.
2. In the XMLVariables for ChildWorkflow window, choose the required variables to be passed to the child workflow.
3. Click the **Update** button.

Note:
- These XML variables are expected to exist in the child workflow. Make sure to define XML variables with same name and type as defined in parent workflow.

- To work with any XML variables or resource activities, SKEventData XML variables must be passed in the child workflow, where SKEventData contains some events information related to parent workflow which needs to be passed to child workflow. If SKEventData variable is not created, user can create XML variable of type XMLDocument with the same name 'SKEventData' by locating the 'SKEventData.xml' from the following path:*[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks*\Web\BPMUITemplates\Default\XMLSchemas.

See Workflow - Workflow - XmlVariables for ChildWorkflow for a detailed description of the XMLVariables for ChildWorkflow property window.

Wait for the child workflow? - This property determines whether the parent workflow should wait for the child workflow to complete.

Property Type: Optional (This property needs to be set only if required.)

Execution mode - This property determines whether the child workflow should be executed synchronously i.e., whether the child workflow can redirect the client to some URL.

Property Type: Optional (This property needs to be set only if required.)

Control child workflow? - This property determines whether the workflow control actions like pause, resume, and abort that are performed on the parent workflow should also be applied to the child workflow.

Property Type: Optional (This property needs to be set only if required.)

Note: Both properties namely Wait for the child workflow? and Control child workflow? should be set as No, if you do not want the parent workflow to wait till child workflow completes the execution.

Is Blocking Output Supported? - This property can be used along with the BlockingOutput activity.

The Workflow activity is a sleeping activity. The BlockingOutput activity may not return the expected results if any sleeping activity is executed before it. Use the **Is Blocking Output Supported?** property to overcome this behavior.

By default, the value for this property is set to **No**. Set the value to **Yes** if you want to use this property with the BlockingOutput activity. You can also set the value for this property through the Expression Editor.

Along with this property, set the values for the following properties as mentioned below.

- Set the value of **Wait for the child workflow?** to **No**.
- Set the value of **Execution mode** to **Synchronous**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Set the value of **Control child workflow?** to **No**.

Property Type: Optional (This property needs to be set only if required.)

Note: Consider a scenario where the **Set Workflow Variable Out** activity is used in the child workflow and then the return values are specified in the parent workflow. In this scenario, the BlockingOutput activity will not work.

Output Values

The property in the Output Values has been described below. You can use this property to specify the output values of the activity.

Return Values - This property can be used to add all the possible output values from the child workflow, for conditional routing. The child workflow should set a value to the 'ParentWorkflowOutput' variable. This variable will hold the output value from the child workflow. The value of the 'Wait for the child workflow?' property should be set to 'Yes'.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Return Values property

1. To specify the Return Values, click the button for the 'Return Values' property to open the property window.
2. In this window, enter the return value in the Return Values field.
3. After entering the return value details, click Add Row to add this return value and display a new row for entering another return value.
4. Remove rows if necessary by selecting them and clicking Remove Selected.
5. After specifying all the return values, click Update.

See Workflow - Return Values for a detailed description of the Return Values property window.

Activity outputs

The Workflow Activity does not have any standard return value. It will output the value specified by the user in the Return Values property.

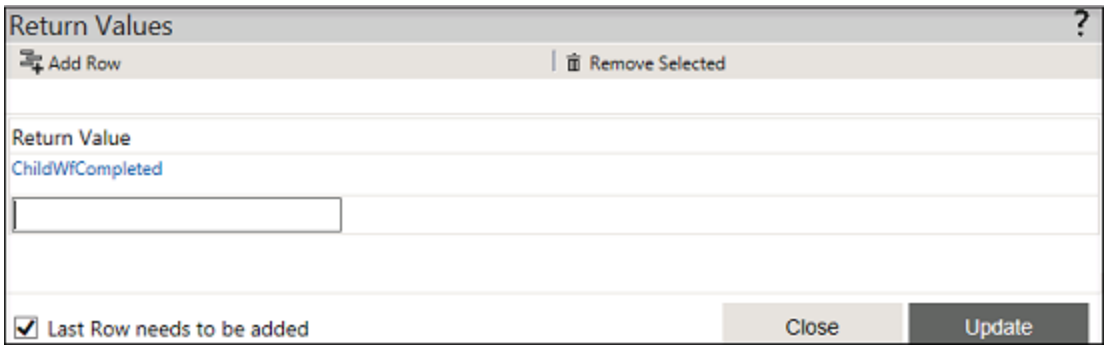
Example for a Workflow Activity

Configuring Parent Workflow to wait till Child Workflow Completes Execution

Steps to be followed for Parent Workflow configuration:

1. Set Wait for the child workflow property as Yes to wait for the child workflow to complete the execution.
2. Use Return Value property to set the routing condition after completion of child workflow. Example: ChildWfCompleted.

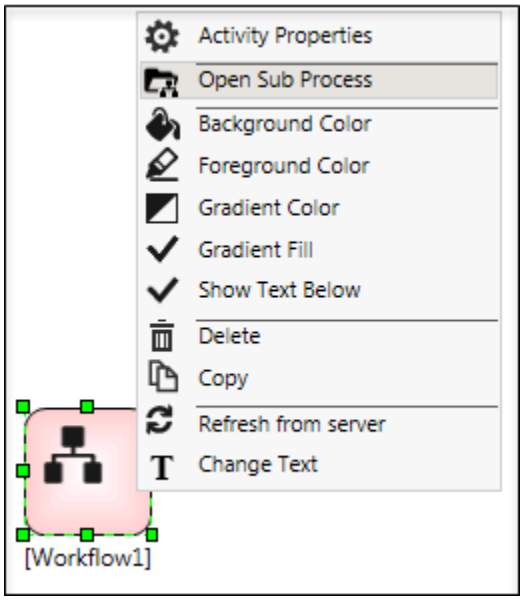
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



This will dynamically create a link output named 'ChildWfCompleted' for the activity. Once the child workflow complete its execution, it returns back to the parent workflow with the return value specified in the child workflow (In this case, child workflow should return 'ChildWfCompleted' value) and then it will continue with further execution.

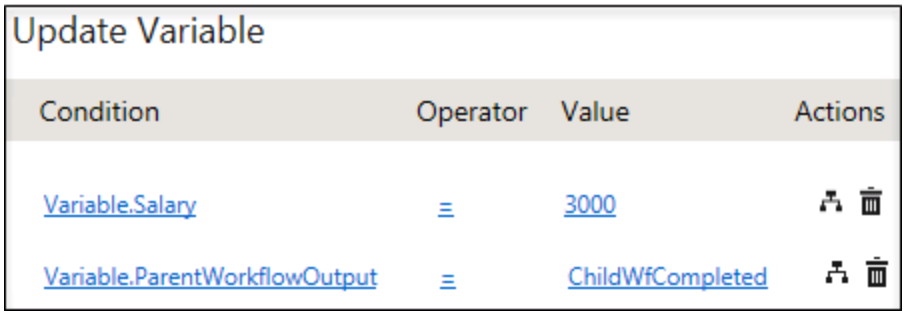
Steps to be followed for Child Workflow configuration:

1. Right-click the workflow activity to open child workflow using the context menu item Open Sub Process.



2. Declare a variable named 'ParentWorkflowOutput' of type string in the child workflow Start action.
3. Use Update Variable action to set the variable 'ParentWorkflowOutput' value as 'ChildWfCompleted' (This value should be same as the value for the Return Values property in parent workflow action properties).
4. The Update variable action to update the parent workflow output should be placed after the logical conclusion of child workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



After configuring child workflow, follow the steps given below:

1. Go back to the parent and set routing condition based on the output 'ChildWfCompleted'
   a. Parent workflow goes to sleep mode after the child is executed.
   b. Child completes execution with Update Variable as the last action.
   c. Once the child completes execution, the value set in Update variable is given as output for the workflow action in parent.

Note: If the child workflow needs to update any of the parent workflow variables, it is a must for Parent workflow to wait for the child workflow to complete.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Workflow - Contents for ChildWorkflow

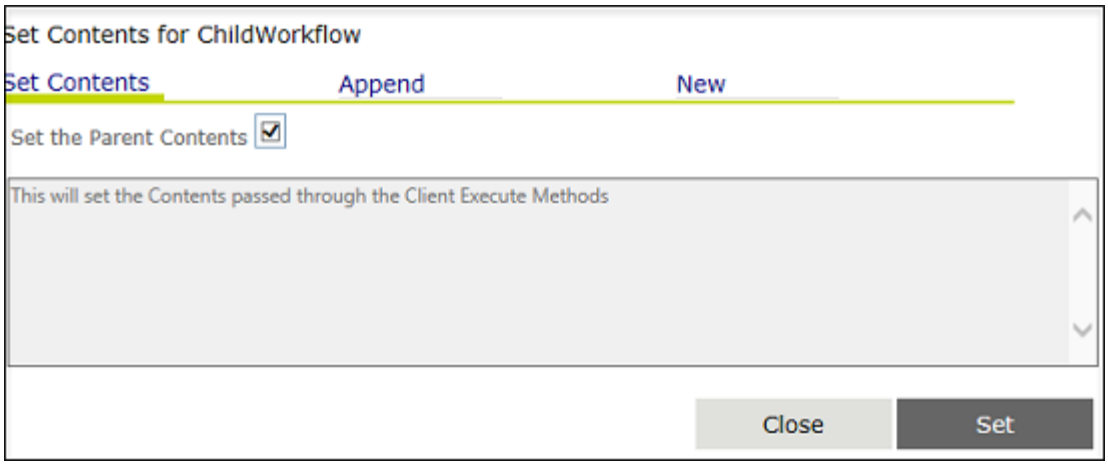
Contents for ChildWorkflow Property window

You can specify the contents to be passed to the child workflow in this property window. To open this window, click the button for the 'Contents for ChildWorkflow' property.

The window has three tabs – **Set Contents**, **Append**, and **New** – to specify the contents to be passed to the child workflow. Each tab opens the corresponding window to specify the content to set.

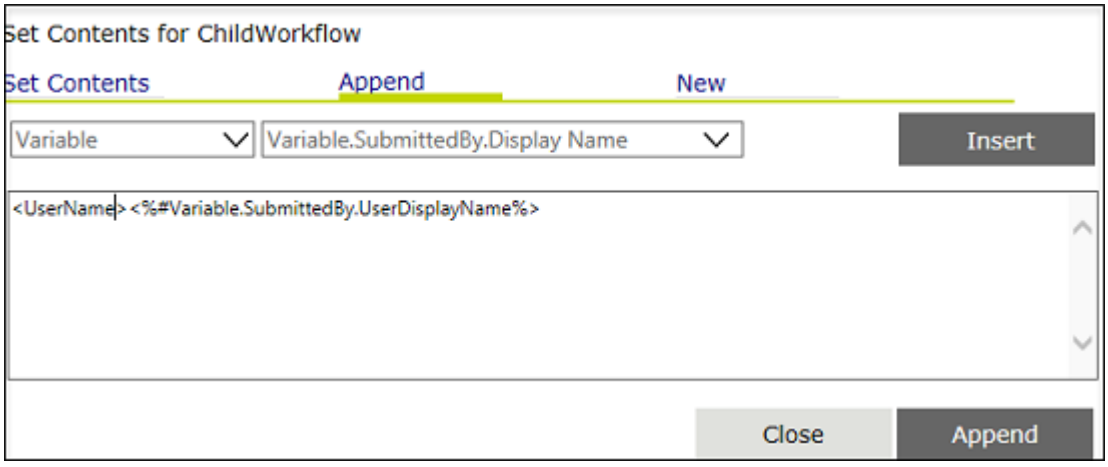
The following image shows the **Set Contents** tab of the **Contents for ChildWorkflow** property dialog box:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window shown above is displayed when the **Set Contents** tab is clicked. It is the default selection and is opened when this property window is launched. You can pass the contents that are passed through the Client. Execute Method to the child workflow in this window. You can click the **Set the Parent Contents** check box to do this. You can click **Set** to set the parent variables.

The following image shows the **Append** tab of the **Contents for ChildWorkflow** property dialog box:

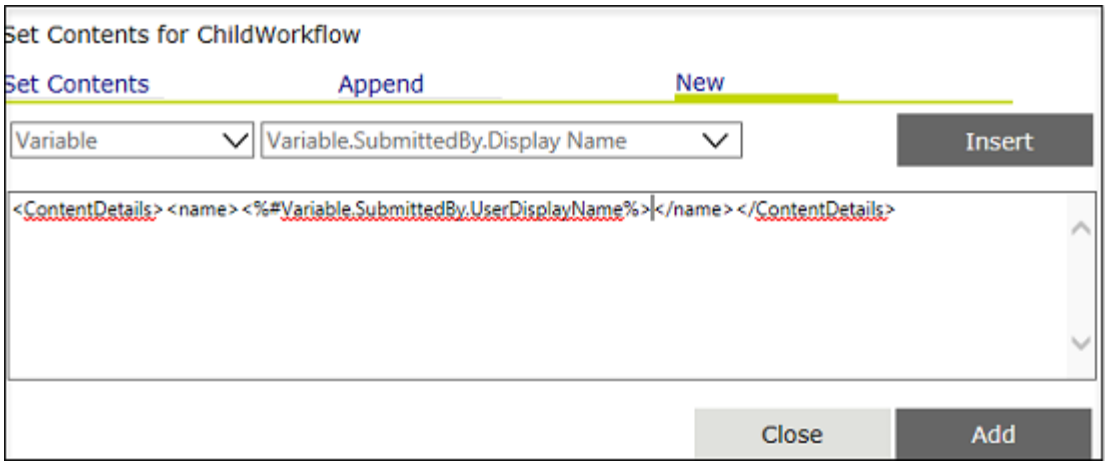


The window shown above is displayed when the **Append** tab is clicked. You can append values to the parent workflow contents in this window. This window has two drop downs for selecting the value type (Variable or Content) and the value respectively (Variable or Content collection). After specifying the values, you can click the **Insert** button to add it to the text area.

Click **Append** to append all the specified values to the parent workflow contents being passed to the child workflow.

The following image shows the **New** tab of the **Contents for ChildWorkflow** property dialog box:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window shown above is displayed when the **New** tab is clicked. You can add new content values to be passed to the child workflow in this window. This window has the value type (Variable or Content) and value (Variable or Content collection) drop downs for selecting the respective options. After specifying the values, you can click **Insert** to add it to the text area.

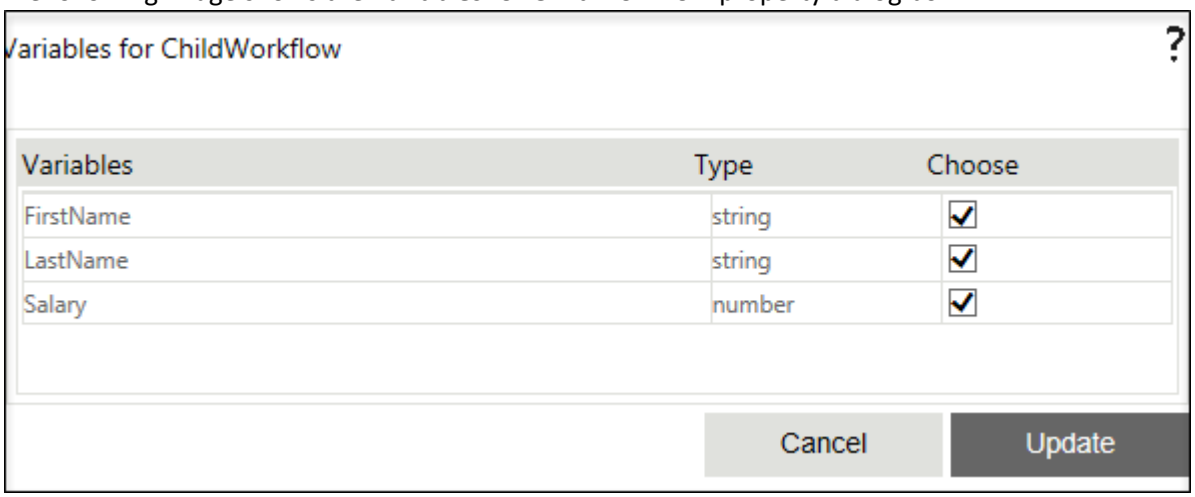
You can click **Add** to add all the specified values to be passed to the child workflow.

Workflow - Variables for ChildWorkflow

Variables for ChildWorkflow Property window

You can specify the variables to be passed to the child workflow in this property window. To open this window, click the button for the 'Variables for ChildWorkflow' property.

The following image shows the **Variables for ChildWorkflow** property dialog box:



The window has a grid that lists the parent workflow variables that can be passed to the child workflow. It has the following columns – **Variables**, **Type**, and **Choose**. You can click the **Choose** check box to select the variable. You can click **Update** to set selected the variables to be passed.

Note: You can create a local variable (that is, variables prefixed with '@', for example, @EmpId) which can be passed to the child workflow. For the child workflow to get this variable value, it should have a variable with the same name but without the '@'. For example, if a parent workflow has a local variable '@EmpId', then the child workflow should have a variable 'EmpId' to get its value.

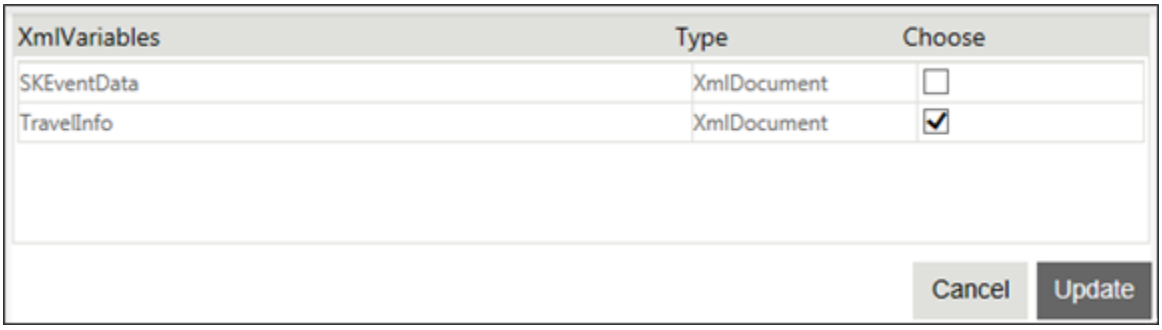
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Workflow - XmlVariables for ChildWorkflow

XmlVariables for ChildWorkflow Property window

You can specify the XmlVariables to be passed to the child workflow in this property window. To open this window, click the button for the 'XmlVariables for ChildWorkflow' property.

The following image shows the **XmlVariables for ChildWorkflow** property dialog box:



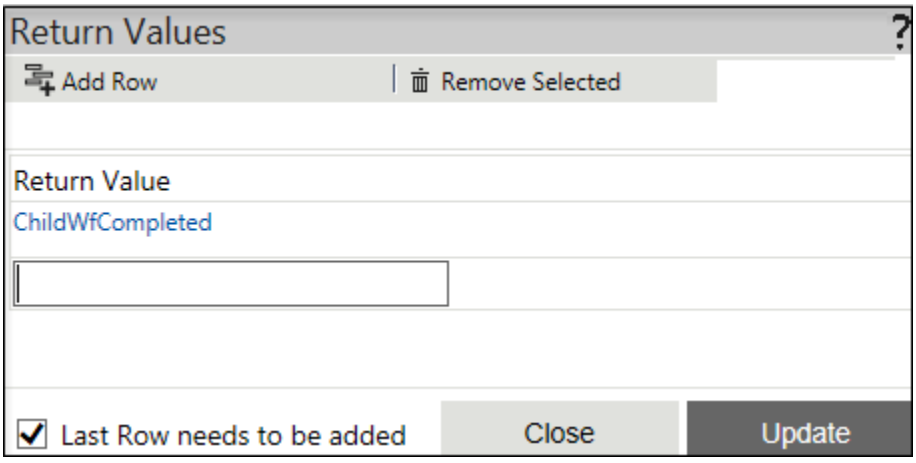
The window has a grid that lists the parent workflow XMLvariables that can be passed to the child workflow. It has the following columns –**XmlVariables**, **Type**, and **Choose**. You can click the **Choose** check box to select the XmlVariable(s). You can click Update to set selected the XmlVariable(s) to be passed.

Note: For **XMLVariables** of type **XMLDocument** the **Internal Storage** should be the same for both the parent and the child workflow.

Workflow - Return Values

Return Values Property window

You can set the return values for the child workflow in this property window. To open this window, click the button for the 'Return Values' property. It displays the values in a grid, which also has a blank row where you can enter a new return value.



The window has two tabs at the top to manage the rows of return values in the grid.

- **Add Row** - You can click this tab to add the newly entered return value and display a new row to the grid where you can add another value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Remove Selected** - You can click this tab to remove the return value selected in the grid.

The grid has the following parameters that you need to specify for the new return value.

- **Return Value** - You can use this field to enter the return value.

There are two buttons and a check box at the bottom of the screen.

- **Update** - You can click this button to save the newly added return value(s)
- **Close** - You can click this to close the property window without saving the newly added return value(s)
- **Last Row needs to be added** - You need to select this check box to save the last row added.

How to Pass Variables/Contents from Parent Workflow to Child Workflow?

The Workflow Activity enables the execution of a second (child) workflow from a (parent) workflow. The workflow activity can be used to call a child workflow from a parent workflow. The **Variables for the child workflow** property and **XMLVariables for the child workflow** property enable parent workflow to pass variable and XML variable information to the child workflow respectively.

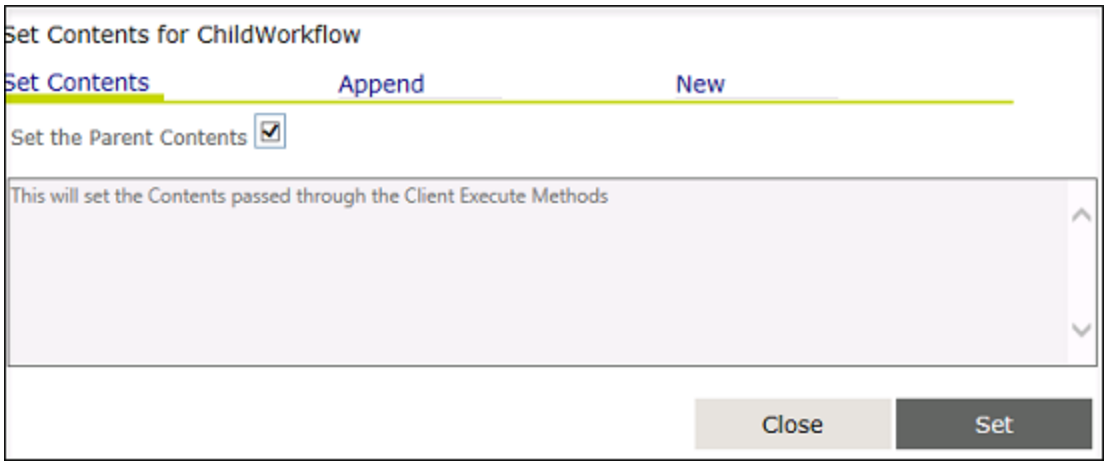
To pass variables or contents from parent workflow to child workflow, follow the steps given below:

1. In the parent workflow, right-click **Workflow Activity**, and then select **Activity Properties**.
2. Enter the repository name for **Repository** property. The child workflow should belong to the same repository of the parent workflow. The repository property of a child workflow is exposed for backward compatibility.
3. Enter the name of the workflow to be executed in the **Workflow** property.
4. Enter the version only if any specific version of the (child) workflow to be executed in the Version property. Leave this as blank, if the latest published version of (child) workflow has to be executed.
5. Build the content to be passed to the child workflow, if required in the **Contents for ChildWorkflow** property. Use one of the following options to specify the content to be passed:
   - Set the contents to be passed as same as Parent Contents
   - Append some more contents to the parent workflow contents
   - New workflow contents

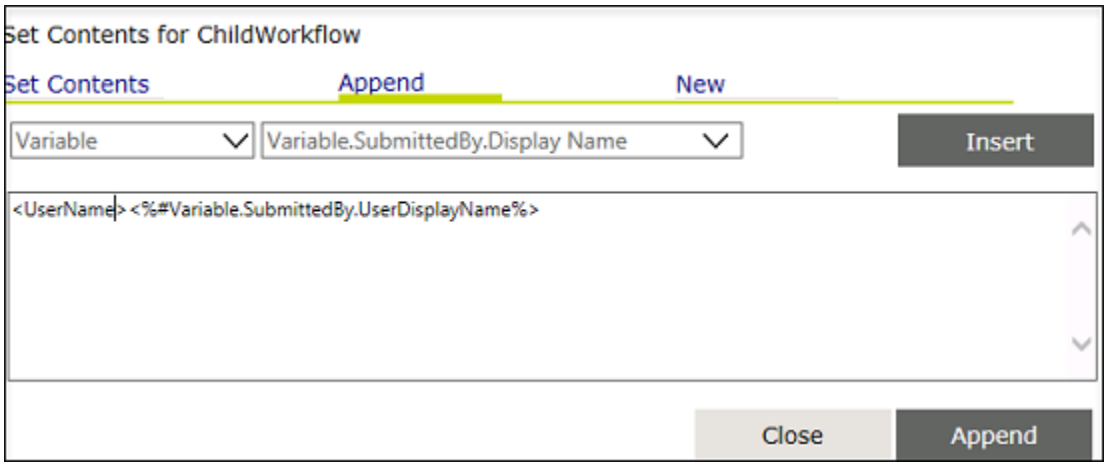
To set the Contents for ChildWorkflow property

1. To specify the contents to be passed to the child workflow, click the button for the 'Contents for ChildWorkflow' property to open the property window.
2. In the **Set Contents** window, the **Set Contents** tab is selected by default and the corresponding window is displayed.
3. Click the **Set the Parent Contents** check box to pass the parent contents XML to the child workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

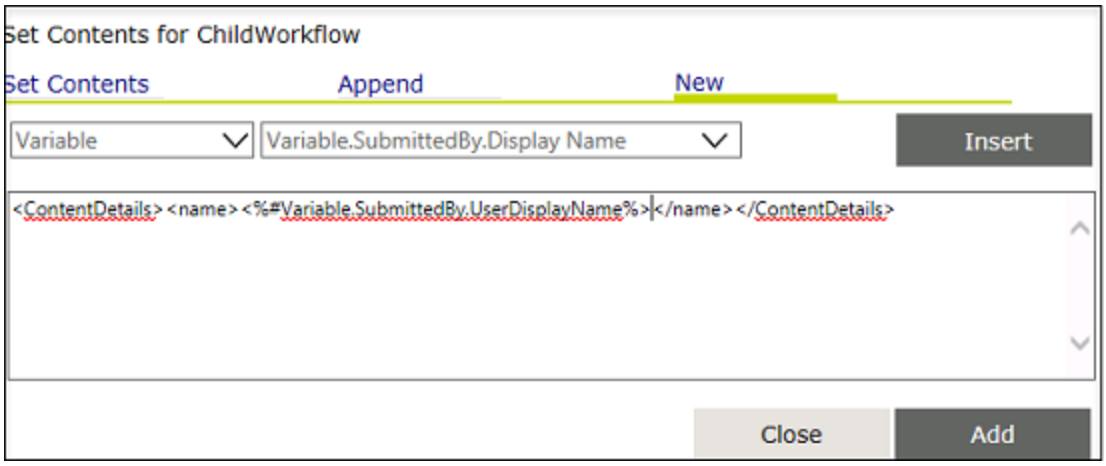


4. Click **Set** to set this property.
5. Alternatively, you can append XML content to the parent workflow contents and pass them to the child workflow. To do this, click **Append** to open the corresponding window.
6. Select a type of value (Variable or Content) and the corresponding value collection from the respective drop-down and click **Insert** to add the selection to the text area



7. Add more XML contents if necessary. After adding all the values, click Append to save the setting.
8. Third options to build complete XML content and can pass these contents to child workflow. To do this, click New to open the corresponding window.
9. In this window, select a type of value (Variable or Content) and the corresponding value collection from the respective drop-down and click the Insert button to add the selection to the text area.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

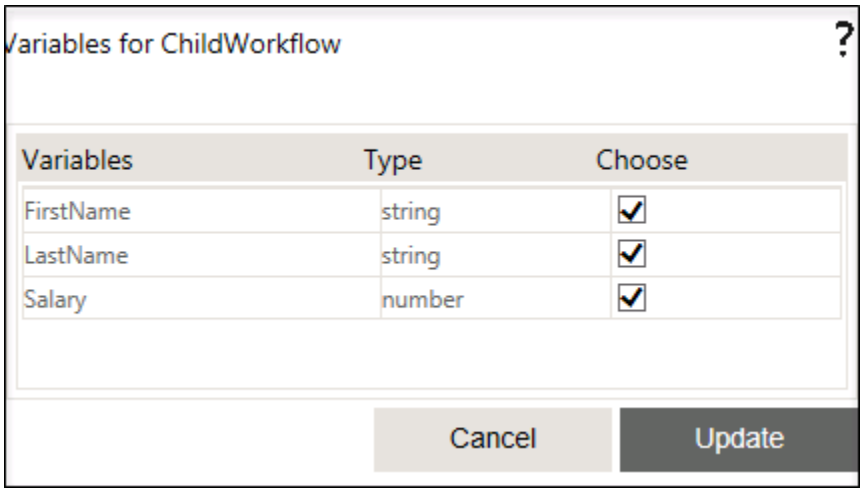


10. After specifying the new content, click Add button to save the settings.

Select the variables that need to be passed to the child workflow from Variables for ChildWorkflow property.

To set the Variables for ChildWorkflow property

1. To specify the variables to be passed to the child workflow, click the button for the 'Variables for ChildWorkflow' property to open the property window.
2. In the **Variables for ChildWorkflow** window, select the required variables to be passed to the child workflow.



3. After specifying the data, click Update.

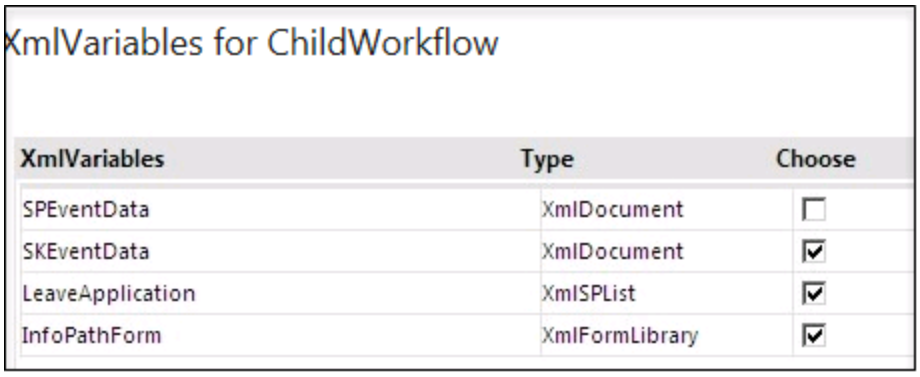
Note: These variables are expected to exist in the child workflow. Make sure to define variables with same name and type as defined in parent workflow.

Select the XML variables that need to be passed to the child workflow from XMLVariables for ChildWorkflow property.

To set the XMLVariables for ChildWorkflow property

1. To specify the XML variables to be passed to the child workflow, click the button for the **'XMLVariables for ChildWorkflow'** property to open the property window.
2. In the **XMLVariables for ChildWorkflow** window, choose the required variables to be passed to the child workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click the Update button.

Note: These XML variables are expected to exist in the child workflow. Make sure to define XML variables with same name and type as defined in parent workflow.
To work with any XML variables or resource activities, SKEventData XML variables must be passed in the child workflow, where SKEventData contains some events information related to parent workflow which needs to be passed to child workflow. If SKEventData variable is not created, user can create XML variable of type XMLDocument with the same name 'SKEventData' by locating the 'SKEventData.xml' from the following path: *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks*\Web\BPMUITemplates\Default\XMLSchemas.

Configuring Parent Workflow to wait till Child Workflow Completes Execution

To configure Parent Workflow

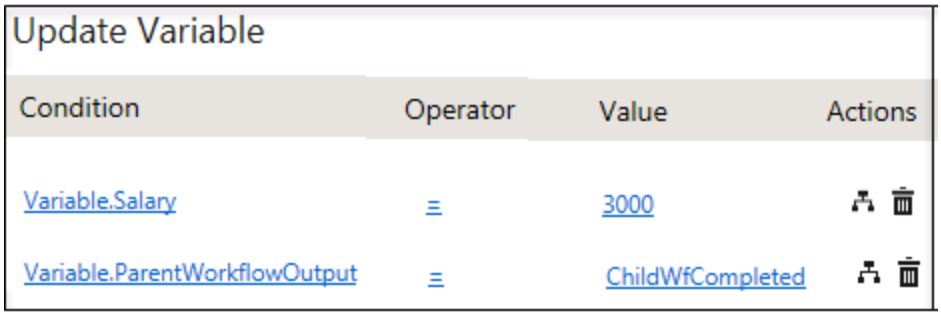
1. Set Wait for the child workflow? property as Yes to wait for the child workflow to complete the execution.
2. Use Return Values property to set the routing condition after completion of child workflow. Example: 'ChildWfCompleted'.

This will dynamically create a link output named 'ChildWfCompleted' for the activity. Once the child workflow complete its execution, it returns back to the parent workflow with the return value specified in the child workflow (In this case, child workflow should return 'ChildWfCompleted' value) and then it will continue with further execution.

To configure Child Workflow

1. Right-click the workflow activity to open child workflow using the context menu item Open Sub Process.
2. Declare a variable named 'ParentWorkflowOutput' of type string in the child workflow Start activity.
3. Use **Update Variable** activity to set the variable 'ParentWorkflowOutput' value as 'ChildWfCompleted' (This value should be same as the value for the Return Values property in parent workflow action properties).
4. The **Update Variable** activity to update the parent workflow output should be placed after the logical conclusion of child workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



After configuring child workflow, do the following steps:

1. Go back to the parent and set routing condition based on the output 'ChildWfCompleted'.
2. Parent workflow goes to sleep mode after the child is executed.
3. Child completes execution with Update Variable as the last action.
4. Once the child completes execution, the value set in Update variable is given as output for the workflow action in parent.

   Note: If the child workflow needs to update any of the parent workflow variables, it is a must for parent workflow to wait for the child workflow to complete.
5. Select **Asynchronous** from **Execution mode** drop-down list.
6. Select **Yes** from **Control child workflow?** drop-down.

Get Execution ID of the Child Workflow

You can get Execution ID of the Child Workflow from the Parent Workflow as follows.

To access Execution ID of the Child Workfow

1. Create a variable in the parent workflow with a name such as **"ChildWFExecutionId" + <<Child Workflow Action Name>>**.

   For example: **ChildWFExecutionIdWorkflow1**
2. Access the variable **<%#Variable.ChildWFExecutionIdWorkflow1%>**.

   The variable will return the value of the Child Workflow Execution ID.

Update Variable Activity

Activity description

The Update Variable Activity can be used to update the value of Variables and XmlVariable that can be used in the subsequent activities. Using this activity, the variable for For-Loop activity can also be initialized.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

The Update Variable activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking the Activity Properties tool in the tool bar or by selecting the appropriate option from the right-click menu for the Update Variable activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Exception Behavior

These options of Wonderware Skelta BPM 2009 have now been integrated with **Raise Error on Unhandled Linked Output** in the latest version of the product.

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The property in the Configuration group has been described below. You can use these properties to configure the functionality of the activity.

Set Variable - This property allows the user to specify the values for existing Variables or XMLVariables.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

See Variables for updating Variables.

See Forms Xml Variables to work with Forms XMLVariables.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity outputs

The UpdateVariableValue Activity returns the following values -

Updated: Displayed when the Variable or XMLVariable is updated

Not Updated: Displayed when the Variable or XMLVariable is not updated

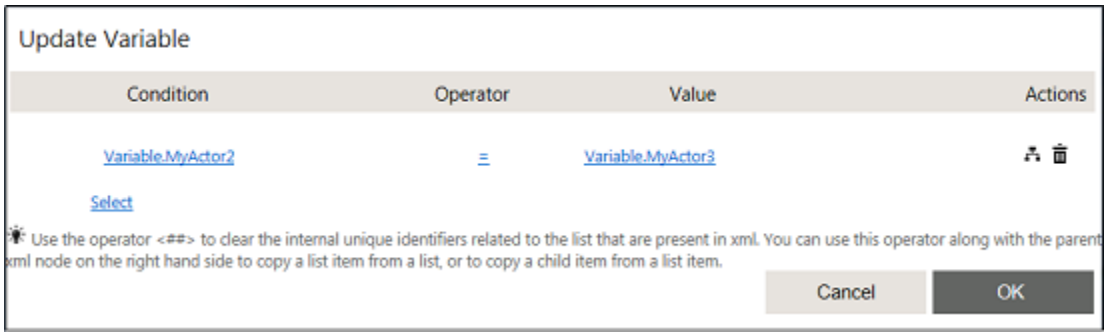
Variables

You can select the variables or XMLvariables and set the values to them in this property window. To open this window, click the button for the 'Set Variable' property.

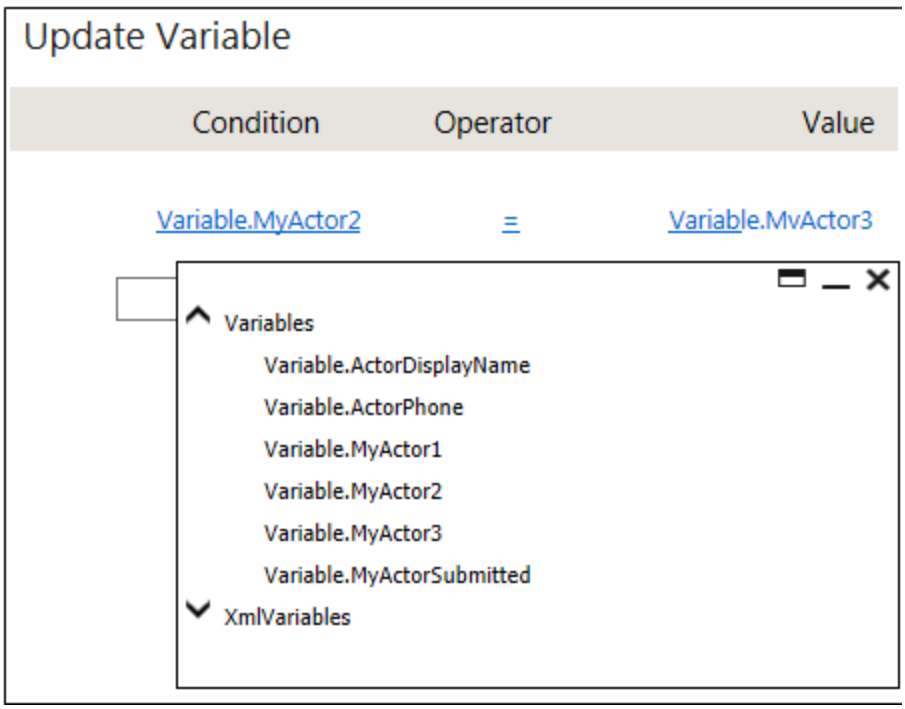
The following window shows the Update Variable Property window:

Steps to update a variable:To assign values to variables, click the button for the 'Set Variable' property to open the property window.

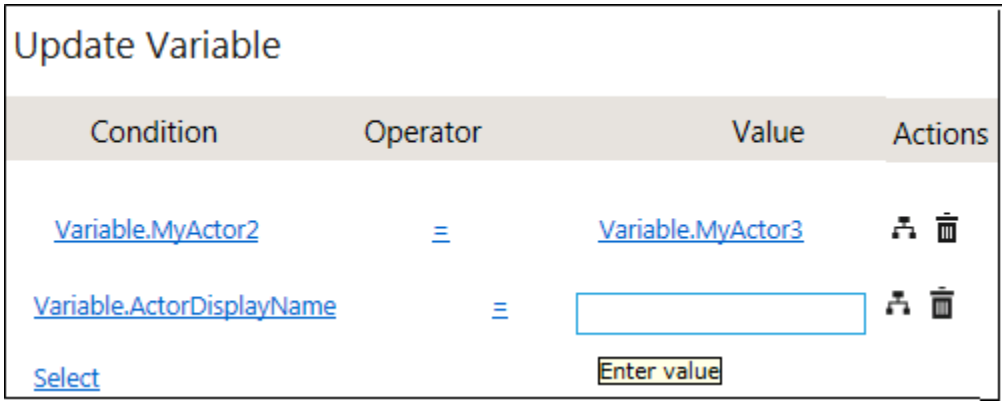
1. In this window, select the variable to which you want to assign or update the value.
2. Depending on your selection on the left hand side, the value can be specified directly in a text box, or as options in a drop down. You can select a variable collection, content collection, XML Variables collection, or build an expression. In the case of Variables, you will need to select the specific XML element from the schema. For example, select 'Name' on the left hand side.



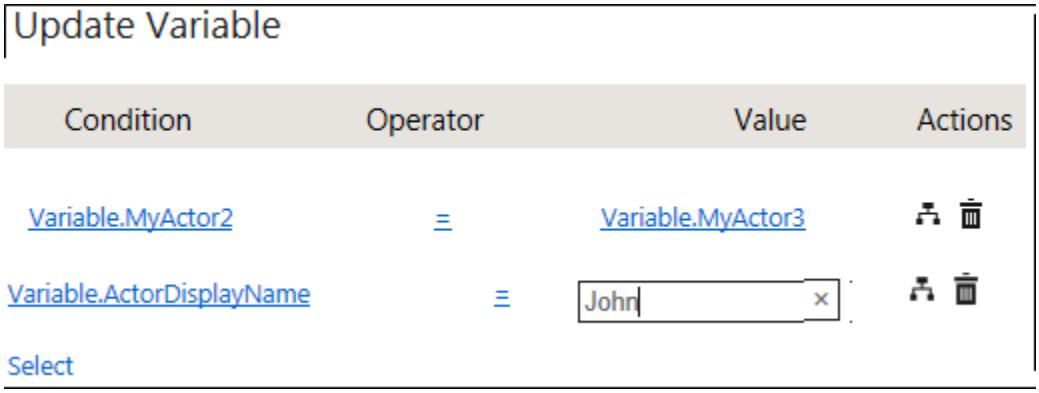
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. After selecting variable, the following screen is displayed to enter the value.



4. Enter the value for the variable in the text box. For example, enter name as 'John' as shown in the following figure.
The window shown in the image below is used to Update the Variables and their values.



5. If you want to create more variables, then click on Select to add variables.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. After setting the value for the variables, click **OK** button to save the settings.

   Note: Follow the same steps for updating XMLVariable.

See Forms Xml Variablesfor more information.

Note: For number data types, the minimum and maximum values are not validated. Enter only valid numeric values for number data types.
For numeric, DateTime, Date, or Time data types, ensure that the respective value is based on the culture set. You can enter the input or use the Date and Time picker to enter the required date and time.
Date.CurrentDate is not supported. Use the tree view option to pick the required date and time.
The time format supports long data type to enable you to enter seconds.

Forms Xml Variables

Updating the list item that alerted the workflow

AVEVA Work Tasks allows you to update the list item that alerted the workflow using XML List type XML variable. Update Variable activity can be used to add or update an item to the list. To update an item you need to select the scope as Current from Choose the List Item to Work on options. You can also use **List Navigator Activity** to set the scope.

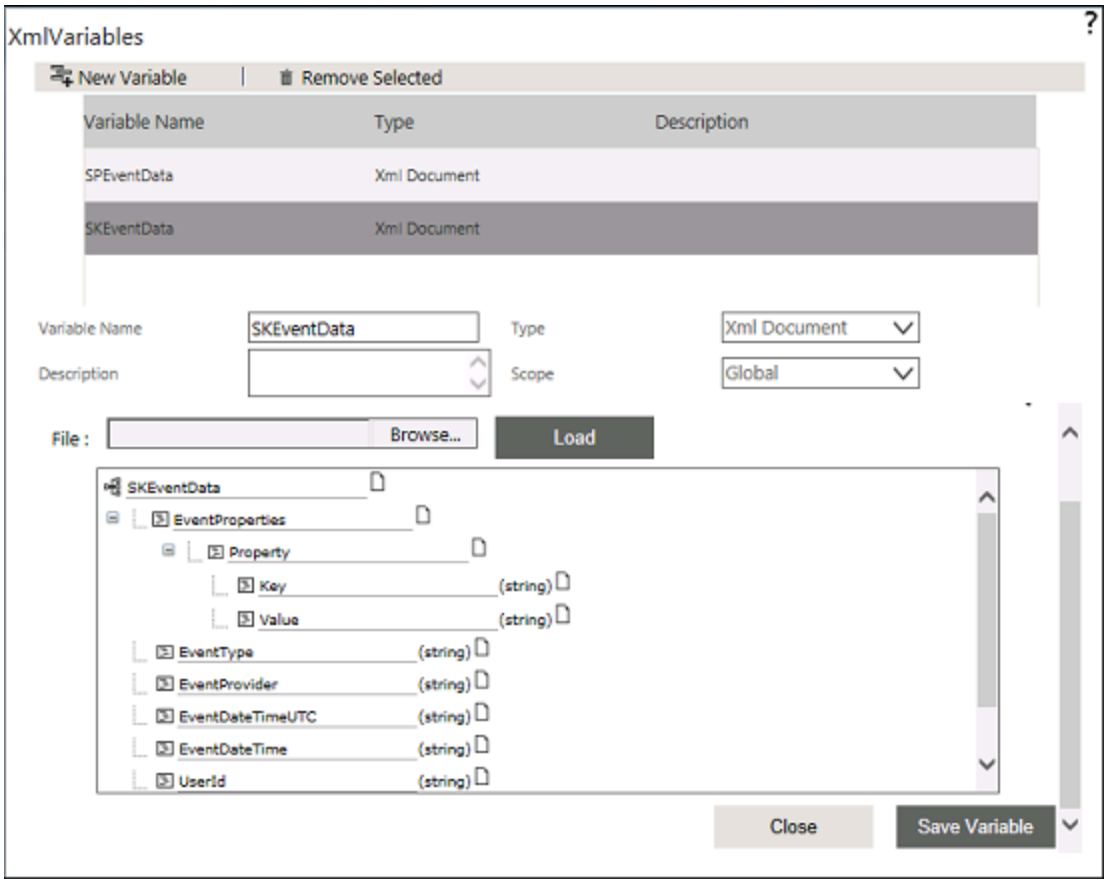
Assume that you have created a list called **Purchase Orders** with the following fields and alerted the workflow.

- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price
  - Item Total
- Total Purchase amount

Now you want to update an item in the same list. Follow the procedure.

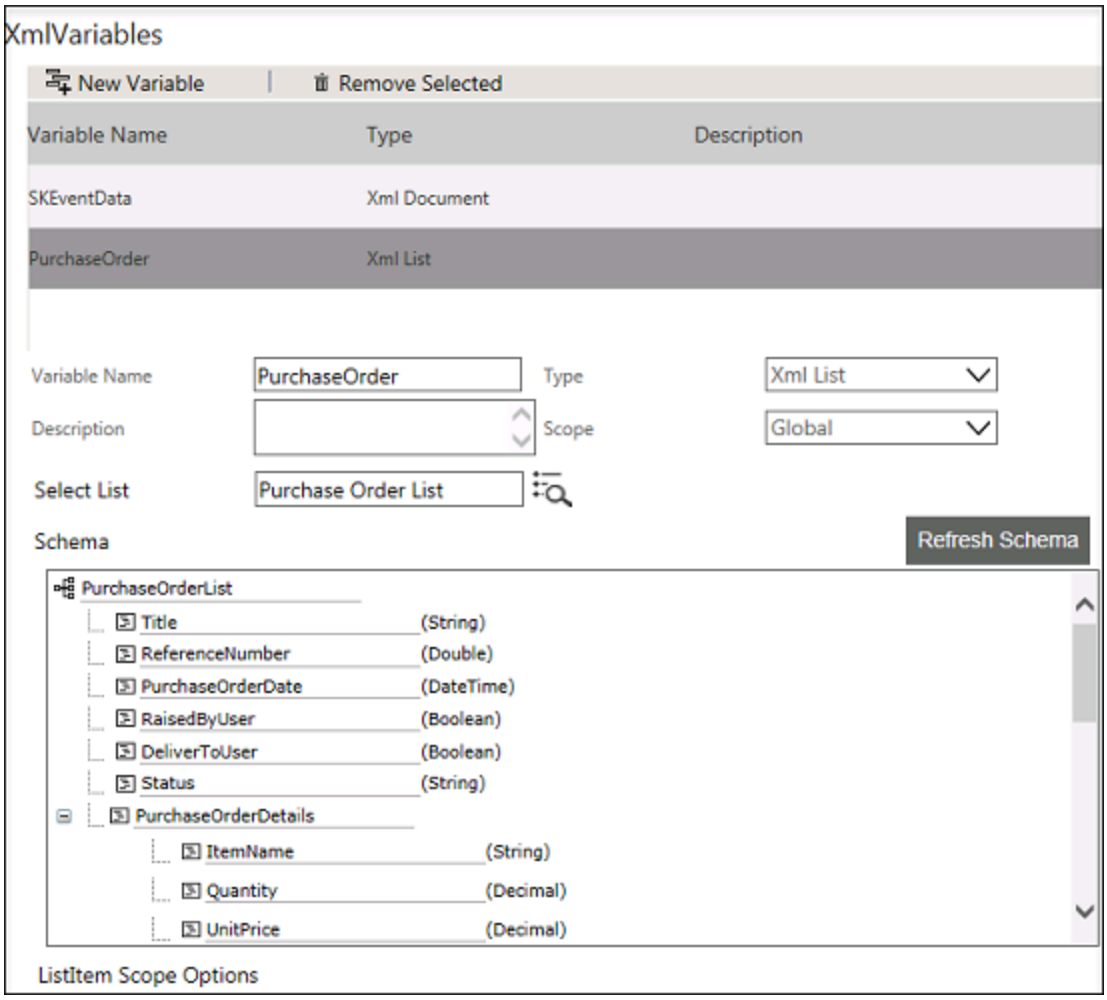
1. Log on to the **Enterprise Console** and go to workflow designer page.
   You need to create an **XML List** type XML variable for the **Purchase Orders** list. If the variable is not created, follow the below mentioned procedure to create.
2. Click the edit icon in the **XMLVariables** property from the **Start Activity** Properties. The following screen appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click **New Variable** to add a new variable.
4. Enter the variable name as **PurchaseOrder**.
5. Select **XML List** as **Type** from the drop-down list.
6. Enter the List name or click the icon to select the list. In this case select **Purchase Order List**.
7. Select the scope **Current** option from the **Choose the ListItem to work on** options to point to an item in the **Purchase Order List**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

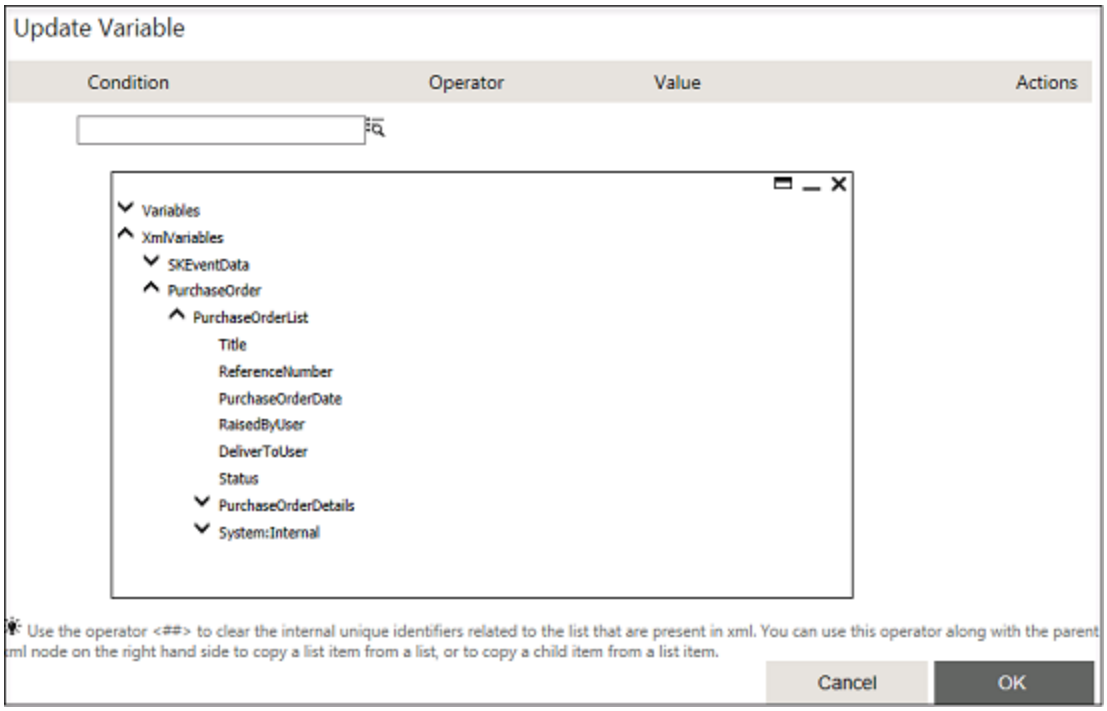


8. Now you can see the XML nodes in a tree-view that are mapped to the controls while creating the list.
9. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window.

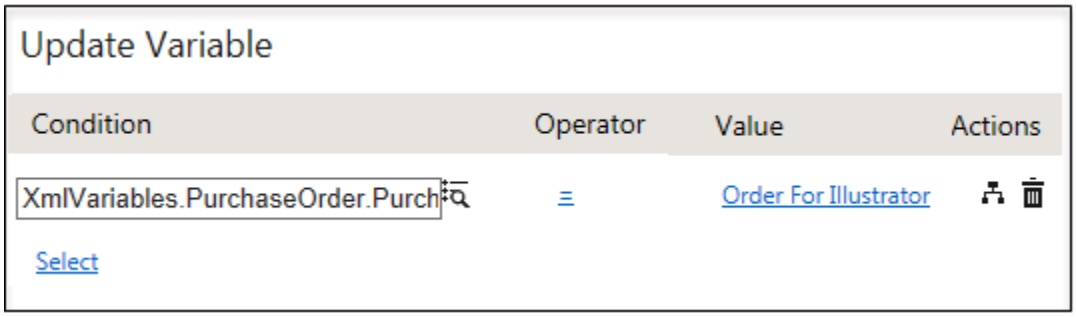
Note: If you set the scope as New while creating the XML List type XML variable, you can change it in to Current using List Navigator activity.

10. Drag and drop the **Update Variable** activity to update an item in the list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



11. Click the edit icon in the **Set Variables** property.
12. Click the **Select** link and select the respective field of the **XML List** type XML Variable created for the list (in this case **PurchaseOrder**). The fields displayed in the tree-view are representing the XML nodes of the controls in the list.
13. Select a field for that you want set the value from the tree-view. Then enter the value for the field in the right-hand side text box or click to set the value.
14. If you want to set the value, select **enter a value** from the tree-view, and then enter the value. Also you can select variables or contents or any XMLvariables nodes directly by selecting the respective node from the tree-view. If you want to combine multiple node values, you can build an expression by selecting **Build Expression**.



15. The same way you can set the value for other fields that you want to update. In this case, you are only updating the field 'Title' which has the value 'Order for Illustrator'.
16. Click **OK**. When you execute this workflow, the item get updated based on the value that you have set.

Note: For number data types, the minimum and maximum values are not validated. Enter only valid numeric values for number data types. For numeric, DateTime, Date, or Time data types, ensure that the respective value is based on the culture set.
You can enter the input or use the Date and Time picker to enter the required date and time.
Date.CurrentDate is not supported. Use the tree view option to pick the required date and time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The time format supports long data type to enable you to enter seconds.

Update Parent Variables Activity

Activity description

In AVEVA Work Tasks, a parent workflow variable can be updated from a child workflow provided both child and parent workflow belong to the same repository. This can be achieved by using Update Parent variables activity.

Activity Properties

The Update Parent Variables activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right-click menu for the Update Parent Variables activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

Use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

Use these properties to configure the functionality of the activity.

Parent Workflow Application Name - This property specifies the name of the application to which the parent workflow belongs. You can specify the application name directly here or click on the icon to open the Expression Builder and build an expression for the application name.

The parent workflow should belong to the same repository of the child workflow. This property of a child workflow is exposed for backward compatibility.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Parent Workflow Name - This property specifies the name of the parent workflow. You can specify the workflow name directly here or click on the icon to open the Expression Builder and build an expression for the workflow name.

The parent workflow should belong to the same repository of the child workflow.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

List Parent Variables - This property specifies the parent workflow variables to be updated. Click on the edit icon to open the

'List Parent Variables' property window.

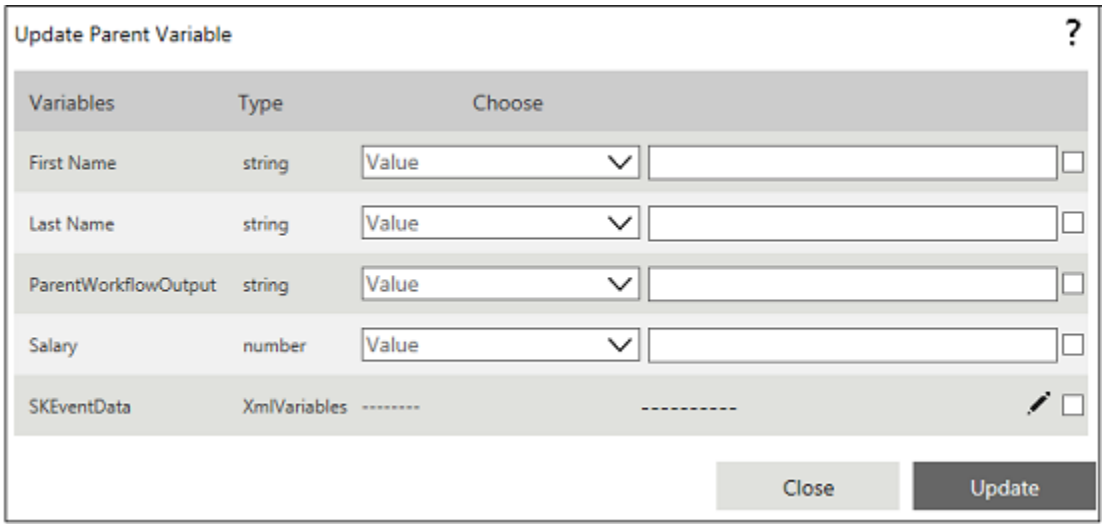
Property Type: Mandatory (This property must be set if the activity is to be executed.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Set the List Parent Variables property

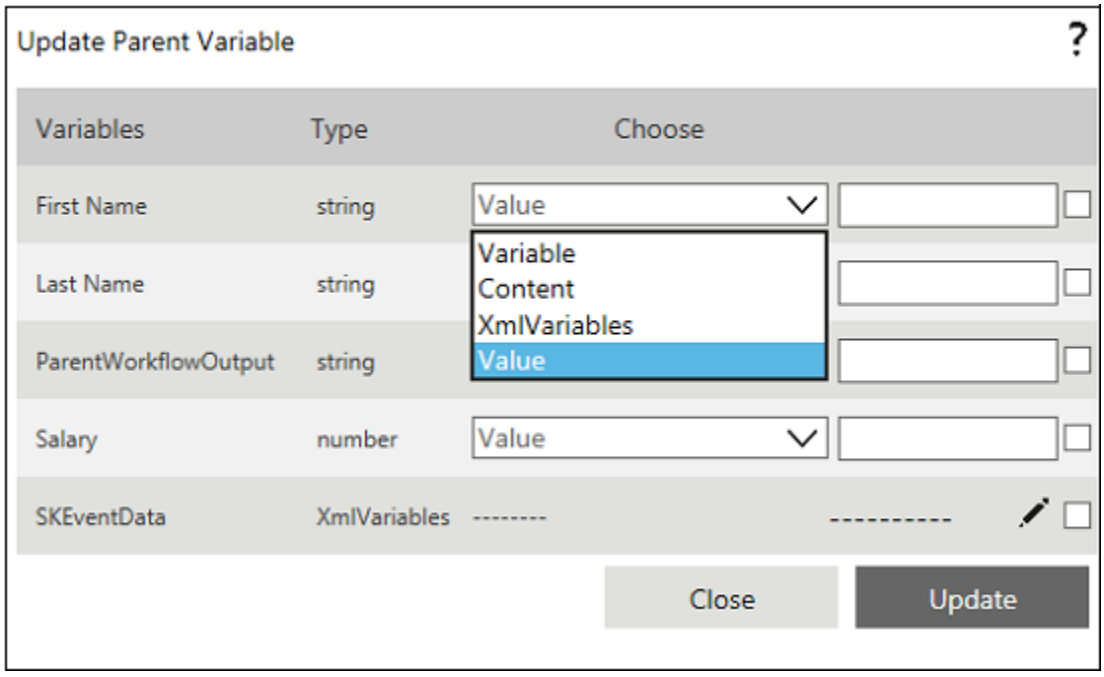
To update the value of variable:

1. Click the edit button of the **List Parent Variables** property. The **Update Parent Variables** window appears. Specify the parent workflow variables in this window.
2. Select the check box of the respective parent variable.



3. Choose the option using which the parent variable to be updated. The parent variable values can be updated using the Value, Variable, Content, or XML Variable of the child workflow.
4. Depending on the choice, select the respective options or node, and then click **Update** to save.

Note: The value can also be hardcoded using the Value option as shown in the following image.



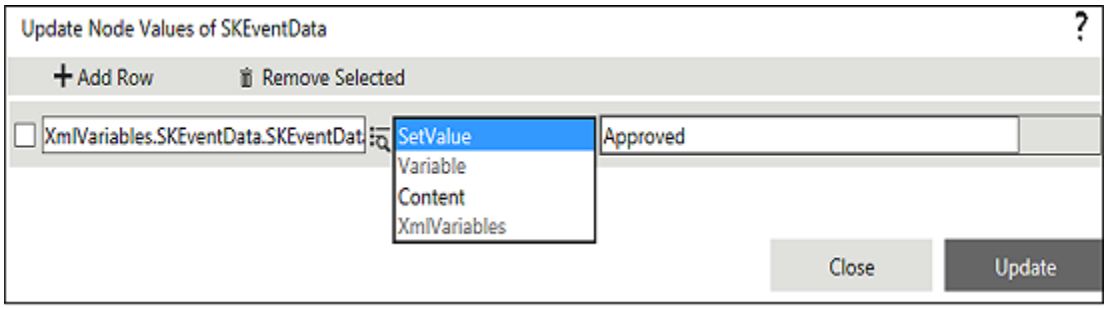
5. After selecting and assigning values to the variables, click **Update** to save.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To set the value of XML Variable

1. Select the check box of the respective parent variable.
2. Clicking the icon on the left hand side of the check box opens the window to specify the node to be updated of the respective parent XMLVariable.
3. Select the node to be updated from the tree view.
4. Choose the option using which the parent XML variable node to be updated. The parent XML variable node value can be updated from the variable, content or XML variable of the child workflow.
5. Depending on the choice select the respective option/node from the RHS.

Note: The value can also be hardcoded using the SetValue option as shown in the following figure.



6. You can specify multiple nodes for updating by using the **Add Row** option.
7. In order to remove any node that is set from updating, select the respective entry by selecting the check box in the left hand side and click **Remove Selected**.

See Update Parent Variables - List Parent Variables for a detailed description of the List Parent Variables property window.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

- Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs: The Update Parent Variables Activity returns the output value set in the Set Workflow Output property.

Exception Behavior

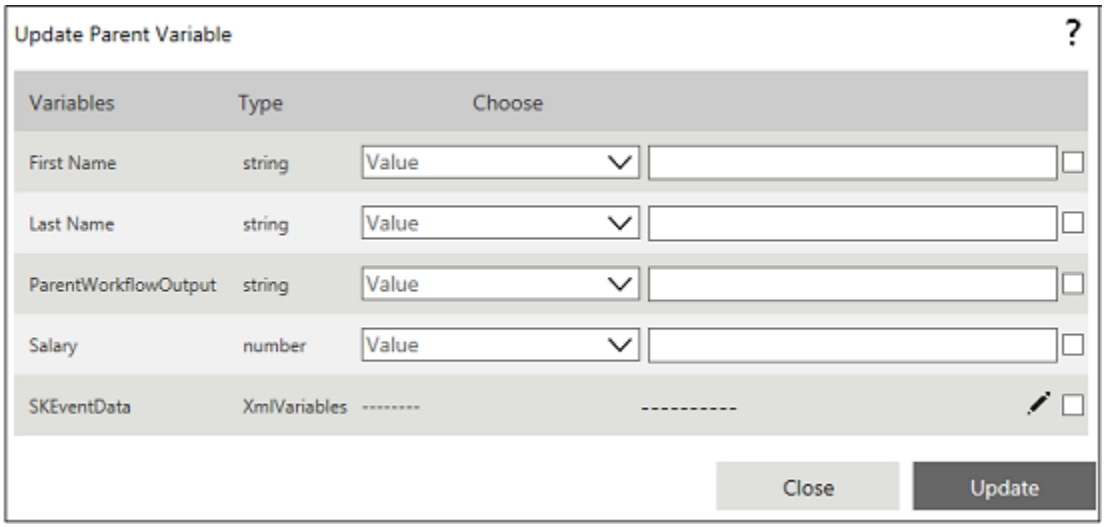
Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Update Parent Variables - List Parent Variables

You can set the List Parent Variables for this activity in the Update Parent Variable property window.



Update Parent Variable property window

The **Update Parent Variable** property window has a grid that lists all the variables belonging to the parent workflow which are specified in the Parent Workflow Name property. Each row in the grid represents a parent workflow variable, showing the variable name and type under the respective columns. The window also contains the following controls to specify the value to update the variable with:

- The Value type drop-down menu: Select the type of value to be assigned to the parent variable. Values that you can specify are Variable, Content, Value, or XMLVariables.
- The Value text box/drop-down menu: Specify the actual value to assign to the parent variable. The control displayed depends on the option selected in the Value type drop-down menu. If you select "Value", a text box appears. You can enter a discrete value in this box to complete the expression. If you select "Variable", "Content", or "XMLVariables", a drop-down menu appears in this field. This drop-down menu contains variable collection, content collection, or XMLVariables collection from which you can select an appropriate option.
- The check box: Select this check box to update the value of the corresponding parent variable at the workflow execution time.
- To set values for an XML Variable, click the icon, and then enter the required values in the **Update Node Values** window. For more information, see How to Update a Parent Workflow Variable from a Child Workflow?.

Note:
- For number data types, the minimum and maximum values are not validated. Enter only valid numeric values for number data types. For numeric, DateTime, Date, or Time data types, ensure that the respective value is based on the culture set. You can enter the input or use the Date and Time picker to enter the required date and time. The time format supports long data type to enable you to enter seconds. If you enter a valid value for a numeric, date, time, or datetime data type, and if you enter unwanted strings after the valid values, then while submitting the values, the extra strings at the end are truncated. For example, for a DateTime value entered as 01/31/2015 10:11:12ababa, then ababa is truncated by default.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Only single instance of Update Parent Variables activity can be used in child workflow.

How to Update a Parent Workflow Variable from a Child Workflow?

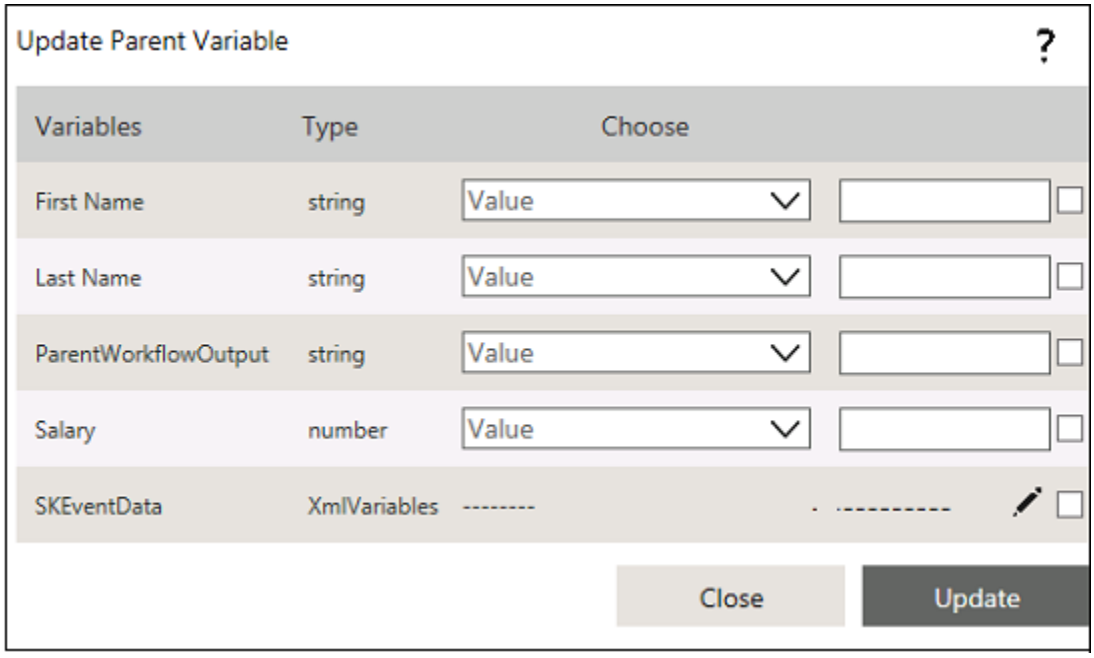
1. In AVEVA Work Tasks, a parent workflow variable can be updated from a child workflow provided both child and parent workflow belong to the same repository. This can be achieved by using **Update Parent Variables** activity.

To update parent workflow from a child workflow, do the following steps:

1. Right-click **Update Parent Variables Activity** and select **Activity Properties**.
2. Enter the parent workflow's repository name in the **Parent Workflow Application Name** property. The parent workflow should belong to the same repository of the child workflow. This property of the Update Parent Variable activity is exposed for backward compatibility.
3. Enter the parent workflow name in the **Parent Workflow Name** property. The parent workflow should belong to the same repository of the child workflow.
4. Click **List Parent Variables** property to update the variables in the parent workflow.

To update the value of variable

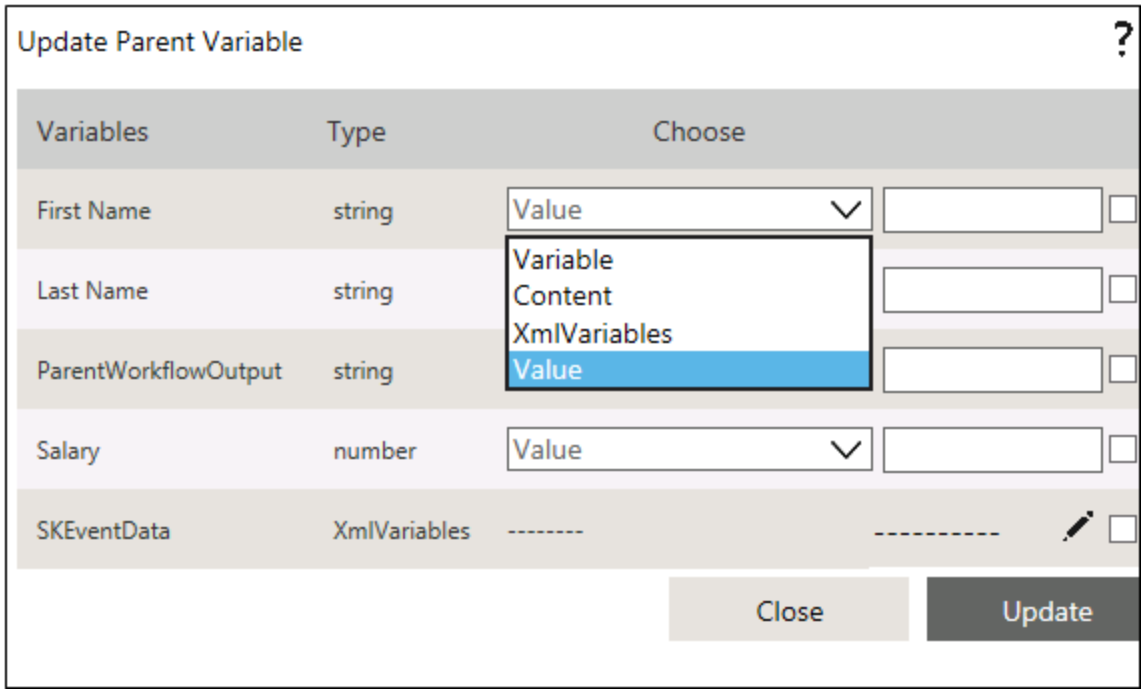
1. To specify the parent workflow variables, click the button for the **'List Parent Variables'** property to open the property window.
2. In the **Update Parent Variables** window, select the check box of the respective parent variable.



3. Choose the specific option from the list using which the parent variable to be updated. The parent variable value can be updated from the Value, Variable, Content, or XML Variable of the child workflow.
4. Depending on the choice, select the respective text box/options/node from the RHS and click **Update** to save.

   Note: The value can also be hard coded using the Value option as shown in the following figure.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

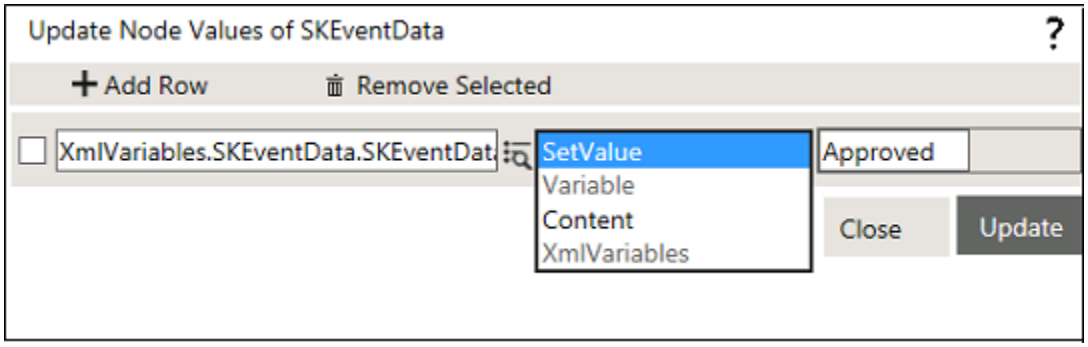


5. After selecting and assigning values to the variables, click Update to save.

To set the value of XML Variable

1. Select the check box of the respective parent variable.
2. Click the icon on the left hand side of the check box. The **Update Node Values** window appears. Specify the node to be updated for the respective parent XML Variable.
3. Select the node to be updated from the tree view.
4. Choose the option using which the parent XML variable node to be updated. The parent XML variable node value can be updated from the SetValue, Variable, Content, or XMLVariable of the child workflow. Depending on the choice, select the respective value/options/node from the RHS.

Note: The value can also be hard coded using the SetValue option as shown in the following figure.



5. You can specify multiple nodes for updating by using the **Add Row** option.
6. To remove any node that is set from updating, click the respective check box of the entry to be deleted, and then click **Remove Selected**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Synchronize Activity

Activity description

The Synchronize Activity synchronizes the execution of different activities. The engine does not proceed with the execution of further actions until all the actions leading to this activity are completed.

Activity Properties

The Synchronize activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Synchronize activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

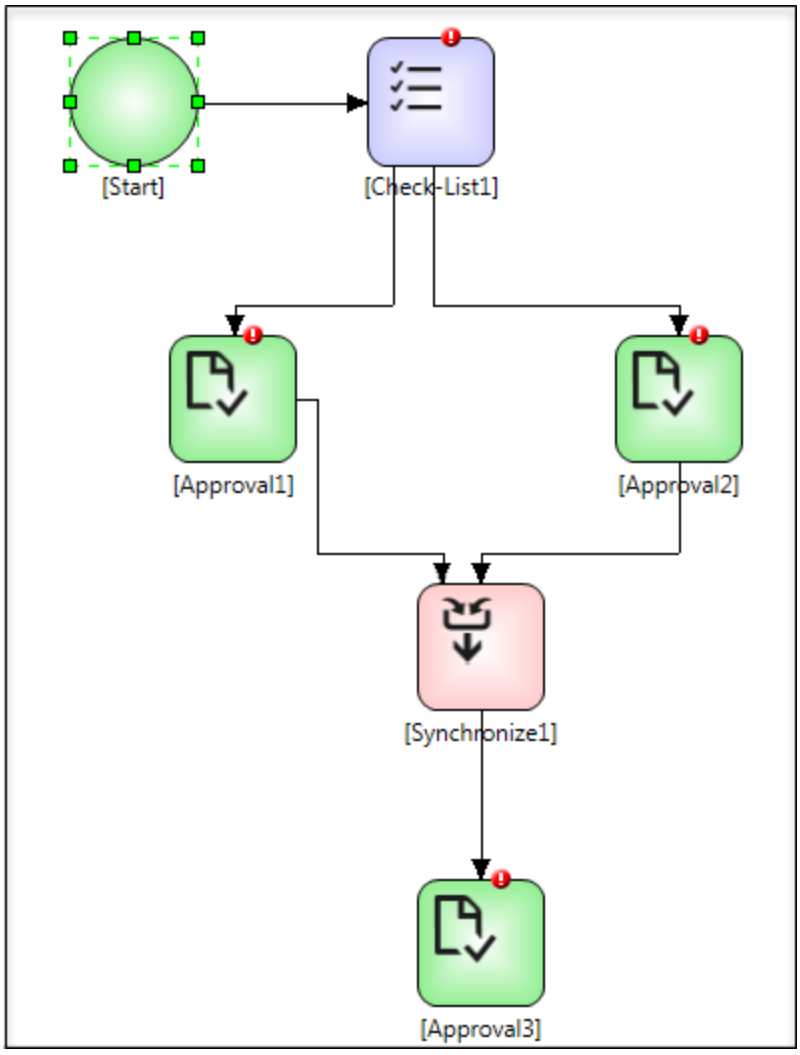
Activity outputs

The Synchronize Activity does not return any value.

Example for a Synchronize Activity:

Suppose an Employee applies for leave, then his leave has to be approved by two managers. When the leave is approved by both the managers, employee should get the information. In the following workflow, a checklist connected to approvals. The two approvals are connected to synchronize activity. Only when the output of both the approvals is approved the information will go to the applied user.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Set Workflow Output Variable

Activity description

The Set Workflow Output Variable activity can be used to set the output of the parent workflow from the child workflow.

Note: Alphanumeric, space, and dot are the only supported characters for workflow output variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

The Set Workflow Output Variable activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the tool bar or by selecting the appropriate option from the right-click menu for the Set Workflow output variable activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

You can use this property to configure the functionality of the activity.

Set Workflow Output - This property specifies the output of the workflow. You can specify the output value directly here or click the icon to open the Expression Builder and build an expression for the output.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

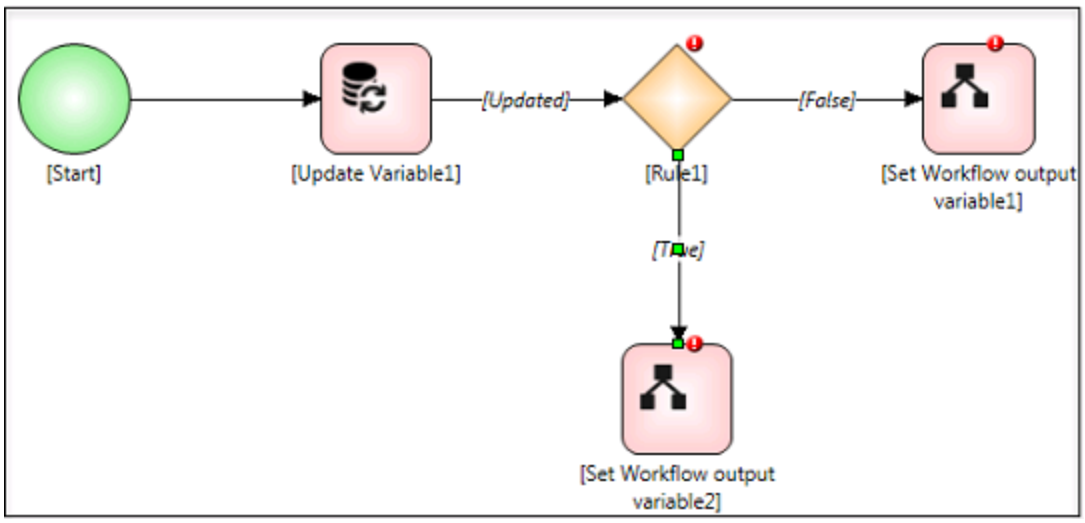
Activity outputs

The Set Workflow output variable Activity returns the output value set in the Set Workflow Output property.

Example for setting Workflow Output Variable

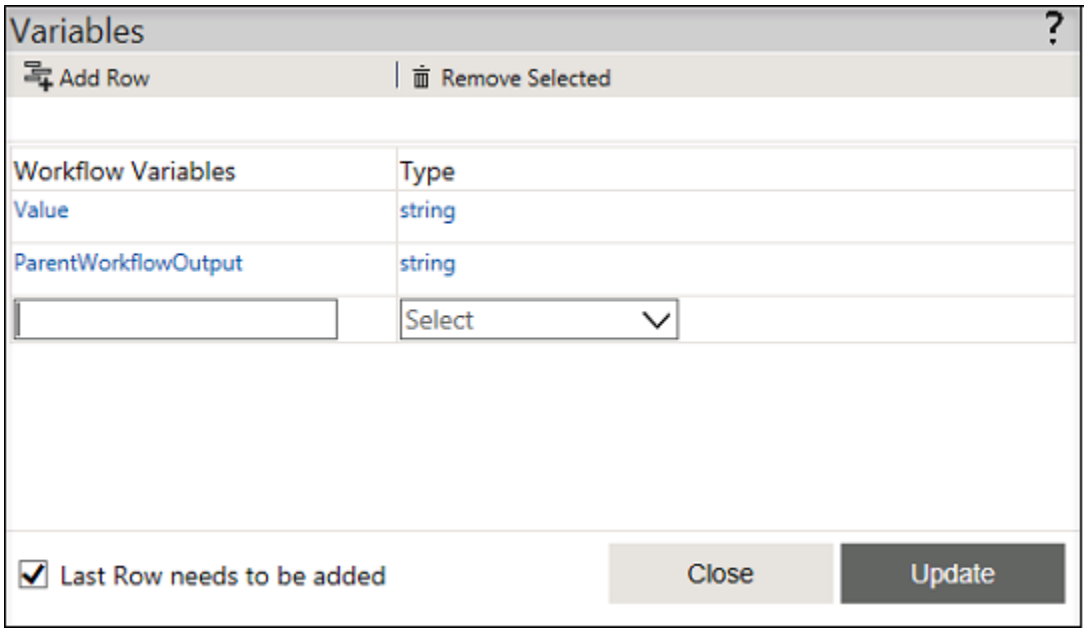
1. Design a workflow as shown in the below figure and name it as ChildWF.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



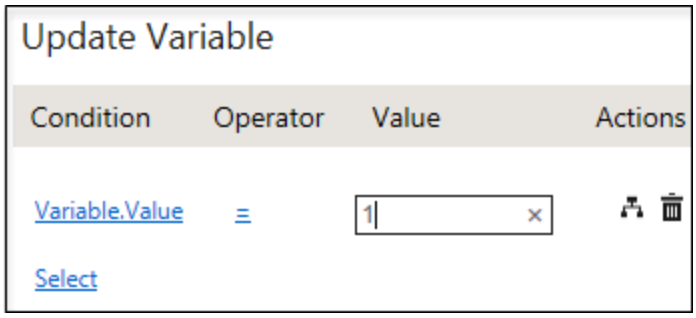
2. Perform the following actions:
   a. In the Start Activity, declare two variables Value and ParentWorkflowOutput as String.

Note: The user must declare a same variable name in the child workflow which was declared for the Parent workflow. In the above example, user must declare by the name "ParentWorkflowOutput" of type "String" in the child workflow.



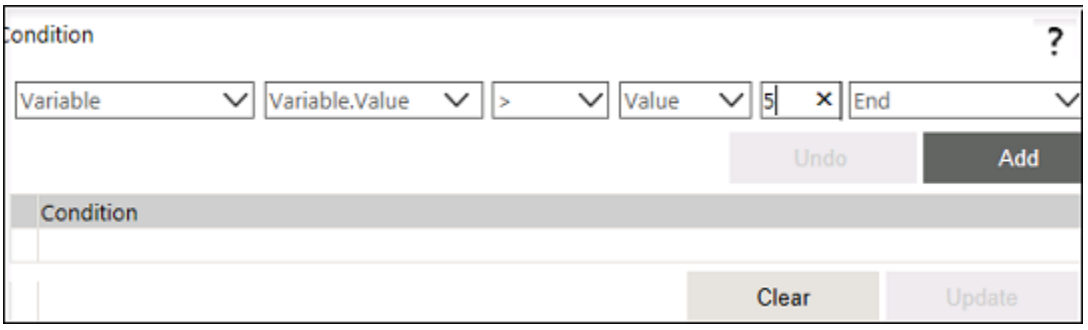
   a. In the Set Variable property of Update Variable set the value of variable Value as 1 as shown in the figure.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



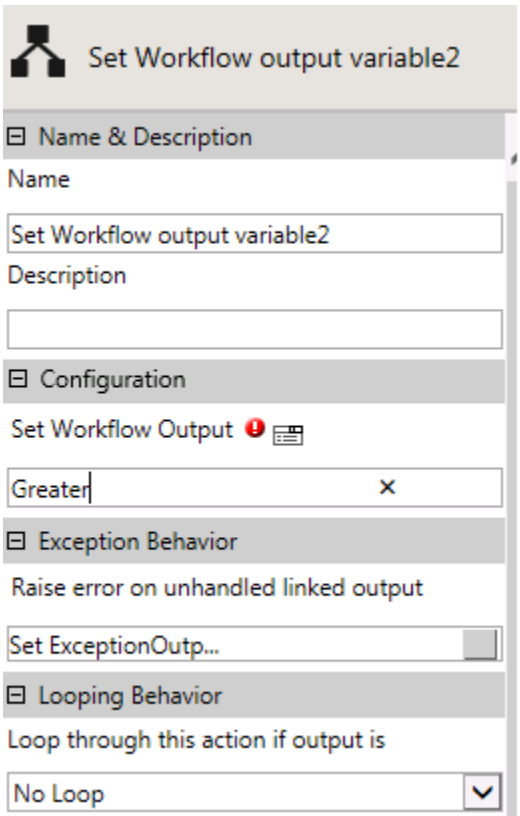
b. On the link connecting Update Variable and Rule activity set the output as Updated.

c. In the Configuration property of Rule, set the condition [Variable.Value]>["5"] as shown in the below figure.



d. On the link connecting Rule and SetWorkflowOutputVariable2 Activity set TRUE.

e. In the Set workflow output property of SetWorkflowOutputVariable2 Activity set it as Greater as shown in the following figure.

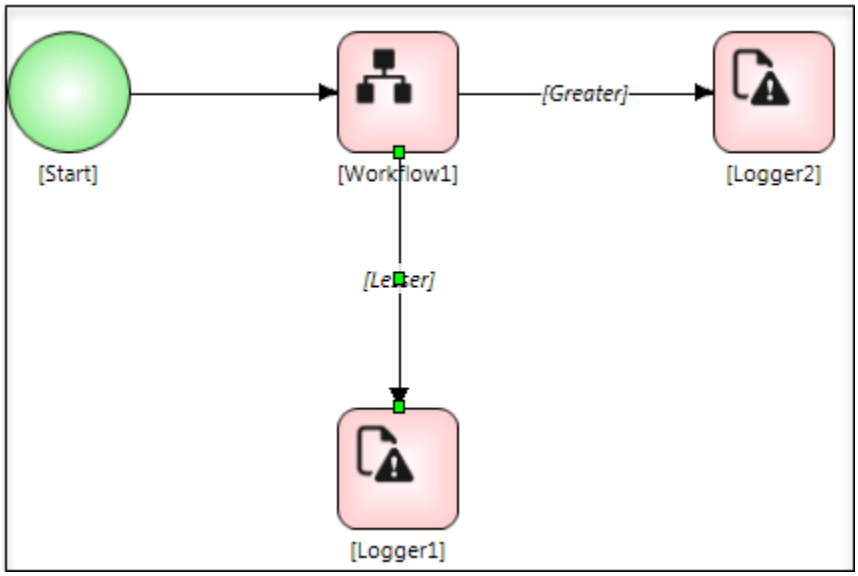


© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

f. On the link connecting Rule and SetWorkflowOutputVariable1 Activity set FALSE.

g. In the Set workflow output property of SetWorkflowOutputVariable1 Activity set it as Lesser as shown in the following figure.

3. Design a workflow as shown in the below figure and name it as ParentWorkflow.



4. Perform the following actions:

1. In the Repository property of Workflow Activity, enter Repository Name. For Example: SamRep.
2. In the Workflow property of Workflow Activity, select Previously created Workflow Name. For Example: ChildWorkflow.
3. In the Wait for the Child Workflow property, select "YES".
4. In the Execution Mode property, select "Asynchronous".
5. In the Return Values property of Workflow Activity:
   - Click Add Row and enter as "Greater"
   - Click Add Row and enter as "Lesser"
   - Click Update to save and click Close to close the window.
6. In the property LogType of Logger1, select Information.
7. In the Log Message of Logger1, enter as the following:

   '################ Lesser'
8. In the link connecting Logger1 and Workflow Activity1 set Lesser as output.
9. In the property LogType of Logger2, select Information.
10. In the Log Message of Logger2, enter as the following:

    '################ Greater'
11. In the link connecting Logger2 and Workflow Activity1, set Greater as output.

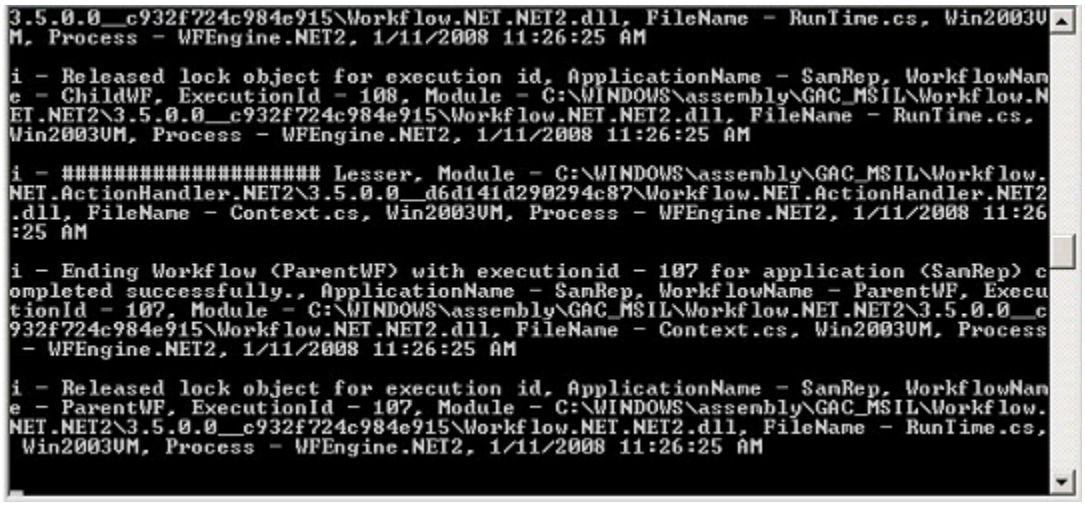
Summary : When you publish the parent workflow, the child workflow is also triggered and the path selected after the Rule activity is stored in the variable "ParentWorklfowOutput". The output of child workflow can be traced in the parent workflow.

Output: The ParentWF is published, and the workflow1 Activity invokes the ChildWF. Value of the variable Value

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

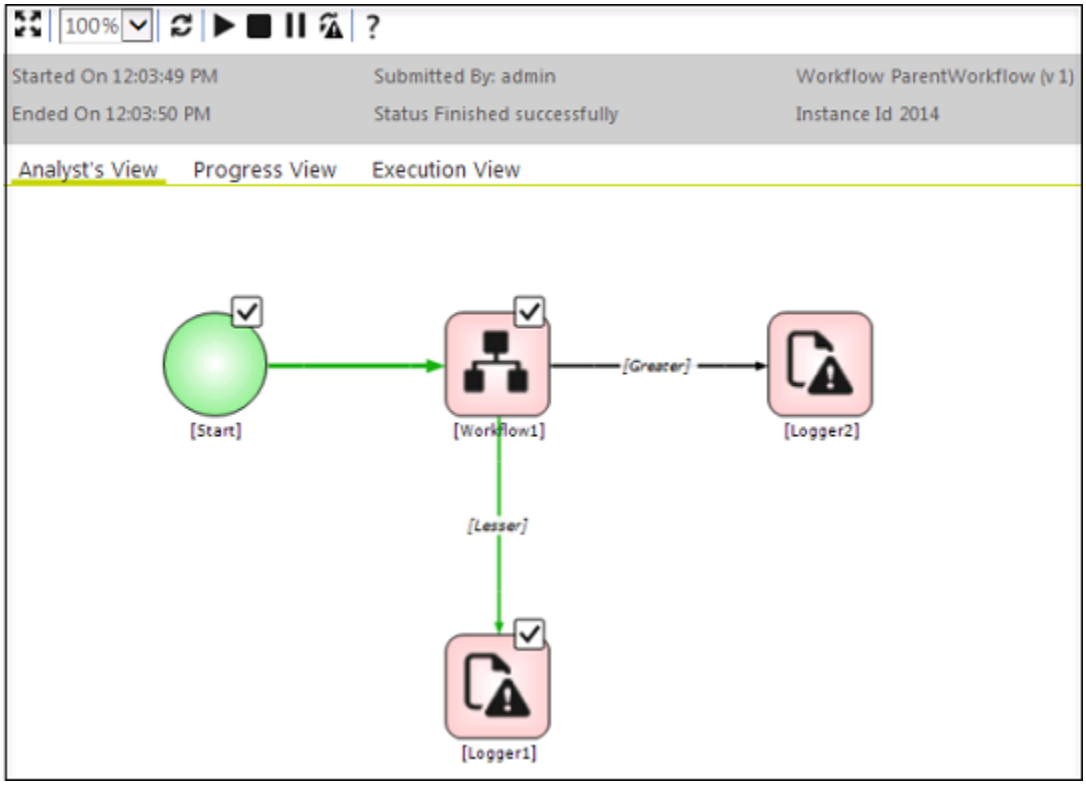
is set to 1 in Update Variable Activity.

Since the value of Variable value is less then five, the rule activity output is False. so the Set Workflow Output Variable2 is executed and the control is passed to the ParentWF. The value updated in Parent Workflow variable is Lesser and hence the output of the Workflow Activity is Lesser and the Logger1 is executed.



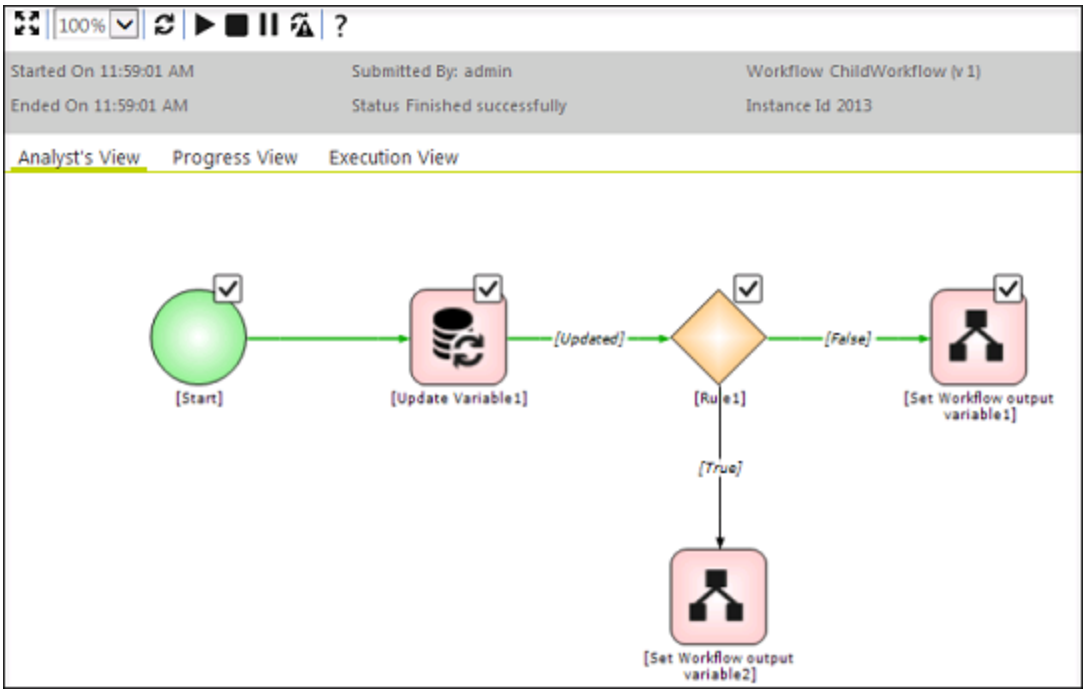
Start Activity in BAM

Parent Workflow in BAM:



Child Workflow in BAM

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



XML Navigator

Activity description

The XML Navigator is used to either locate or navigate to a specific record in a collection within an XMLIterator Variable. This activity enables identification of a specific record for execution of an activity or group of activities.

See also *Example of XML Navigator Activity*

Activity Properties

The properties of the XML Navigator activity can be configured in the Activity Properties window for that activity.

To access Activity Properties, do one of the following:

- Click Activity Properties in the Process Designer tool bar
- Right-click and select Activity Properties

Activity Properties for this activity are organized under:

Name & Description

Specify the name and description for the activity in this group.

Name - Enter the name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - Enter a brief description for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration

The Configuration group has properties for selection of the XMLIterator Variable to be looped through.

XMLIterator - Select the XML Iterator variable to be navigated. All XMLIterator variables as declared in Start activity are available for selection.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Create New Node - This field defaults to No. Select Yes to create a new blank node at the specified location.

Property Type: Optional (This property needs to be set only if necessary.)

Location - This field identifies the location to be navigated to. It defaults to First. The options are:

First: Navigates to the first record in the collection.

Last: Navigates to the last record in the collection.

Next: Navigates to the next record based on the current location.

Previous: Navigates to the previous record based on current location.

Index: Navigates to the record identified in the Index field.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Index - Enter the index number of the record to navigate to. This field is mandatory if the Location field is set to Index.

Property Type: Optional (This property needs to be set only if necessary.)

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs

The XML Navigator Activity returns the following values -

No more records: Indicates that all records have been looped through

Record Found: Indicates that a record is found in the collection

Index out of bounds: Indicates that the index entry specified in the location does not exist

Example for XML Navigator

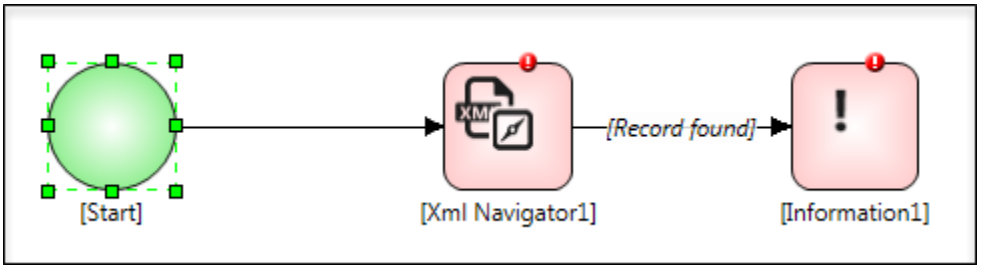
Consider already an XML file called Sample.XML is exists. The contents of it are as shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<root>
<Item>
<itemdID>I001</itemdID>
<desc>some stuff1</desc>
<qty>10.20</qty>
</Item>
<Item>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

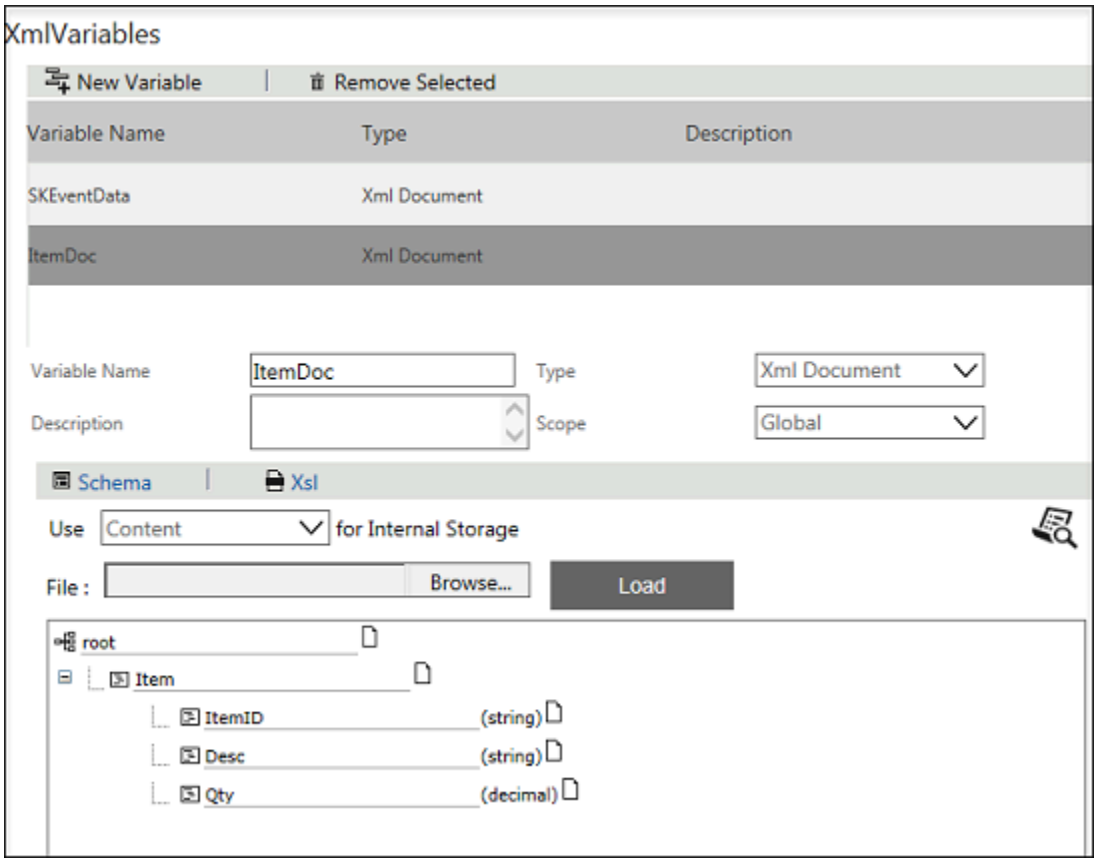
```
<itemdID>I002</itemdID>
<desc>some other stuff2</desc>
<qty>20.20</qty>
</Item>
<Item>
<itemdID>I003</itemdID>
<desc>some more stuff3</desc>
<qty>30.20</qty>
</Item>
</root>
```

Step 1: Design a Workflow as shown in the below figure.



Step 2: In the Start Activity, declare two XML Variables.

1. Click the New Variable link. Under Schema enter the following details:
   Variable Name: ItemDoc, Type: XML Document and Scope: Global
   Use Content for internal storage and browse the above created Sample.XML file and load it.

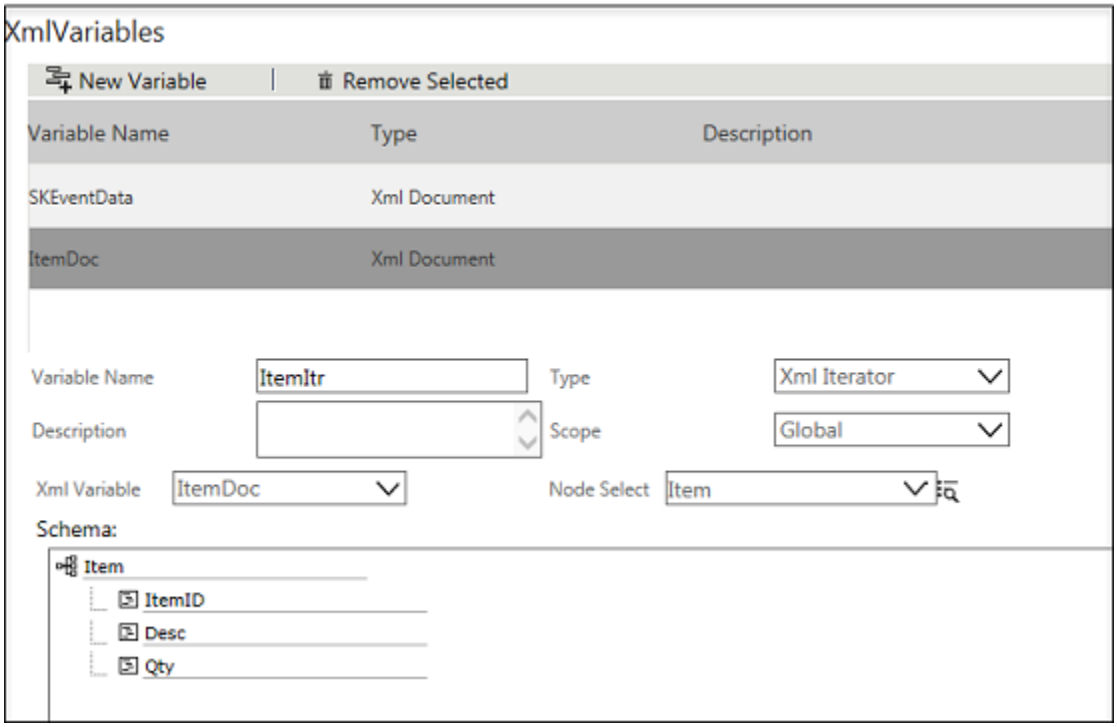


© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Click the **New Variable** link. Under Schema, enter the following details:

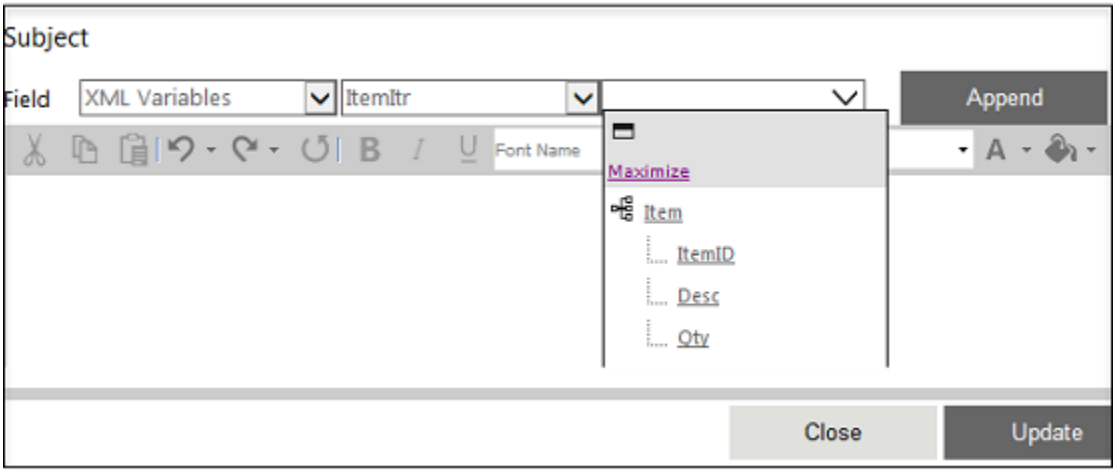
   Variable Name: ItemItr,
   Type: XML Iterator,
   Scope: Global,
   XML Variable: ItemDoc
   Node Select: Item

3. Click **Save Variable** and close the window



Step 3: In the location property of XML Navigator Activity, select "Last". For the XML Iterator property, select the Iterator created that is, ItemItr. In the Information Activity, the ItemItr is displayed in Subject column of the work item.

Step 4: Click the Subject property of Information activity, following window will open.



Select XML Variable form the first drop-down. In the second drop-down, select the Iterator created, that is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

ItemItr. In the third drop-down, select the Item ID from the tree displayed. Click **Append**, and then click **Update**.

Step 5: On the link connecting XML Navigator Activity and Information Activity, select "Record Found".

Output

The item ID will be displayed in the Subject Column of the Information in work item.

Scheduler Activities

The following activities are found under Scheduler Activities:

- **Set Custom Timer Activity**: The Set Custom Timer Activity can be used to schedule the execution of multiple tasks at specified times, set using the corresponding workflow variable.
- **Control Custom Timer Activity**: The Control Custom Timer Activity can be used to control a Set Custom Timer activity in the workflow dynamically.
- **Scheduler Activity**: The Scheduler Activity can be used to schedule a task at a specified time, set on the basis of weekly, daily as well as hourly schedules.
- **Time Trigger Activity**: The Time Trigger Activity can be used to put on hold workflow execution.

Control Custom Timer Activity

Activity Description

The Control Custom Timer Activity can be used to control a Set Custom Timer activity in the workflow dynamically. There can be multiple Control Custom Timer activities controlling a single Set Custom Timer activity at different stages of the workflow process.

Action Properties

The Control Custom Timer activity has to be configured by specifying appropriate values for the different properties in the Action Properties area. The Action Properties can be accessed by clicking on the Action Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Control Custom Timer activity. The Action Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the

functionality of the activity.

Action Name - This property allows the user to specify the name of the Set Custom Timer activity to be controlled.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Action Output - This property allows the user to specify the output of the Control Custom Timer activity. The drop down has three options – Pause, Resume, and Stop – to respectively pause, resume or stop the Set Custom Timer activity being controlled.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Note: When you pause an activity, the activity is paused, but the status of the activity does not change.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Control Custom Timer Activity returns the following values -

Successful: Displayed when the activity is successful in controlling the specified Set Custom Timer activity.

UnSuccessful: Displayed when the activity is unsuccessful in controlling the specified Set Custom Timer activity.

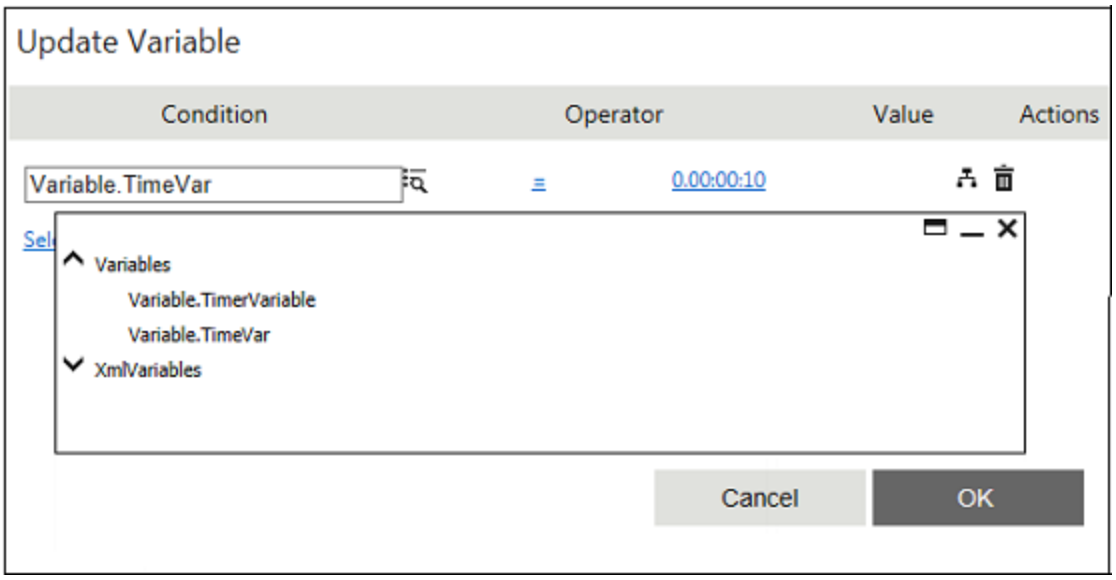
Example:

To control a Set Custom Timer activity in the workflow (See the workflow given below)

Step 1: In Start, declare "timevar" as String type.

Step 2: In **Update Variable**, assign specific value to "timevar" string.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

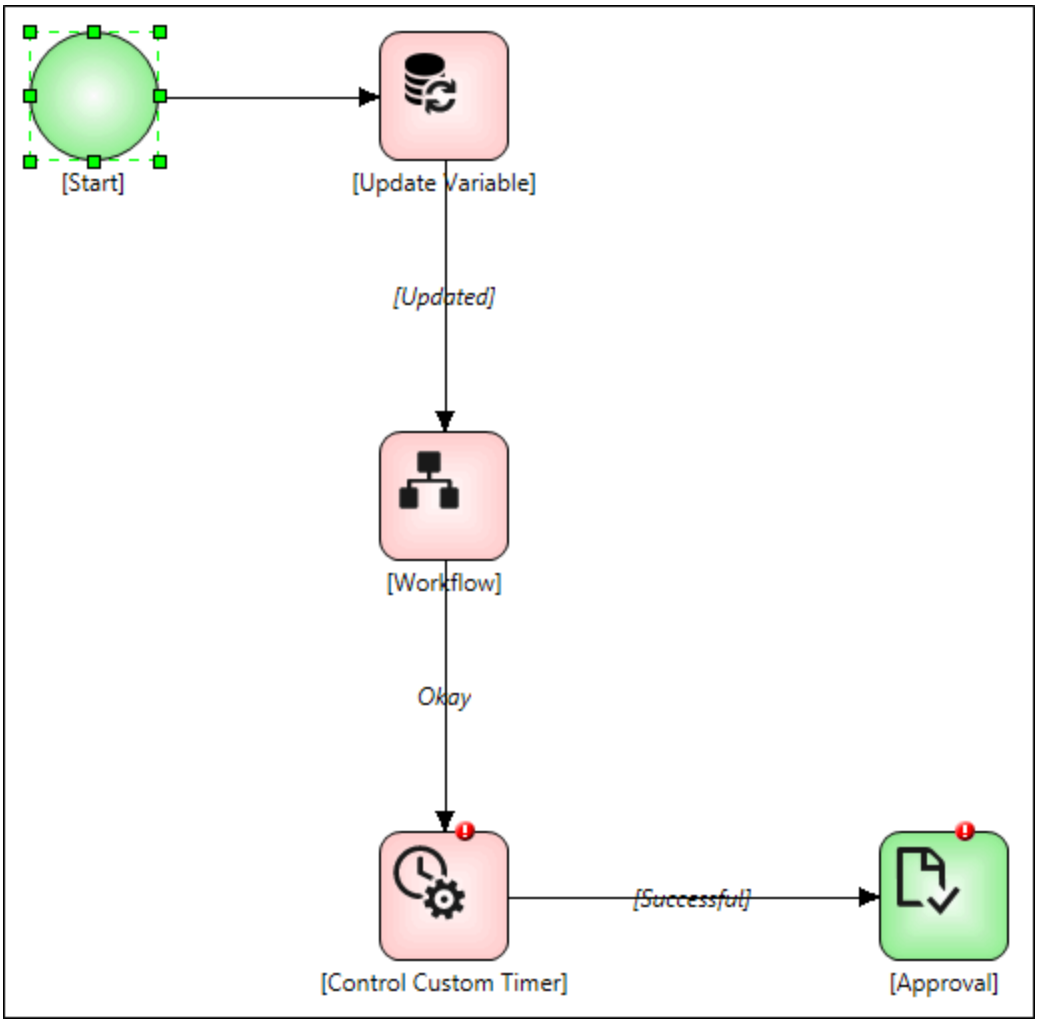


Step 3: In **Set Customer Timer activity** used to **schedule** the execution of tasks at specified times, set using the corresponding workflow variable.

Step 4: In **ControlCustomTimer** you can set the name of the **Set Custom Timer** activity to be controlled in Action Name (Here SkeltaWF is Set Custom Timer) and in Action Output you can Resume the workflow.

Output

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Note: The **Pause** and **Resume** functionality of Workflow is not supported in this activity.

Set Custom Timer Activity

Activity Description

The Set Custom Timer Activity can be used to schedule the execution of multiple tasks at specified times, set using the corresponding workflow variable. This workflow variable with the format '^[CustomTimerSetActionName]ExecutionDetailsId', is to be defined in the Start activity for each Set Custom Timer activity used in the workflow. The scheduling can be made recurring or non-recurring. The output of the activity can be linked to the next activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

The Set Custom Timer activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the tool bar or by selecting the appropriate option from the right-click menu for the Set Custom Timer activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Schedule Time - The Schedule Time property allows the user to specify the time setting for scheduling the specific workflow action.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the Schedule Time property

1. To specify the schedule time, click the button for the 'Schedule Time' property to open the property window.
2. In this window, select the variable with the schedule time setting from the **Variable Description** drop down.
3. If the schedule is to recur, specify "Yes" in the **Recurring?** drop down.
4. Specify the output of this activity in the **Output** field.
5. After entering the details, click **Add Row** to add this schedule time setting and display a new row.
6. Remove rows if necessary by selecting them and clicking the **Remove Selected** button.
7. After setting the XML data to the variables, click **Update**.

See Set Custom Timer: Schedule Time for a detailed description of the Schedule Time property window.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity outputs:

The Set Custom Timer Activity returns the following values -

Paused: Displayed when the controlled activity is paused

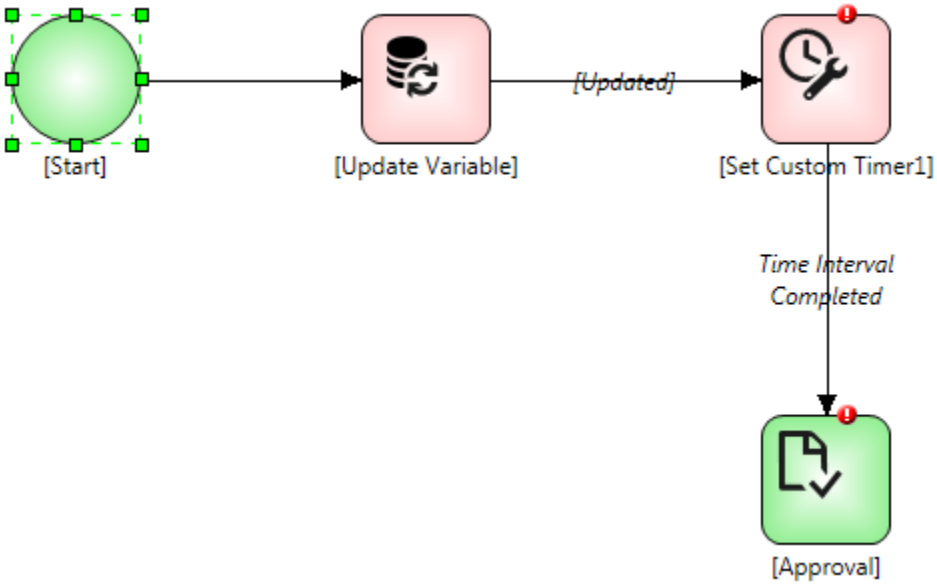
Resumed: Displayed when the controlled activity is resumed

Stopped: Displayed when the controlled activity is stopped

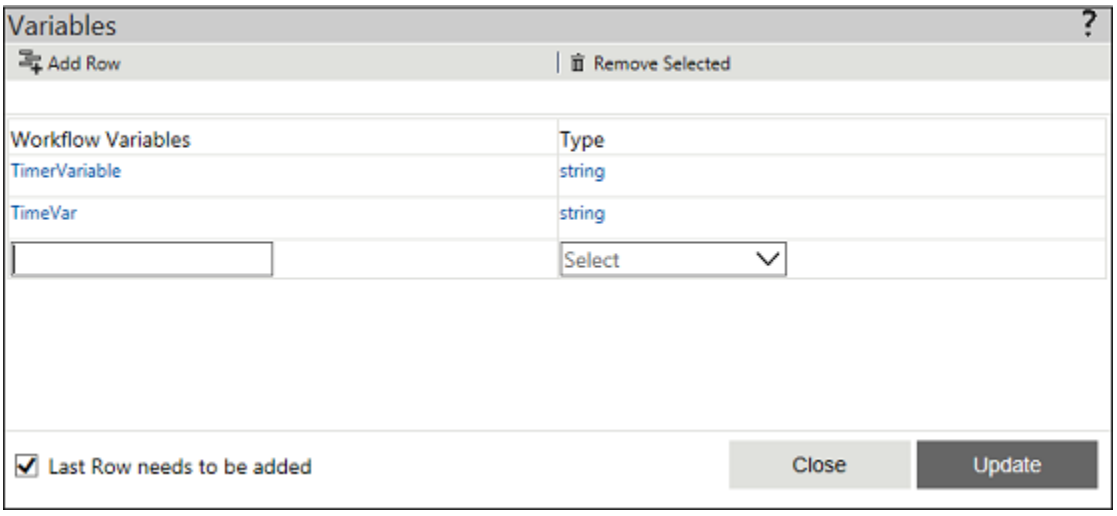
In addition to these standard return values it also returns the scheduled time as output.

Example for a setting Custom Timer Activity:

Step 1: Design a workflow as shown below.



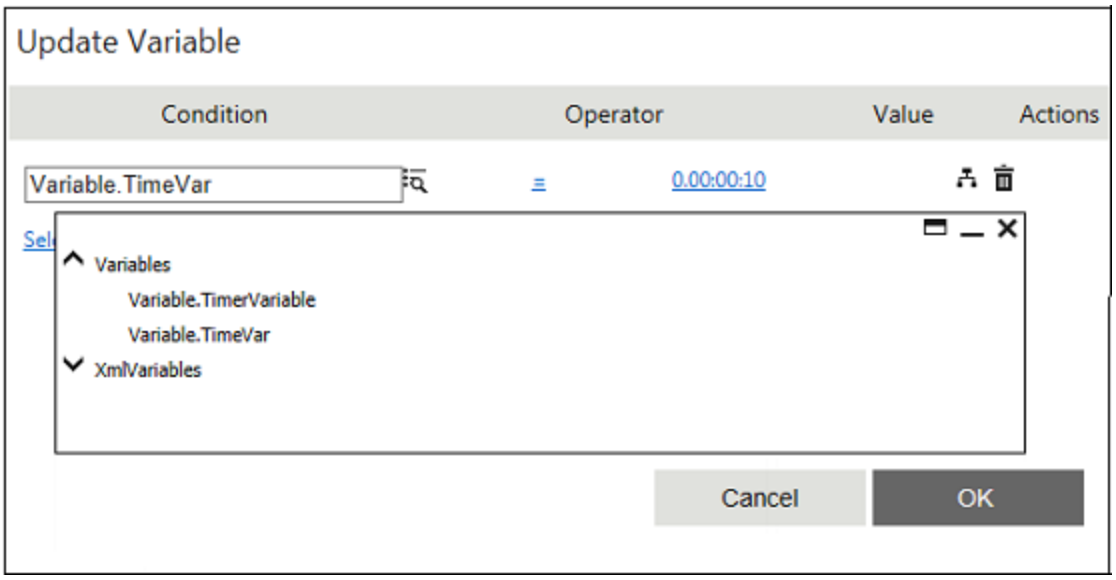
Step 2: Declare a variable of type String in the Start activity. For example, TimerVariable as shown in the below figure.



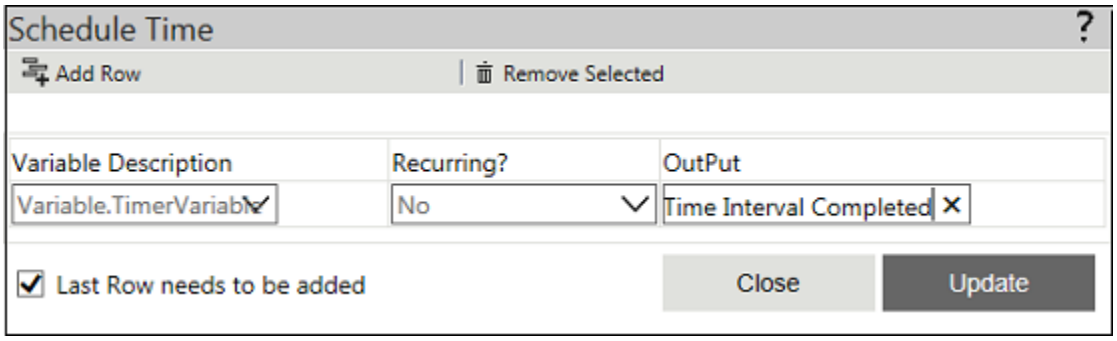
Step 3: This variable is updated using the engine activity "Update Variable ".

For example, Timer Variable = 0.00:00:10 means 10 seconds.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Step 4: On successful updating of the variable, set custom timer activity.



When there is a schedule time over the information is sent to the assigned actor.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details

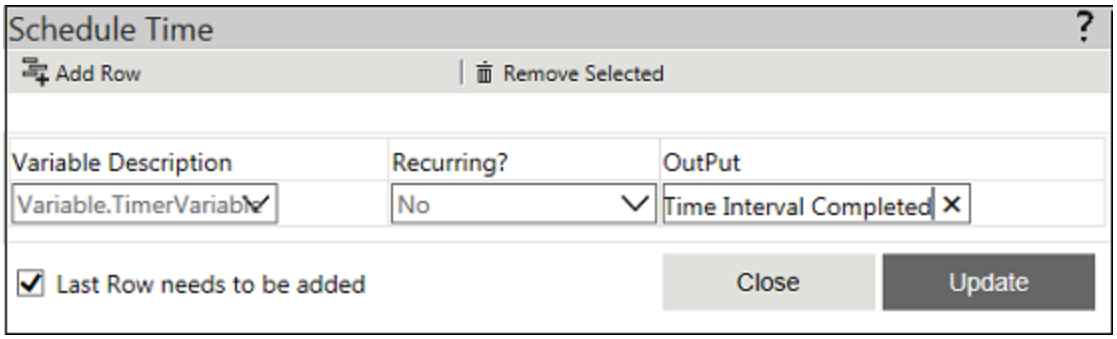
Note: The **Pause** and **Resume** functionality of Workflow is not supported in this activity.

Set Custom Timer: Schedule Time

Schedule Time Property window

You can set the schedule time property for the Set Custom Timer activity, used to execute some other activity, in this property window. To open this window, click the button for the 'Schedule Time' property. The time is assigned to workflow variables in this window. It displays the settings for the variables in a grid, which also has a blank row where you can enter the settings for a new variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Schedule Time property

The window has two tabs at the top to manage the rows of schedule settings in the grid.

- **Add Row** - You can click this tab to add the newly entered schedule settings and display a new row to the grid where you can add another schedule setting
- **Remove Selected** - You can click this tab to remove the schedule setting selected in the grid.

The grid has the following parameters that you need to specify for the new schedule.

- **Variable Description** - You can select the workflow variable, which holds the schedule time for the execution of the specified workflow activity, from the drop down for this field.
- **Recurring?** - You can select "Yes" in this drop down to indicate that the schedule is recurring.
- **Output** - You can enter the output message for this activity in this field.

There are two buttons and a check box at the bottom of the screen.

- **Update** - You can click this button to save the newly added schedule settings.
- **Close** - You can click this to close the property window without saving the newly added schedule settings.
- **Last Row needs to be added** - You need to select this check box to save the last row added.

Scheduler Activity

Activity Description

The Scheduler Activity can be used to schedule a task at a specified time, set on the basis of weekly, daily as well as hourly schedules. The output of the activity can be linked to the next activity. There is an option to set the recurrence count so that the user can restrict the number of recurrences.

Activity Properties

The Scheduler activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the tool bar or by selecting the appropriate option from the right click menu for the Scheduler activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Scheduler Details - The Scheduler Details property allows the user to specify the time settings for scheduling the specific workflow activity.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Scheduler Details property

1. To specify the Scheduler Details, click the button for the 'Scheduler Details' property to open the property window.
2. Click on one of the three tabs - **Weekly**, **Time Span**, and **Date Criteria** - to set the corresponding property value.
3. By default, the first tab, i.e. **Weekly**, is selected and the corresponding user interface is shown. You can specify the weekly schedule settings here.
4. Select the day of the week on which the activity is to be scheduled from the Time drop down.
5. Next specify the time at which the scheduled activity should be executed on the selected day in the **hh:mm:ss** field in the "hh:mm:ss" format.
6. Now, click the **Recurring?** check box to schedule the activity to recur.
7. If it is to recur more than once, specify the count in the **Recurrence Factor/Count** field.
   - If the count is set to '0' then the activity keeps on recurring.
   - If it is set to '2' the activity ends after the second recurrence.
   - If it is set to '1' or 'NO' the activity recurs only once.
8. If the **Time Span** tab is clicked, it displays the corresponding user interface. You can specify a time span for the execution of the activity here.
9. Enter the time interval of the schedule in the first field, in the "d.hh:mm:ss" format, where "d" denotes the number of days the schedule is set for while the time setting denotes the number of hours, minutes, or seconds. For example, if this value is set as 0.00:30:00, the activity will be executed every 30 minutes. If it is set as 1.00:30:00, the activity will be executed every 1 day and 30 minutes or 24 hours and 30 minutes.
10. Click the **Recurring?** check box to schedule the activity to recur.
11. If it is to recur more than once, specify the count in the **Recurrence Factor/Count** field.
12. If the **Date Criteria** tab is clicked, it displays the corresponding user interface. You can specify a date based schedule here.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

13. Here, the date value can be selected using Content or Variable collection or by entering discrete values.
14. After specifying the value you must select one of two options - "Initiate before" or "Initiate after" - from the next drop down and then specify the time interval in number of days in the field provided.
15. After making all the schedule settings, click **Update** to save the settings.

See Scheduler: Scheduler Details for a detailed description of the Scheduler Details property window.

Output Path - The Output Path property allows the user to specify the output path that can be used to link to the next activity, and can be treated as the output of the Scheduler activity.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

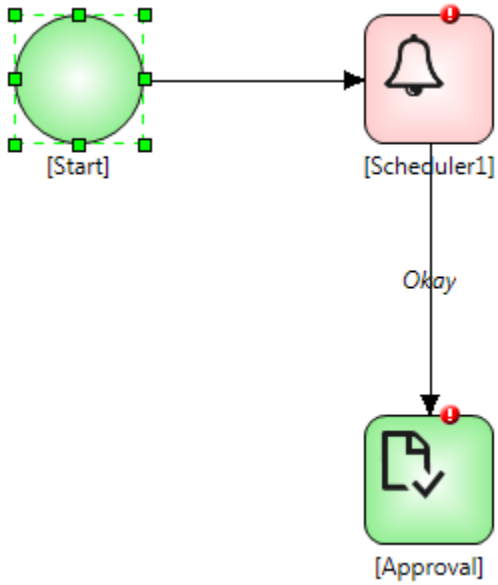
Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Scheduler Activity does not have any standard return value. It will output the value specified by the user in the Output Path property.

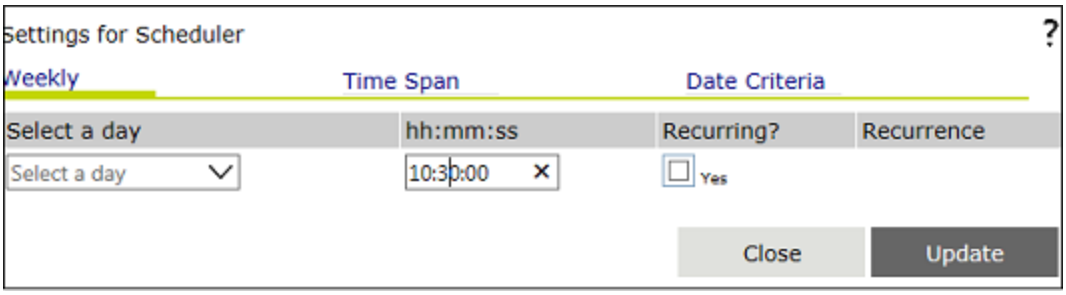
Example for scheduling an activity on every Monday at 10.30 a.m:

Step 1: Design a workflow as shown in figure.



Step 2: Click the 'Scheduler Details' property and select Scheduler Details, following image appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Step 3: Select a day and set the time and click on the **Update** button to save the settings.

Step 4: Set the value for Output Path.

Step 5: Set Actor for approval.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

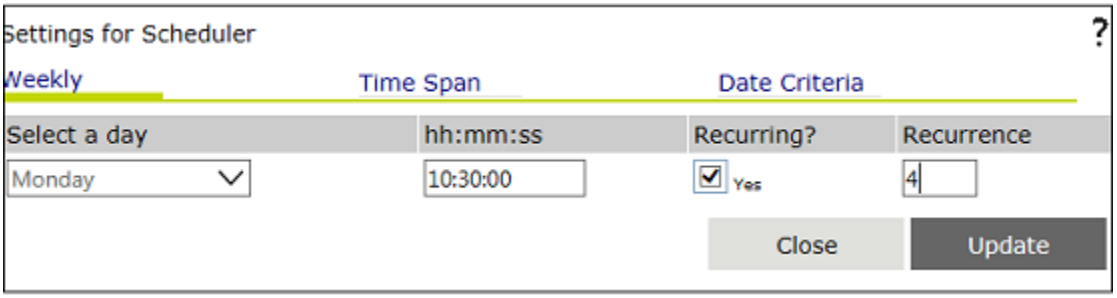
Note: The **Pause** and **Resume** functionality of Workflow is not supported in this activity.

Scheduler: Scheduler Details

You can set the Scheduler Details settings in this property window. To open this window, click the button for the 'Scheduler Details' property. It has three tabs - **Weekly**, **Time Span**, and **Date Criteria** - to set the corresponding property value.

Weekly settings

By default, the first tab, i.e. **Weekly**, is selected. The user interface corresponding to this tab is shown in the image below. You can specify the weekly schedule settings here.



Weekly Settings for Scheduler Details property

This window has the following parameters for setting the weekly schedule.

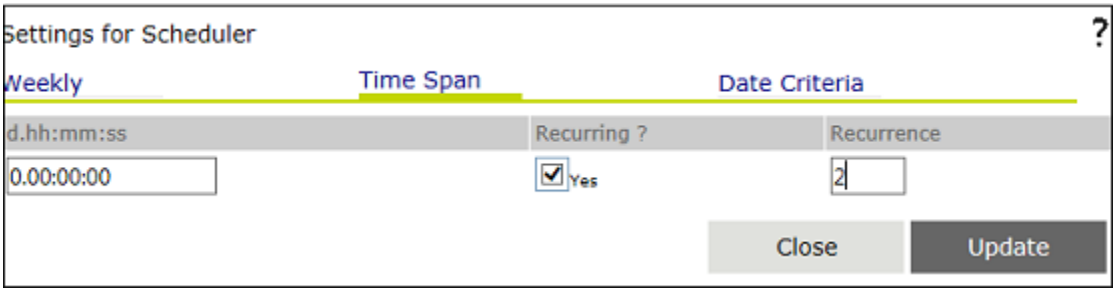
- **Select a day** - You can select the day of the week on which the activity is to be scheduled from the drop down for this field.
- **hh:mm:ss** - You can specify the time at which the scheduled activity should be executed on the selected day in this field in the "hh:mm:ss" format.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Recurring ?** - You can select this check box to schedule the activity to recur. This will enable the next field.
- **Recurrence** - If the schedule is to recur more than once, you must specify the count in this field.

Time Span settings

If the **Time Span** tab is clicked, it displays the user interface shown in the image below. You can specify a time span for the schedule here.



Time Span Settings for Scheduler Details property

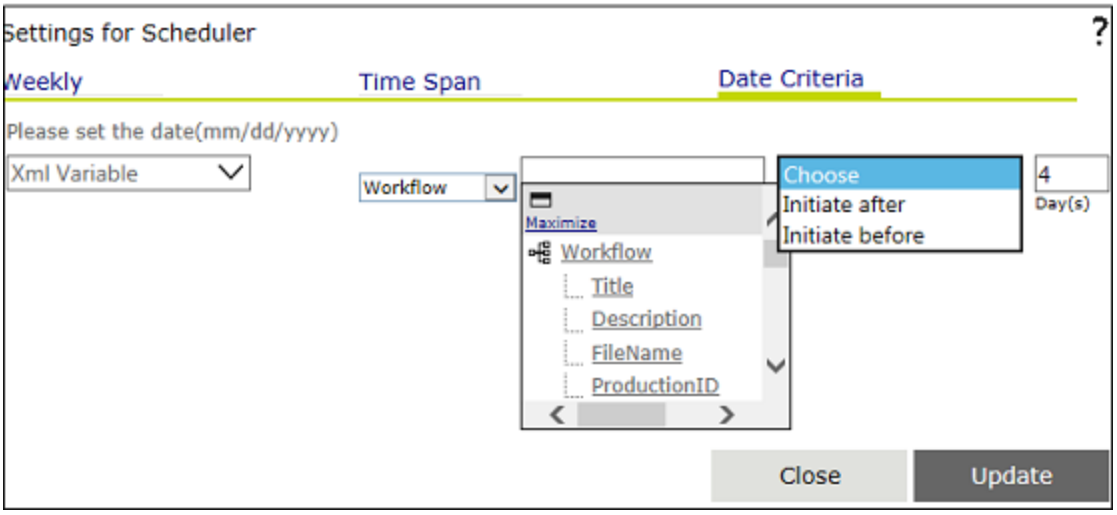
This window has the following parameters for setting the time span schedule.

- **d.hh:mm:ss** - You can enter the time interval of the schedule in this field, in the "d.hh:mm:ss" format, where "d" denotes the number of days the schedule is set for while the time setting denotes the number of hours, minutes, or seconds. For example, if this value is set as 0.00:30:00, the activity will be executed every 30 minutes. If it is set as 1.00:30:00, the activity will be executed every 1 day and 30 minutes or 24 hours and 30 minutes.
- **Recurring?** - You can select this check box to schedule the activity to recur. This will enable the next field.
- **Recurrence** - If the schedule is to recur more than once, you must specify the count in this field.

Date Criteria settings

If the **Date Criteria** tab is clicked, it displays the user interface shown in the image below. Here, you can specify a date-based schedule.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



This window has the following controls for setting the date criteria schedule.

- The XML Variable drop down - You can select the XML variable from the list.
- The XVAR drop down - You can select the date value from this list.
- The schedule criteria drop down - You can select one of two options in this drop down - "Initiate before" or "Initiate after".
- The time interval field - You can specify the time interval in number of days in this field.

Update - Click this option to save the newly added schedule settings.

Close - Click this option to close the property window without saving the newly added schedule settings.

Reschedule Failed Tasks

You can reschedule failed tasks using the taskscheduler service. By default, this feature is disabled. You need to manually enable this feature by modifying the farm database, as per configuration shown below:

Open the farm database and edit the SKService table.

Update the Properties field of the record corresponding to the task scheduler with the code shown below:

Select Properties from SKService where Name='SkeltaTaskScheduler2'

```
<properties>
<property key="loadbalanced"> <![CDATA[False]]></property>
<property key="loadbalanceserverport"><![CDATA[8853]]></property>
<property key="loadbalanceclientport"><![CDATA[8860]]></property>
<property key="port"><![CDATA[8856]]></property>
<property key="fileWatcher"><![CDATA[]]></property>
<property key="maxThreads"><![CDATA[25]]></property>
<property key="serviceexe"><![CDATA[TaskSked.NET2.exe]]></property>
<property key="displayname"><![CDATA[Skelta - Task Scheduler]]></property>
<property key="enableReScheduling"><![CDATA[true]]></property>
<property key="lastUpdatedDateTime"><![CDATA[]]></property>
<property key="retryCount"><![CDATA[]]></property>
<propertykey="noOfrecordsforreshedulingTasks"><![CDATA[]]></property>
<property key="rescheduletimeinterval"><![CDATA[]]></property>
<property key=" repositories"><![CDATA[]]></property>
</properties>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration details

- enableReScheduling : If this property key is missing or if it is set to False, the rescheduling feature is disabled. By default, this property will not be added. You need to manually add this property and set it to True to reschedule the failed tasks.
- LastUpdatedDateTime: If this property is configured, only the failed tasks whose LastUpdatedDateTime is greater than configured are fetched for rescheduling. By default this value is null and all the failed tasks will fetched if rescheduling is enabled.
- retryCount: This property sets the number of times that the failed tasks should be rescheduled. The default value is three.
- noOfrecordsforreshedulingTasks: This property sets the total number of failed records to be fetched for rescheduling. The default value is 15.
- rescheduletimeinterval: This property sets the time period for which the task scheduler service tries to fetch and reschedule the failed tasks. The default value is 10 minutes.

  Note: 0.00:20:00 indicates that the interval is configured for 20 minutes. This value should always be greater than the fetch interval time.
- repositories: This property performs rescheduling only for the specific repositories. By default, this property value is empty, and all the repositories will be considered for rescheduling if this feature is enabled.

Time Trigger Activity

Activity Description:

The Time Trigger Activity can be used to put on hold workflow execution. It is useful in ensuring the execution of certain activities at specific times. Conditional routing can be performed for different timeouts. Multiple non-recurring timeouts can be scheduled using this activity.

Activity Properties:

The Time Trigger activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the tool bar or by selecting the appropriate option from the right click menu for the Time Trigger activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Timer Trigger - This property allows the user to specify the timeout(s) for triggering activities in the workflow. This will induce a delay in the execution of the affected activities.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Timer Trigger property

1. To specify the timer trigger, click on the button for the 'Timer Trigger' property to open the property window.
2. In this window, enter the time interval for the timer trigger in the **Time Interval** field.
3. Specify the output of this activity in the **Output** field.
4. After entering the details, click **Add Row** to add this timer trigger setting and display a new row.
5. Remove rows if necessary by selecting them and clicking **Remove Selected**.
6. After setting the timer trigger settings, click **Update**.

See Time Trigger: Timer Trigger for a detailed description of the Timer Trigger property window.

Activity outputs:

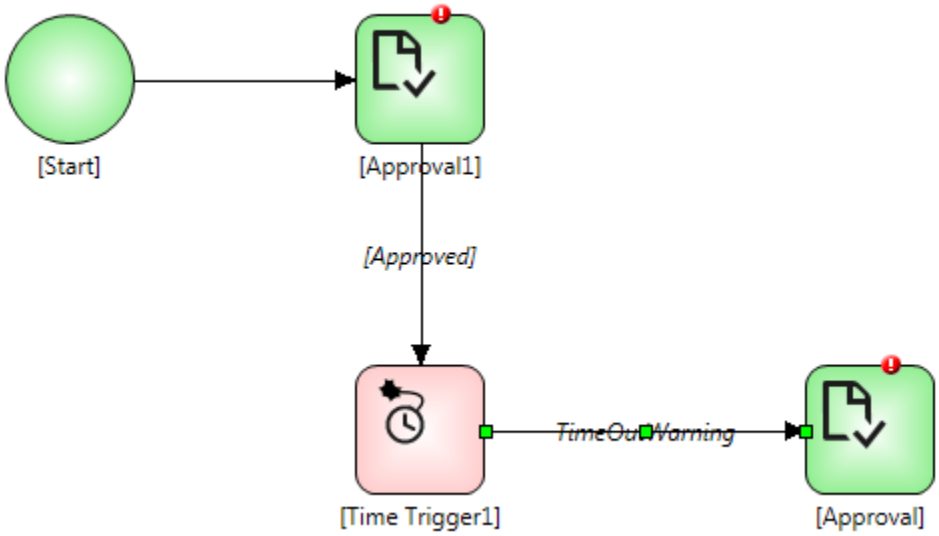
The Time Trigger Activity does not have any standard return value. It will output the value specified by the user in the **Time**

Trigger property.

Example for Time Trigger Activity

To hold workflow execution from Approver1 to Approver2, Use Time Trigger Activity for particular time.

Step 1: Design a workflow as shown in the figure below.



Step 2: Select the Time Trigger Property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Step 3: Set the Time Interval and specify the output value in output field.

Step 4: Click Update button to save changes.

Exception Behavior

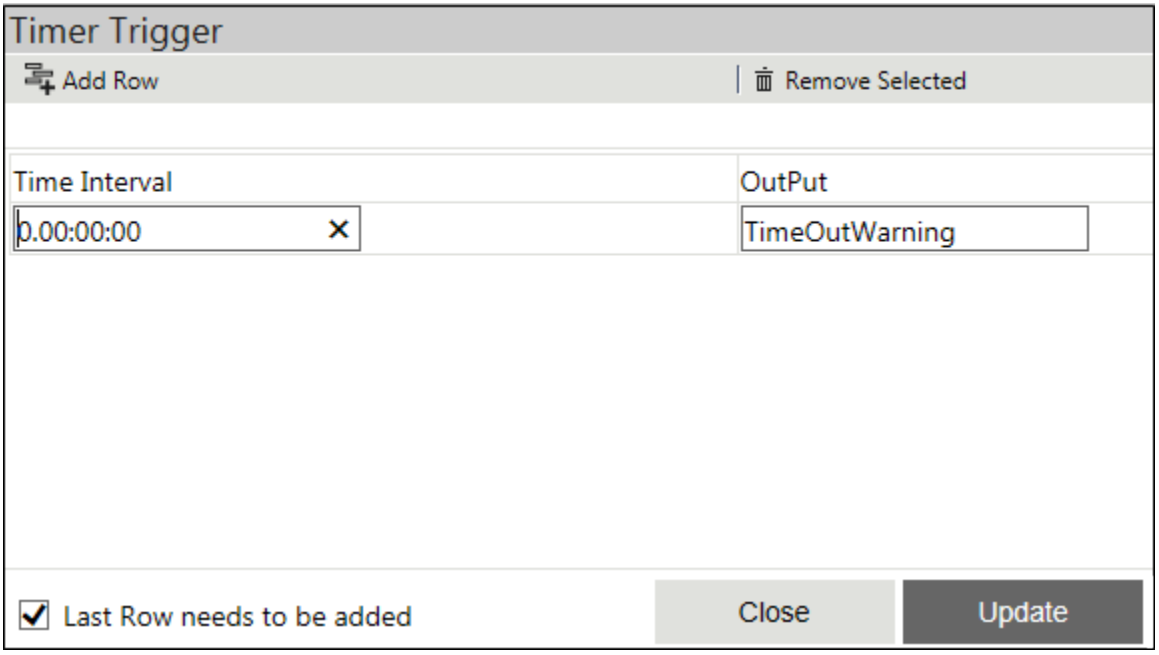
Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Note: The **Pause** and **Resume** functionality of Workflow is not supported in this activity.

Time Trigger: Timer Trigger

You can set the Timer Trigger settings in this property window. To open this window, click the button for the 'Timer Trigger' property. It displays the timer settings in a grid, which also has a blank row where you can enter the settings for new timer trigger.



The window has two tabs at the top to manage the rows of timer trigger settings in the grid.

- **Add Row** - You can click this tab to add the newly entered settings and display a new row to the grid where you can add another setting.
- **Remove Selected** - You can click this tab to remove the timer trigger settings selected in the grid.

The grid has the following parameters that you need to specify for the new timer trigger setting.

- **Time Interval** - You can use this field to set the time interval for the timer trigger.
- **Output** - You can enter the output message for this activity in this field.

There are two buttons and a check box at the bottom of the screen.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Update** - You can click this button to save the newly added timer trigger settings
- **Close** - You can click this to close the property window without saving the newly added timer trigger settings
- **Last Row needs to be added** - You need to select this check box to save the last row added.

Communication Activities

There are two main types of Communication activities:

- Notification Activity: The Notification Activity is incorporated in the workflow process design to send notifications to actors in the form of email.
- AVEVA Work Tasks Pro Push Notification Activity: The AVEVA Work Tasks Pro Push Notification activity is used to send notifications to the AVEVA Work Tasks Pro application.

Email Activity

Activity description

The Email Activity can be used to send an email to a selected set of actors. The actors(s) can be selected by building a query. The workflow engine evaluates this query and sends an email to all the actors resulting from the query. If no actor is selected the email will be sent to all the actors. The subject and body of the email can be customized by appending the resource properties available.

Adding Email attachments using Contents and Variables:

Steps to add Email attachments using Workflow Variables:

To set email attachments using variable, prefix the variable name with "^[ActivityName]".

For example, for the activity named "EmailNotification", name the variable as "^EmailNotificationattachments". Set the value for the variable with physical paths to the file. Use semi-colon as separator to set multiple files.

Steps to add Email attachments using Contents:

To send file attachments with the email, add a content with a name prefixed by the activity name followed by attachments. For example, if a file attachment has to be sent along with an email from an activity named 'EmailNotification', add a content with 'EmailNotificationattachments' as its name. The XPath expression should select the nodes which contain the path for the files to be attached. For example, //fileattachments.

Note:
- Supported file types for attachment include .au, .avi, .bas, .bmp, .cda, .cdr, .chf, .dic, .doc, .docx, .dot, .drw, .dwg, .dxf, .eml, .eps, .gif, .gz, .htm, .html, .ico, .img, .jpeg, .jpg, .ldb, .log, .mid, .midi, .mov, .movie, .mp1, .mp2, .mp3, .mpeg, .mpg, .msg, .nws, .obd, .oft, .pcl, .pcx, .pdd, .pdf, .pic, .pot, .pps, .ppt, .pptx, .pub, .qbb, .qbw, .qdb, .ra, .shtml, .snd, .text, .tga, .tif, .tsv, .ttf, .txt, .uu, .vir, .wb2, .wbk, .wiz, .wk4, .wks, .wpd, .xlk, .xls, .xlsx, and.xlt.

- If the XPath Expression represents an Attachment control node in the Form XML Variable, then the attachment would be added to the Email only if the Attachment value is as per the Attachment control in the form with the properties **Storage Type** as **Database** and **Security Mode** as **Encryption**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

The Email activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Email activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to specify the content of the notification.

To Resource - This property specifies the resource(s) to whom the email is to be sent.

Property Type: Mandatory (This property must be set if the action is to be assigned to any resource(s).)

Steps to set the To property

1. To specify the actors to receive the email, click the button for the 'To' property to open the property window.
2. In this window, build an expression to select the actors using the actors properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Email: ToResource for a detailed description of the To property window.

See the Using XML Variables section in the About XML Variables topic for details on how to use the XMLVariables to build the expression.

To Email Address - This property is used to send an email to a user who is not a part of the resource provider. To send an email at an address that is a part of the resource provider the ToResource property is used.

Scenario

XYZ organization is one of the leading product development companies. Every new feature or product released

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

needs to be notified to all customers. John Smith is one among the them.

The Email activity can be used for this. The **To Email Address** property should be set to John Smith's email address. Whenever the new feature or product is released, the workflow gets trigged to send an email to John Smith.

Features of the To Email Address Property:

The following are the features of the To Email Address property:

- It is possible to send an email at any email address using the Expression Builder or directly entering the email address in the field.
- It is possible to use the variables, XML variables and content, or any expression to set the email address to the property.

Configuration

To configure **To Email Address**, complete the following steps:

1. Right-click **Email Activity** to go to the **Activity Properties**.
2. Set the **To Email Address** property to send email to the resource who is not a part of the resource provider.

Note: If both **ToResource** and **To Email Address** values are set then the **ToResource** property is considered for sending the email.

BCC - This property specifies the resource(s) email id to which a copy of the email is to be sent as a Blind Carbon Copy. To specify the email recipient type the email id directly into the field.

Property Type: Optional (This property needs to be set only if required.)

BCCResource - This property specifies the resource(s) to whom a copy of the email is to be sent as a Blind Carbon Copy. To select the resource(s) click the button against 'BCCResource' in the Properties pane of the Process Designer. In the window that pops up, build the condition for selecting the resource(s) and click Update button.

Property Type: Optional (This property needs to be set only if required.)

Steps to set the BCCResource property

1. To specify the actors who will receive the email as BCC, click the button for the 'BCCResource' property to open the property window.
2. In this window, build an expression to select the actors using the actors properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Email: BCC Resource for a detailed description of the BCCResource property window.

See the Using XML Variables section in the About XML Variables topic for details on how to use the XMLVariables

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

to build the expression.

CC - This property specifies the resource(s) email id to which a copy of the email is to be sent as a Carbon Copy. To specify the email recipient type the email id directly into the field.

Property Type: Optional (This property needs to be set only if required.)

CCResource - This property specifies the resource(s) to whom a copy of the email is to be sent as a Carbon Copy. To select the resource(s) click the button against 'CCResource' in the Properties pane of the Process Designer. In the window that pops up, build the condition for selecting the resource(s) and click Update.

Property Type: Optional (This property needs to be set only if required.)

Steps to set the CCResource property

1. To specify the actors who will receive the email as CC, click the button for the 'CCResource' property to open the property window.
2. In this window, build an expression to select the actors using the actors properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Email: CCResource for a detailed description of the CCResource property window.

See the Using XML Variables section in the About XML Variables topic for details on how to use the XMLVariables to build the expression.

Subject - This property specifies the subject of the email. The subject can be customized for each resource.

Property Type: Optional (This property need not be set. If it is not set, the default subject line from the Actions.xml file is displayed.)

Steps to set the Subject property

1. To specify the subject, click the button for the 'Subject' property to open the property window.
2. In this window, enter the subject text directly in the text editor.
3. If you want to customize the subject you can do so using the Field controls to add resource property values. For example, to add a subject that includes the email id of the actor submitting the approval request, you can do as follows: Type in the text "Expense Statement of ". Select "Variable" and "Variable.SubmittedBy.Email" from the two drop downs for Field and click Append. This will add the placeholder for the variable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the email id.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the subject, click **Update** to save it.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. Click **Close** to close the window without saving the subject.

See Email: Subject for a detailed description of the Subject property window.

Body - This is the actual content of the email sent to the resource(s). The body can be customized for each resource.

Property Type: Mandatory (This property needs to be set.)

Steps to set the Body property

1. To specify the message body, click the button for the 'Body' property to open the property window.
2. In this window, enter the body text directly in the text editor.
3. If you want to personalize the body you can do so using the **Field** controls to add resource property values. For example, to add a body that includes the name of the actor submitting the approval request, you can do as follows: Type in the text "Hello". Select "Variable" and "Variable.SubmittedBy.name" from the two drop downs for **Field** and click **Append**. This will add the placeholder for the variable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the name, resulting in a personalized salutation. Type in a message in the next line to complete the message body.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the body, click **Update** to save it.
7. Click **Close** to close the window without saving the body.

See Email: Body for a detailed description of the Body property window.

Send only one copy to CC/BCC recipients - This property specifies whether to send multiple copies of the Email to CC/BCC recipients. The default setting is 'No'. This property is useful in case there are multiple recipients specified in the 'To' property and only one (or a number less than specified in the 'To' property) recipient in the CC or BCC property. If this properties is set to 'Yes', and there are 5 recipients specified in the 'To' property and one in the CC property, then the CC recipient will receive 5 copies of the same email.

Property Type: Optional (This property needs to be set only if necessary.)

Mail Format - This property specifies the format of the Email. If the value 'HTML' is selected, then the Email will be in the HTML format. If the value 'TEXT' is selected, then the Email will be in plain text format.

Property Type: Optional (This property needs to be set only if necessary.)

Attach Reports

This property is used to configure report to be sent as an email attachment.

Property Type: Optional (This property needs to be set only if necessary.)

The following image shows the Attach Reports property:

Steps to set the Attach Reports Property

1. To specify the **Attach Report** property, click the edit icon to open the property window. The **Attachment Configuration** dialog box appears.
2. In the **Attach Report** property, you can attach a report by selecting the **Report Reference Variable** through the Expression Editor. Alternatively, you can get the Report by clicking the **Report** icon and selecting the required report from the Reports list screen. To attach multiple reports, click **Add** to add new rows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Click **Submit**.

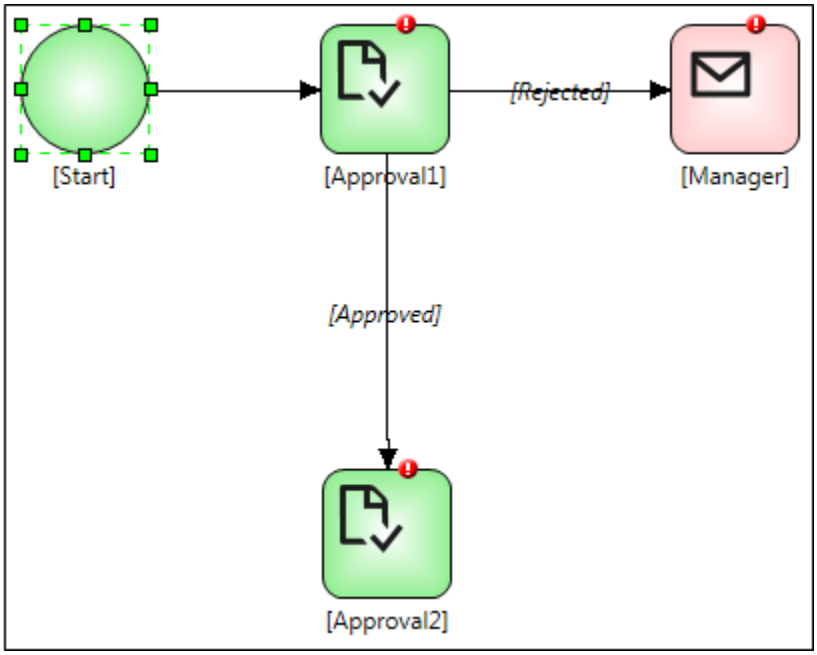
See Email: Attach Reports for a detailed description of this property.

Activity outputs:

The Email Activity does not return any value.

Example for Email Activity

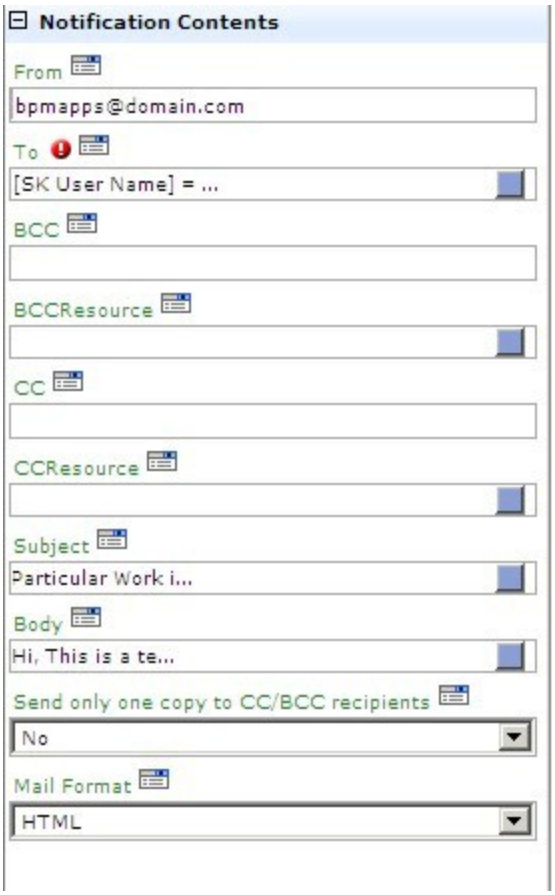
In the following workflow, if a Approver rejects the mail will send to Manager.



Approver 1: If approver1 Rejected a mail, it will send to Manager. If approver1 Approved a mail, it will go for the next Approver, that is Approver2.

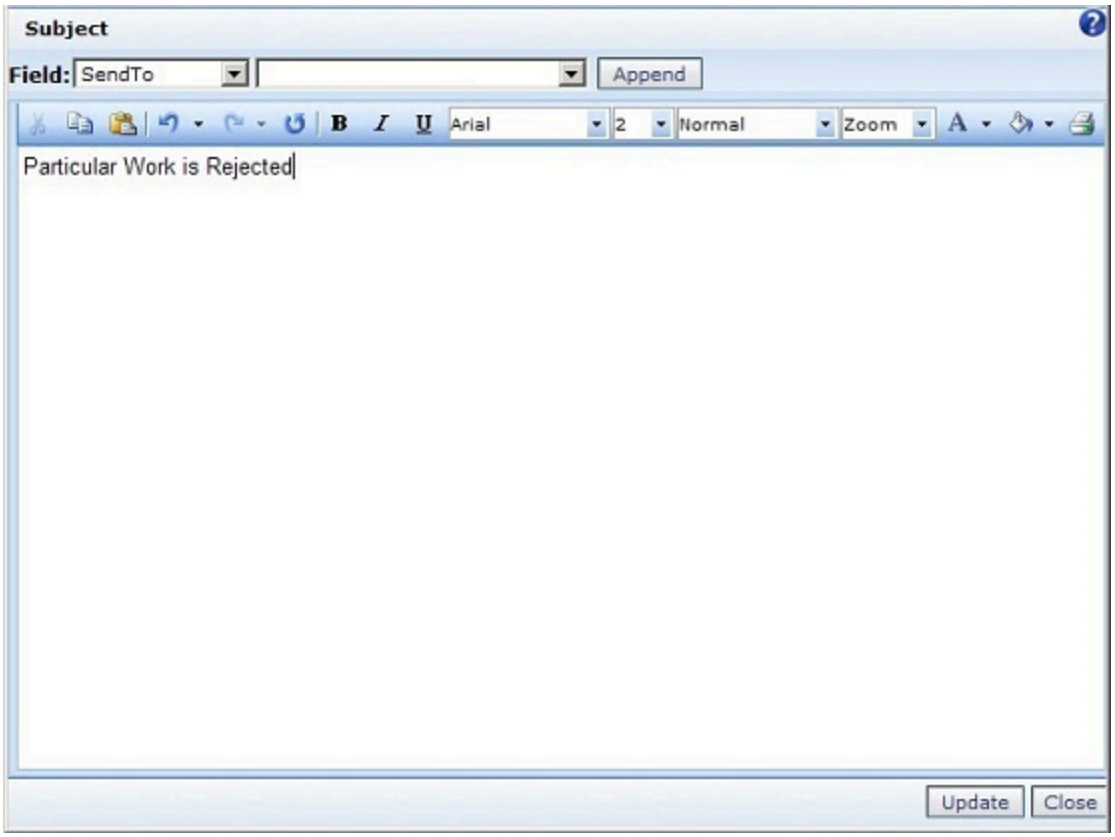
Manager: In Email activity, specifies the resource(s) to whom the email is to be sent in **To** property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

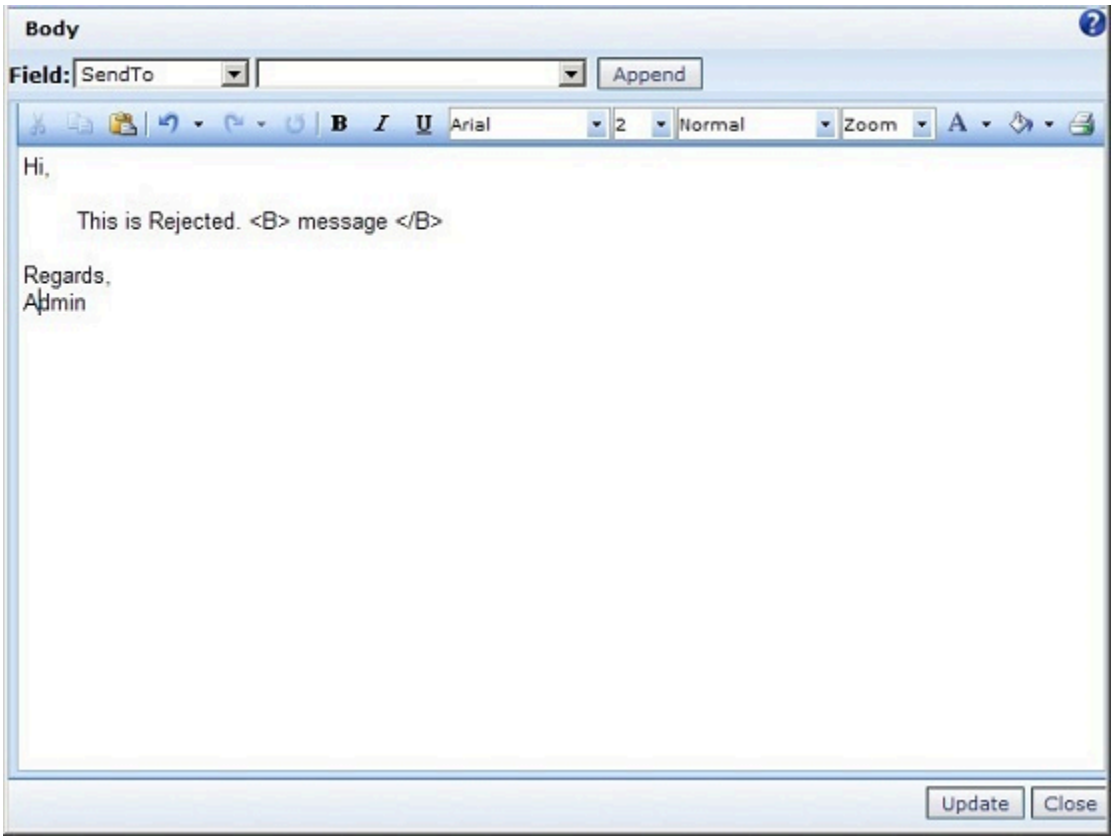


Subject: In this property, specify the subject for the email. After adding the subject, click Update to save it.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Body: This is the actual content of the email sent to the resource(s). After adding the body, click Update to save it.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Email: ToResource

You can set the actors to whom the Email is to be sent in this property window. To open this window, click the button for the 'ToResource' property.

The following image shows the 'ToResource' property dialog box:

The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

Different Methods of Assigning an Actor

- **Assigning an Actor by building an expression:**

  Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

  - The Providers drop-down: You can select the required provider from the Providers drop-down.The Attribute drop down: You can select the attribute to be used to identify the actor. For example, "AdministratorId", in this drop down.The Operator drop down: You can select the operator for the expression here. For example, "="
  - The Value type drop down: You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.

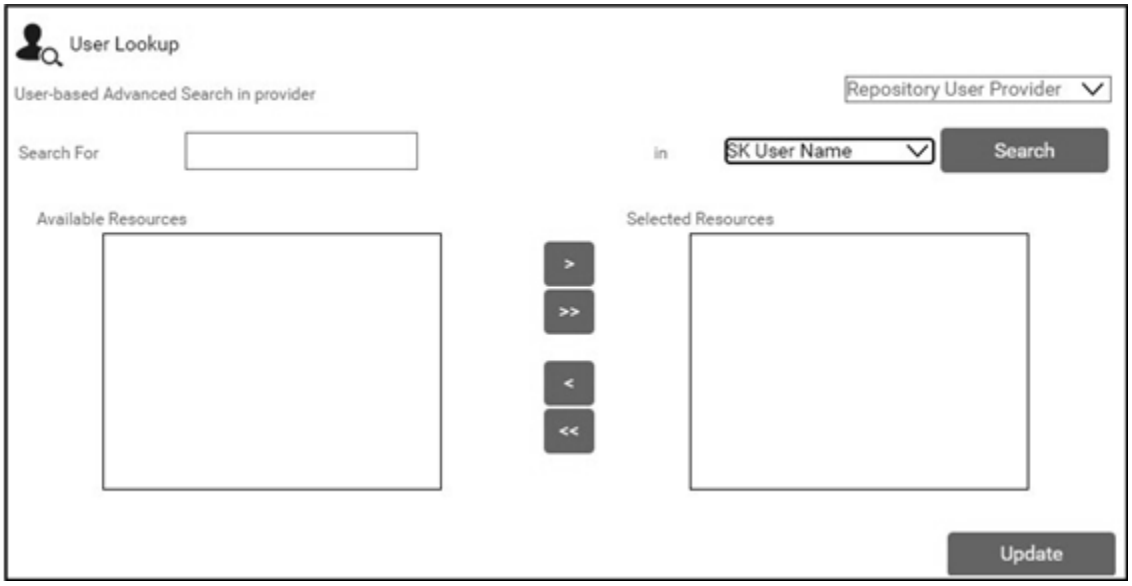
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The Value text box or drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression

At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

- **Assigning an Actor through User Lookup**
  Through the **User Lookup** you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup
  1. Select the **User Lookup** option.
     The **User Lookup** dialog is displayed, search and select the required actor.



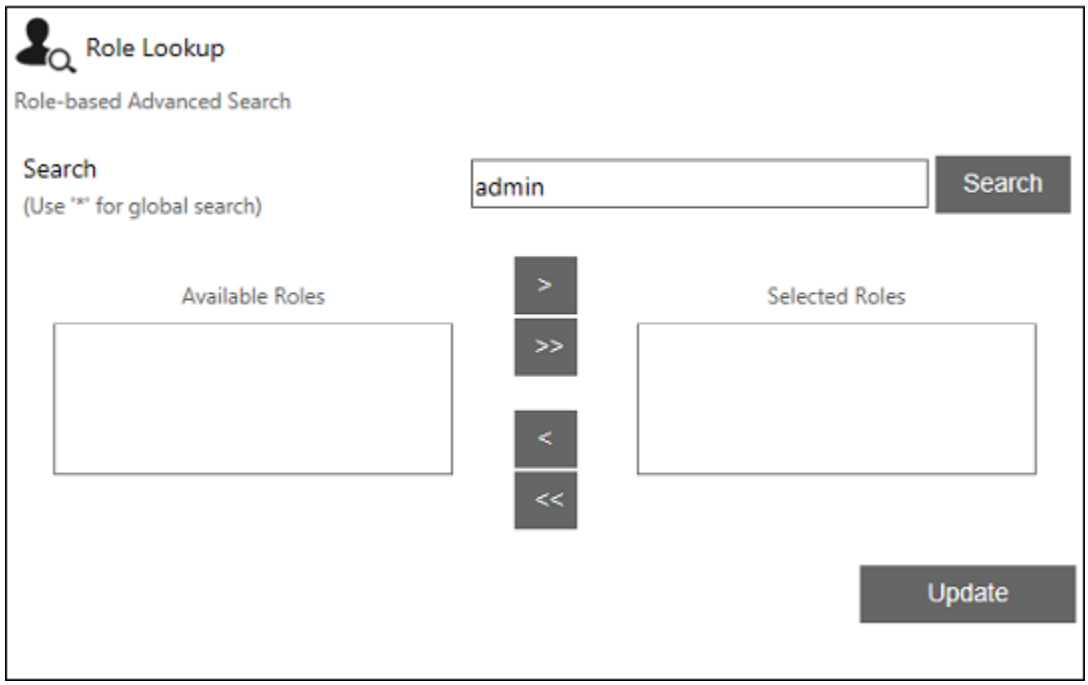
  2. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
     In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Click **Update** in the Assign Actor(s) window to update the selected actor(s) in the **Assign Actor** property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

   Assigning an Actor through Role Lookup
   Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

4. Select the **Role Lookup** option.
5. The **Role Lookup** dialog is displayed, search and select for the required role.

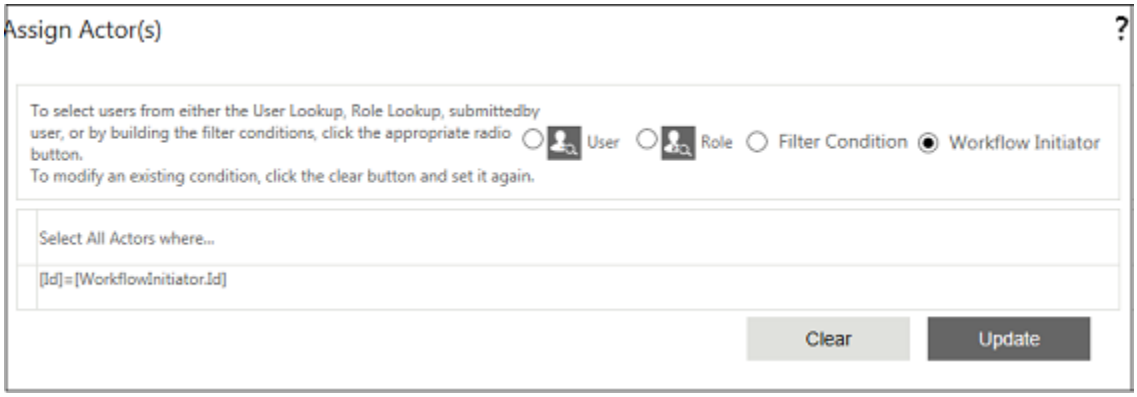


6. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. Click **Update** in the Assign Actor(s) window to update the **Assign Actor** property.

- **Assigning an Actor through Workflow Initiator**
  Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

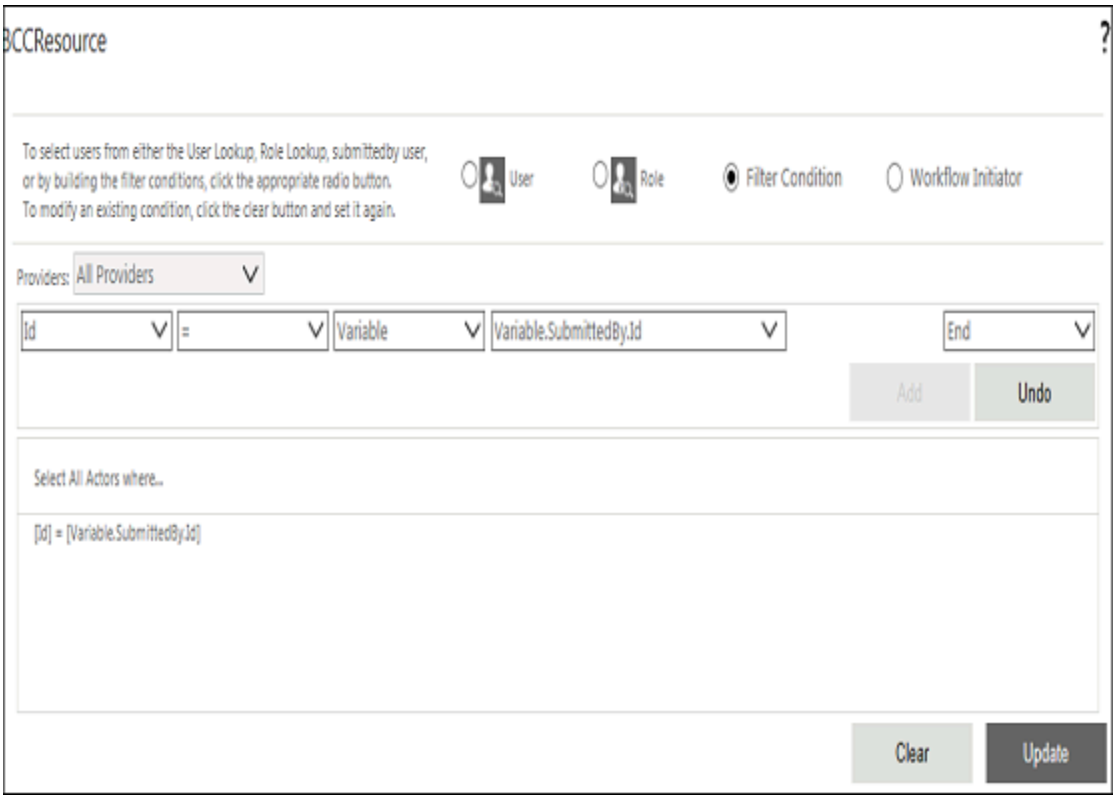


Email: BCC Resource

You can set the actors to whom the Email is to be sent as BCC in this property window. To open this window, click the button for the 'BCCResource' property.

The following image shows the BCCResource property dialog box:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

Different Methods of Assigning an Actor

- **Assigning an Actor by building an expression:**

  Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

  - The Providers drop-down- You can select the required provider from the Providers drop-down.
  - The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
  - The Operator drop down - You can select the operator for the expression here. E.g., "="
  - The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
  - The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
    - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
    - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

collection from which you can select an appropriate option.

- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

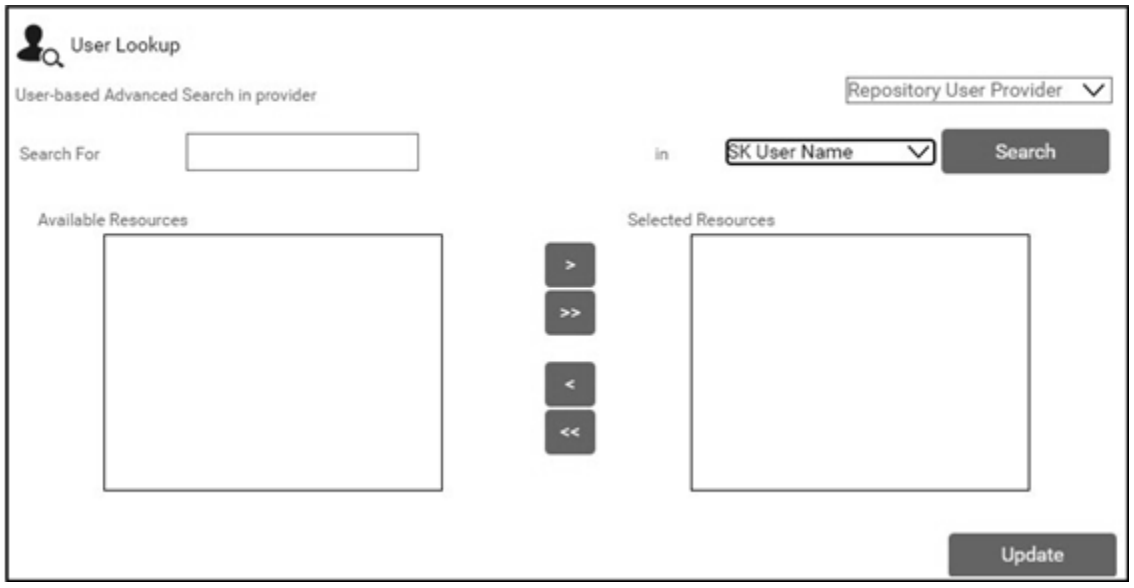
At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

Assigning an Actor through User Lookup

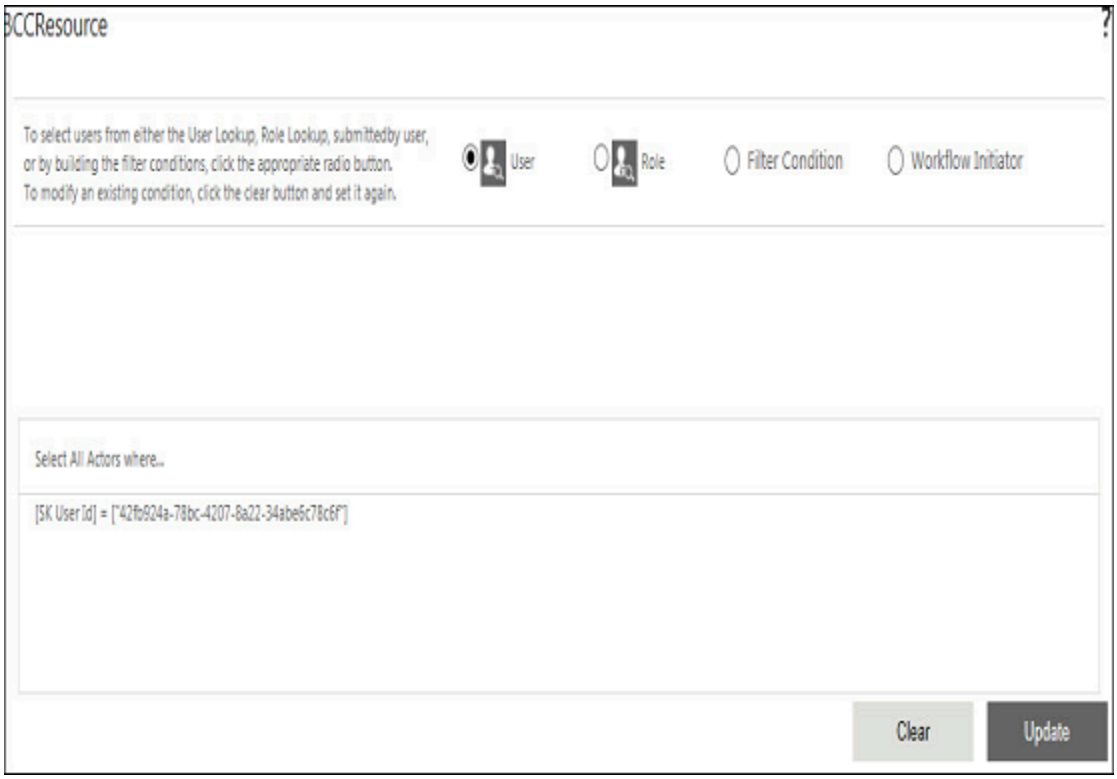
Through the **User Lookup** you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.
2. The **User Lookup** dialog is displayed, search and select the required actor.



3. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



4. Click **Update** in the Assign Actor(s) window to update the selected actor(s) in the **Assign Actor** property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

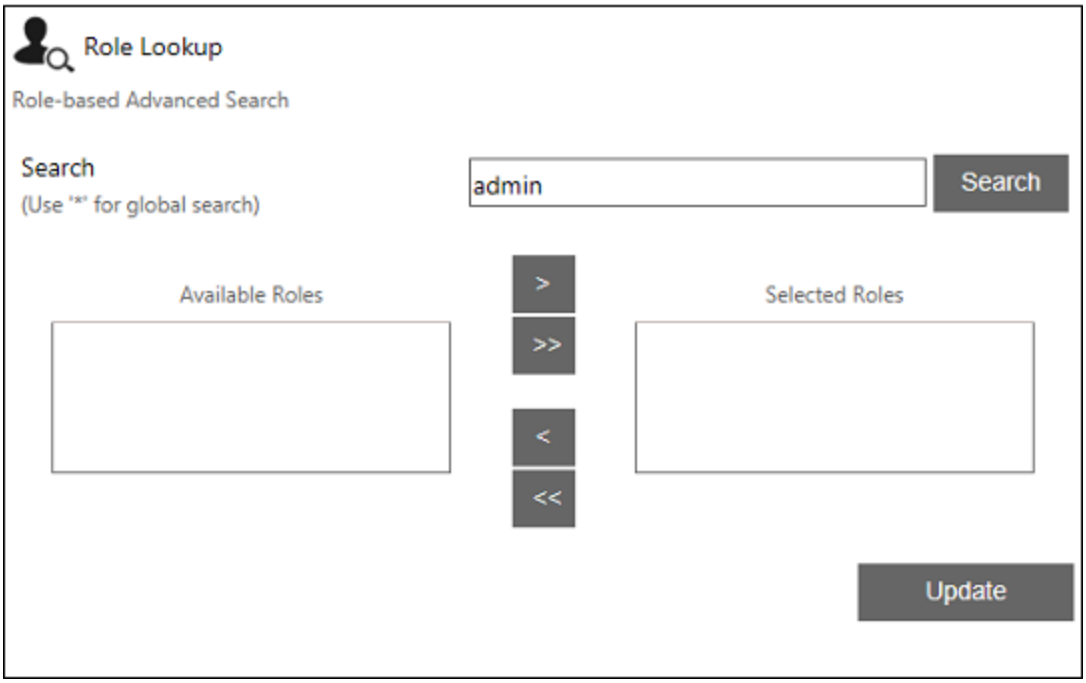
- **Assigning an Actor through Role Lookup**

  Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

  1. Select the **Role Lookup** option.

     The **Role Lookup** dialog is displayed, search and select for the required role.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



2. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

3. Click **Update** in the Assign Actor(s) window to update the **Assign Actor** property.

- Assigning an Actor through Workflow Initiator

  Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor

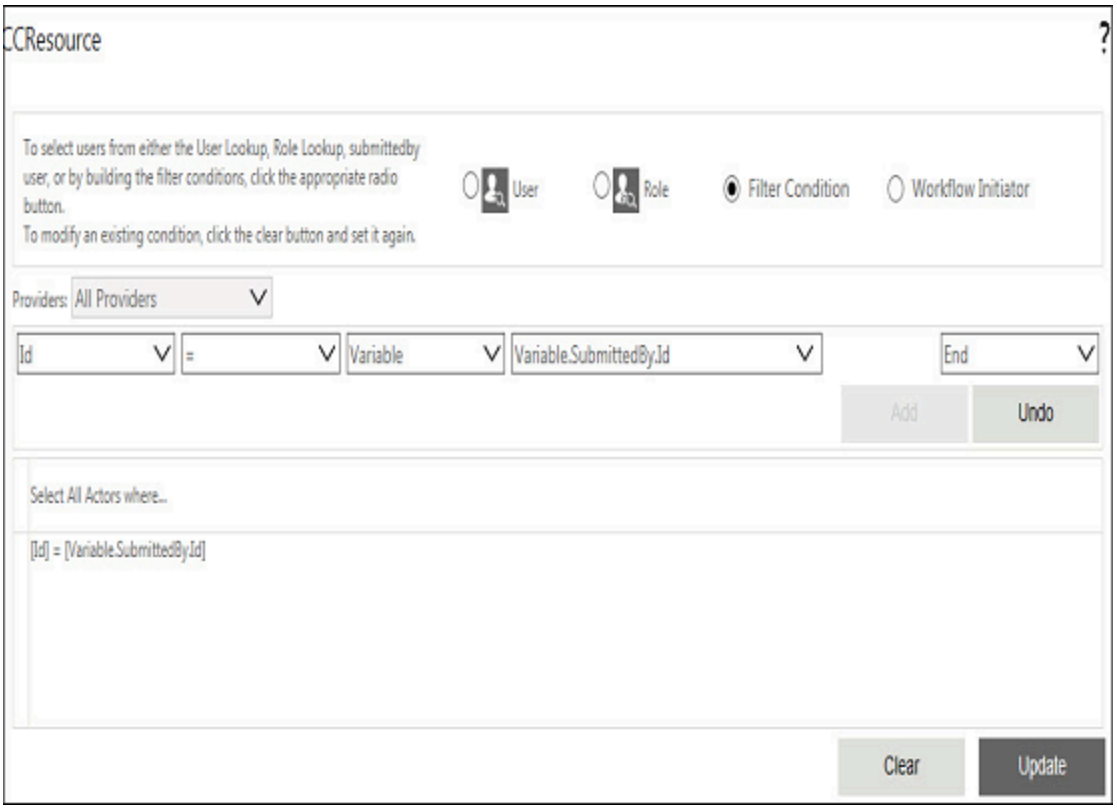
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

window, refer the image below:

Email: CCResource

You can set the actors to whom the Email is to be sent as CC in this property window. To open this window, click the button for the 'CCResource' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

Different Methods of Assigning an Actor

- **Assigning an Actor by building an expression:**

Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.

- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

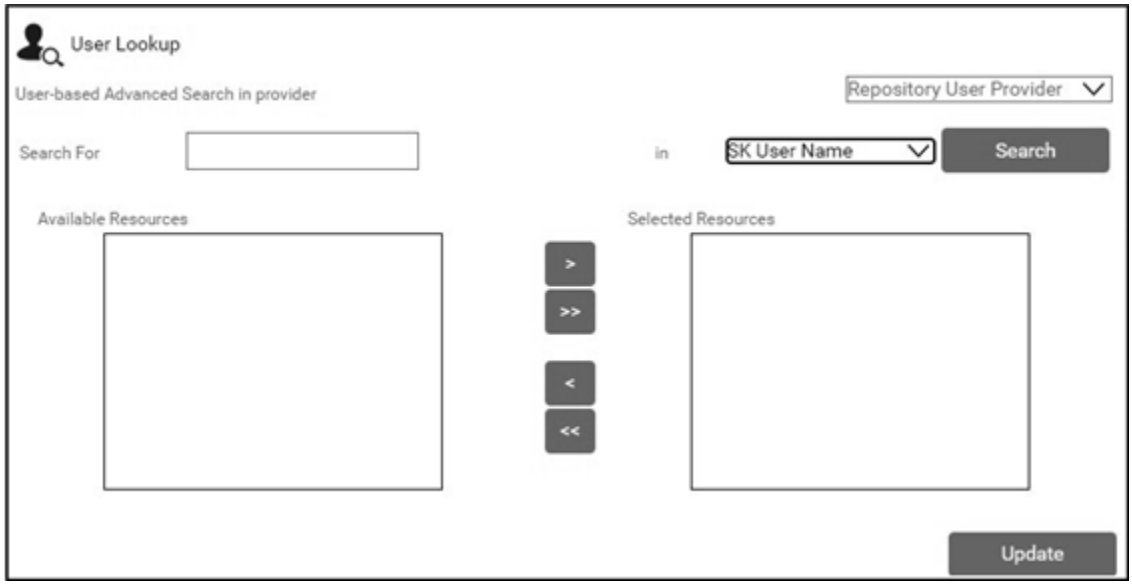
At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

- **Assigning an Actor through User Lookup**

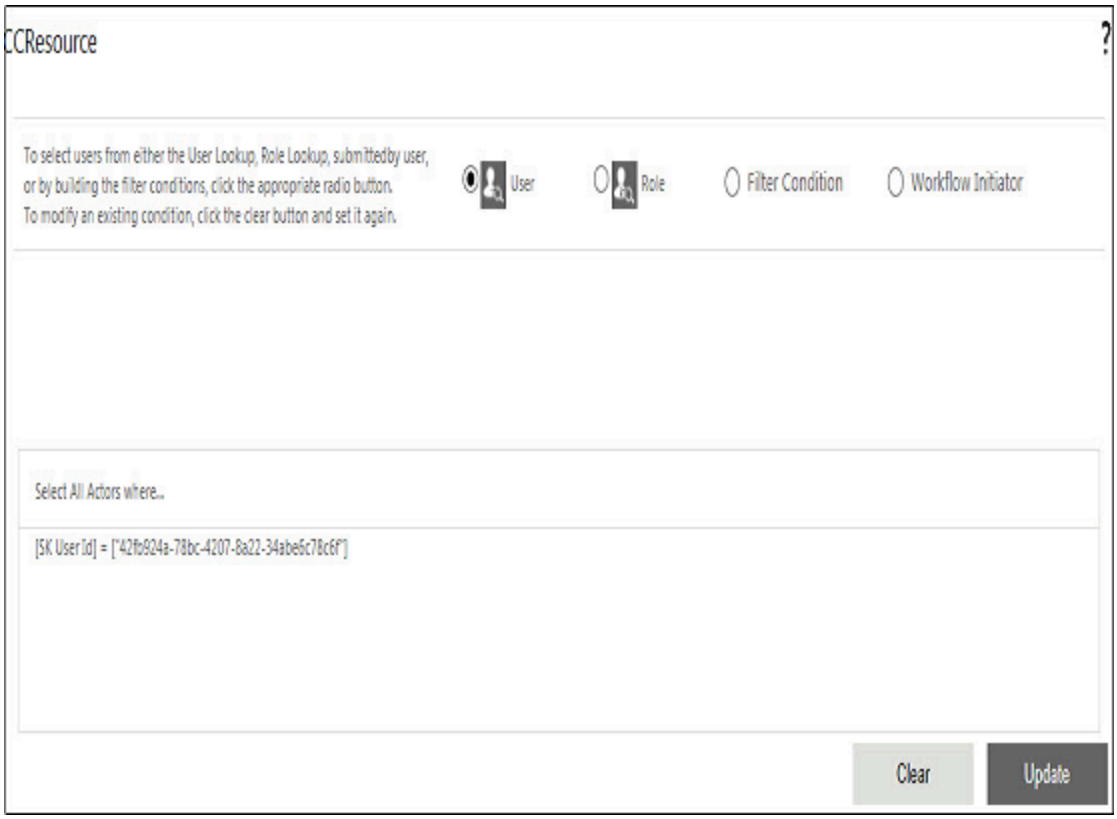
Through the **User Lookup** you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.
   The **User Lookup** dialog is displayed, search and select the required actor.



2. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click **Update** in the Assign Actor(s) window to update the selected actor(s) in the **Assign Actor** property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

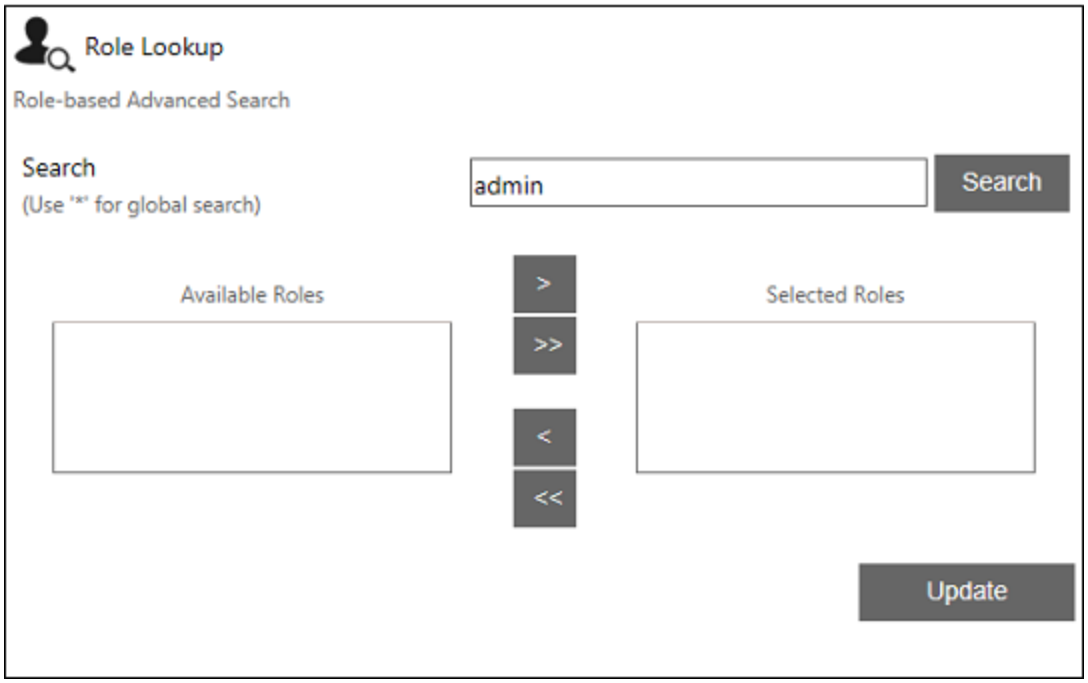
- **Assigning an Actor through Role Lookup**

Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

1. Select the **Role Lookup** option.

   The **Role Lookup** dialog is displayed, search and select for the required role.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



2. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

3. Click **Update** in the Assign Actor(s) window to update the **Assign Actor** property.

- **Assigning an Actor through Workflow Initiator**

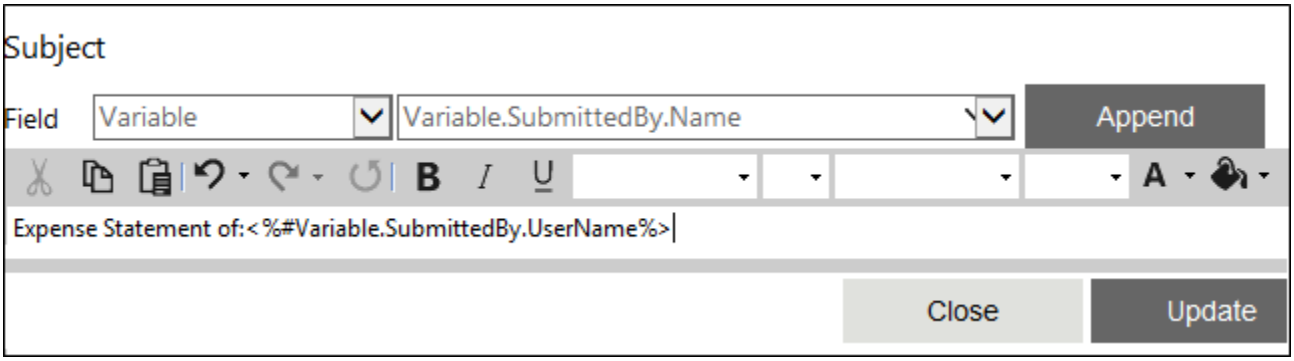
Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Email: Subject

Subject Property window

You can specify a subject for the Email in this property window. To open this window, click the button for the 'Subject' property.



The window has a text editor to enter the subject. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the subject. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the subject in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the subject.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the subject using a combination of discrete value and variable or content. For example, you can specify the subject as "Expense Statement of <%#Variable.SubmittedBy.Email%>". Here the string "Expense Statement of" should be typed directly in the text editor. The second half of the subject is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the **Field** drop down and clicking **Append**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The text tool bar above the text editor allows you to manage and format the subject. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

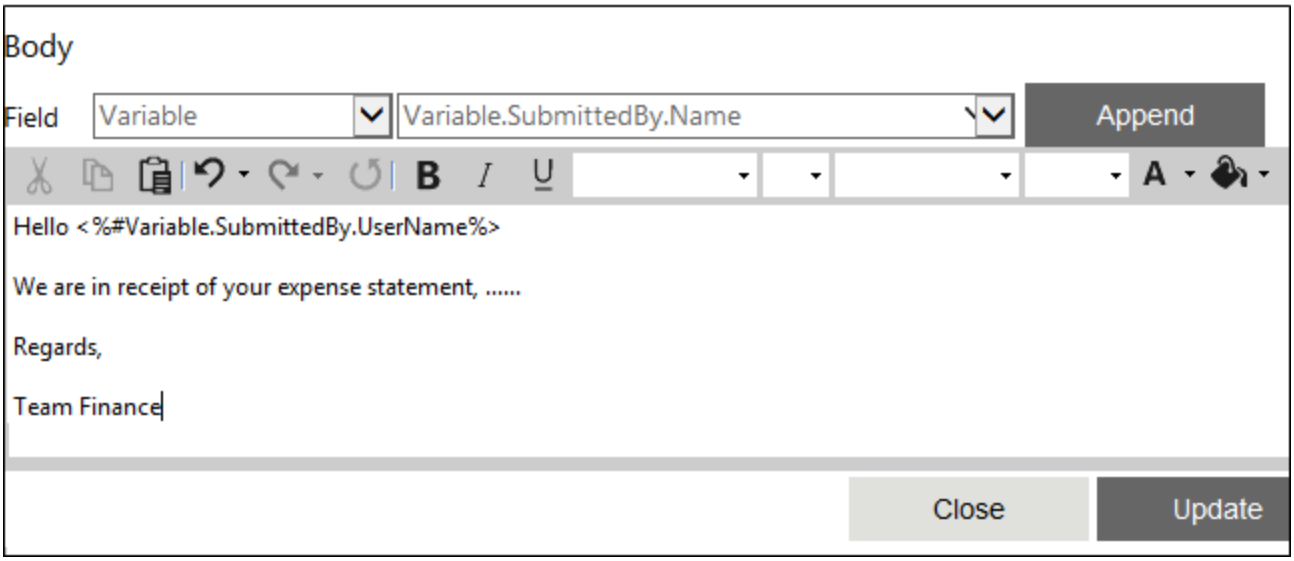
- **Update** - You can click this button to save the subject.
- **Close** - You can click this button to close this window.

Email: Body

You can specify a message Body for the Email in this property window. To open this window, click the button for the 'Body' property.

The following image shows the Email Body property dialog box:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the body text. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/ drop down - You can specify the actual value of the message body. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the body text in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the body.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the body using a combination of discrete value and variable or content. This helps in personalizing the email body for each actor. For example, you can specify the salutation line in the message body as "Hello <%#Variable.SubmittedBy.name%>,". Here, the string "Hello" must be typed directly in the text editor. The second half of the salutation is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the Field drop down and clicking Append. The rest of the message body can be typed in the editor. This will result in the same email message going to different actors with their respective names in the salutation line.

The text tool bar above the text editor allows you to manage and format the body text. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

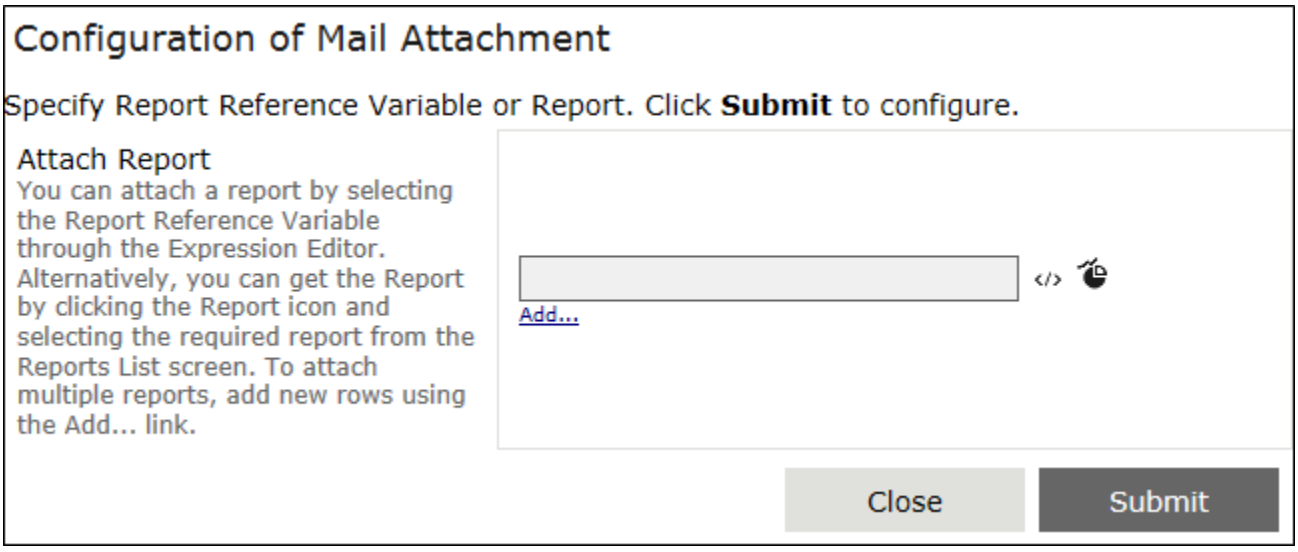
At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the message body.
- **Close** - You can click this button to close this window.

Email: Attach Reports

You can configure the **Attach Reports** property for the email activity in this property window.

In the following **Attachment Configuration** property window, you can configure the Report attachments that you want to mail:



Attach Report

You can use the **Attach Report** property to attach a report generated by the **Report Generator** activity within the workflow or to select a report from the list of generated Reports. You can attach a Report by selecting the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

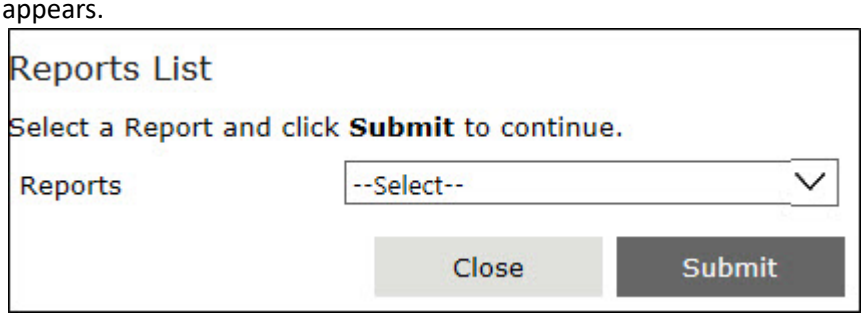
required Report Reference Variable through the Expression Editor. Alternatively, you can select the report from the Reports List screen which contains already generated reports.

Note: There must be at least one **Report Generator** activity before an Email activity in the workflow to select the Report Reference Variable (using Expression Editor) as Attachment.

The Report Reference Variable is generated while configuring the Report Generator activity. By default, the report reference variable will be the **[Template Name]_[activity Name].** For more information on Report reference Variable, see Configure Report Template.

Complete the following tasks to configure an email attachment:

1. Click the edit button for the **Attachment** property.
2. In the **Attach Report** text box, you can either use the expression editor to get the variable of the report or use the Reports List to select the report.
3. To get the variable, click the **Expression Editor** icon to select the required Report Reference Variable.
4. In the **Expression Editor** window, select the required variable.
5. Click **OK** to save variable.
6. Click **Close**. The variable appears in the Attach Report text box.
7. To select the Report from the Reports List window, click the Reports List icon. The Reports List dialog box appears.



8. In the **Reports List** dialog box, select the required Report, and then click **Submit**. This window will close automatically.
9. Click **Close** to close the configuration window.

Configure Communication Activities

This section provides information to configure communication activities. A standard GSM Modem device, such as GSM Commercial Modem 81GC connected to the server is a prerequisite for sending and receiving SMS from AVEVA Work Tasks.

Prerequisites

A standard GSM Modem device, such as GSM Commercial Modem 81GC connected to the server is a prerequisite for sending and receiving SMS from AVEVA Work Tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure Farm Configuration Wizard for Receiving Emails

To configure the AVEVA Work Tasks Communication Service for processing emails, you must set attributes. The properties can be set from the Farm Configuration wizard by clicking the Settings image next to **Communication Service**.

On clicking icon, the **Communication Service Settings** window is displayed.

Communication Server

Under the **Communication Server** tab, you need to configure the following fields:

1. **Listener IP Address** - The IP Address of the Service.
2. **Port** - The Port assigned to the Service.
3. **Wait Task Handler Path** - The Assembly path for handling timeout task & handler for received email (this is used for customization).
4. **Wait Task Handler Class** - The Class name for handling timeout task & handler for received email (this is used for customization).
5. **Block mail Addresses** - The Communication Service will not process mails coming from these addresses like abc@dom1.com.

Enterprise Server

Under the **Enterprise Server settings** section of the **Communication Server** tab, you need to configure the following fields:

1. **Load Balanced** - This field is used to set load balancing for AVEVA Work Tasks. This field is enabled only for the Load Balanced Server installation. To enable load balancing set the Load Balanced value to true.
2. **Load Balanced Server Port** - This field denotes the port number used by Enterprise Server. This value must be modified if the default port value is in use.
3. **Load Balanced Client Port** - This field is used to enter the port number for the Client.

Mail Server

Mail Server configuration is moved to Central Configuration. To configure mail server, see Mail Server Setting.

Supporting Matrix POP3 Service

Assume that, there are two POP3 service providers are running. By default, the first POP3 service is being used. If this POP3 service is stopped and can not be restarted then use the alternate Matrix POP3 service which is running.

To do so, the machine name/IP address of the machine where the Matrix POP3 service is running needs to be provided in the **POP Mail Host** field of the Mail Server tab of the **Farm Configuration wizard** window.

Configure Farm Configuration Wizard for Receiving SMS

The AVEVA Work Tasks Communication Service Settings will check the Phone device periodically for SMSs.

To configure the Farm Configuration Wizard for receiving SMS, you must set SMS Receiver attributes for the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

following fields:

1. **Enable SMS Receiver**
   This is for enabling & disabling periodic checking of the device for SMS. If this is set to "True", the Notification Engine will check for SMS and if "False", the check is disabled.
2. **SMS Time Interval**
   The periodic Time interval for checking SMSs in the Device. Specify it in a time span (e.g. 0.00:01:30 = 1 minute 30 secs). It will delete the SMS from device based on the time interval.
3. **Wait Task Handler Path**
   The Assembly path for handling timeout action & warning.
4. **Wait Task Handler Class**
   The Class name for handling timeout action & warning.
5. **Provider Configuration**
   This is used to enter service-provider-specific details for receiving SMS. Currently details for one provider – GSM – have been given. If you are using any other service, you can add your provider details here.
   Name - Name of the provider. The supporting providers are GSM, Email Provider, and SMS pattern match.
   Assembly - Enter the required dll file path.
   Class - Class Name.
   Device - Name of the device that is connected to the COM port.

   Note: Under each Connection Type, apart from their configuration details, two compulsory nodes – SMSMessengerAssembly & SMSMessengerClass – have to be specified. This is the class which sends SMS for each provider. This class should implement ISMSSender interface.

Configure Farm Configuration Wizard for Sending SMS

The SendSMS Activity can be configured to send SMSs by making the following **SMS Config** settings in the Configuration Wizard:

1. **Resource Fields**

   This is for mapping resource fields(Users Properties) for sending SMS. There are three fields. They are Email, MobilePhoneProvider, and SMSNumber are mapped to the table.

   Email - Mail id of the user. This field will take input from Users Properties of specified field in this field.

   MobilePhoneProvider - Service Provider. This field will take input from Users Properties of specified field in this field.

   SMS Number - Number of cell in which SMS to be sent. This field will take input from Users Properties of specified field in this field.

   Example: For the List, the Email, MobilePhoneProvider, and SMSNumber will take input from Email, Mobile Phone Provider and SMS Number respectively from the User properties.
2. **SMS Sending Connection Types**

   Currently three Connection Types are supported:

   GSM - You can send a SMS through a direct GSM modem or GSM phone.

   EmailProvider - You can send a SMS by specifying connection types.

   SMSPatternMatch - Each service provider has their SMS pattern to identify easily.

   Example: Hutch - Hutch number starts from 9886.*

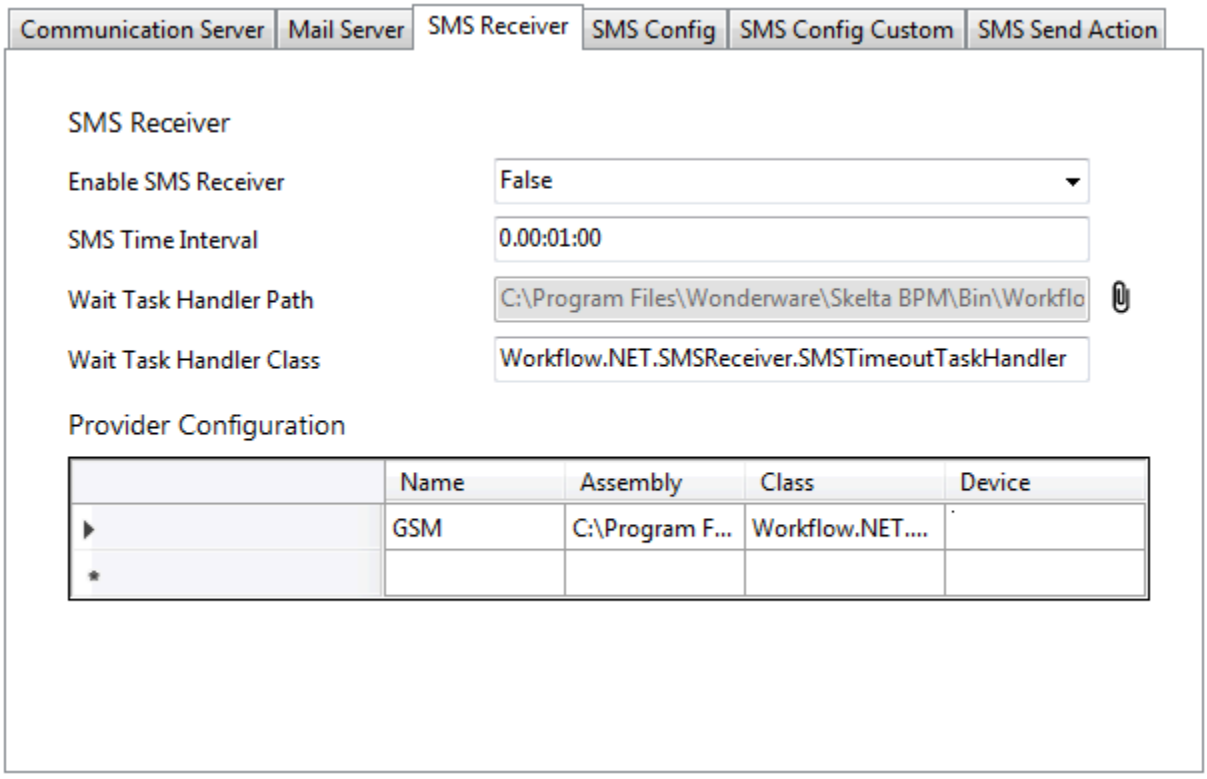
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Airtel - Airtel number starts from 9845.*, 9980.* etc.

Provider Configuration

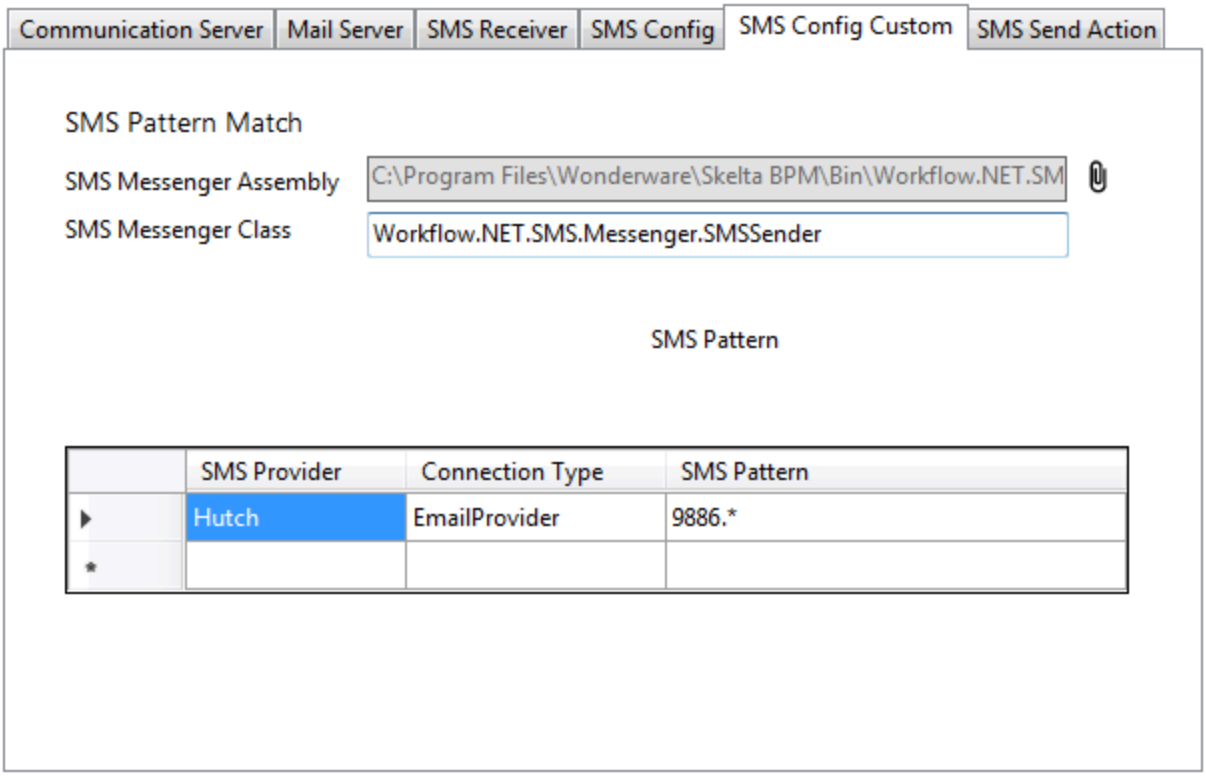
This is for Connection Type specific configuration settings. If you are using a provider other than the default provider specified here, you will have to add the details here. Currently three Connection Types are supported – SMSPatternMatch, GSM, and EmailProvider. Under each Connection Type, apart from their configuration details, two compulsory nodes – SMSMessengerAssembly & SMSMessengerClass – have to be specified. This is the class which sends SMS for each provider. This class should implement ISMSSender interface.

The following are some examples for setting ProviderConfiguration:



Note: As an alternative, "Nokia 6310i Modem" can be replaced by "COM 1", provided the Nokia phone is connected on COM1.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



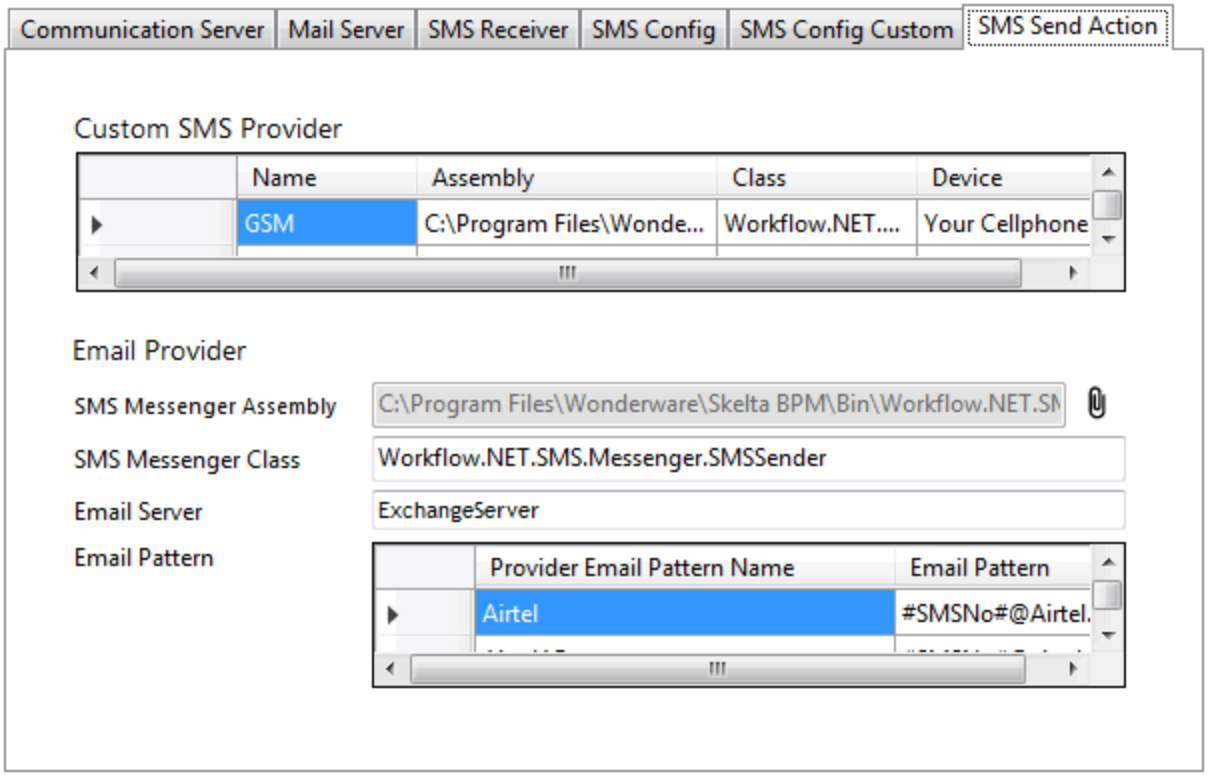
Here are some details about values in the above example:

DeviceSettings - 8 databits, no parity, 1 stopbit

Provider Type - Standard TAP provider

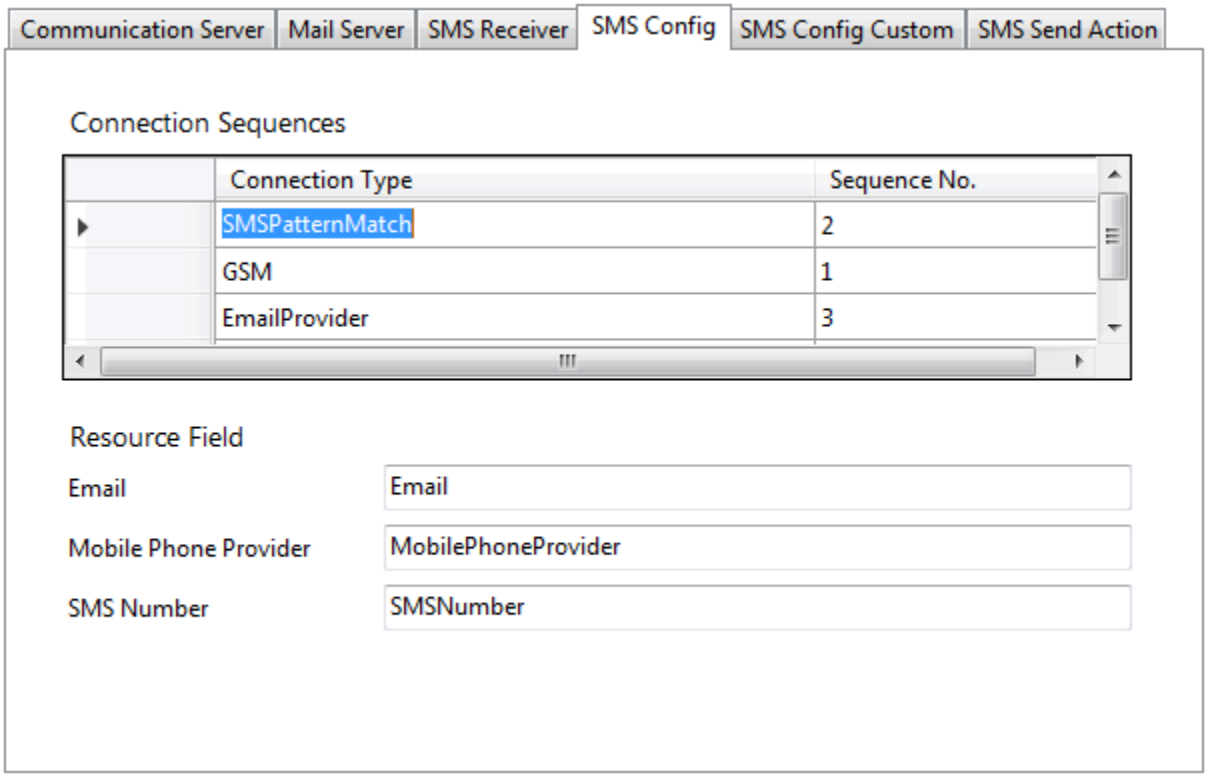
You can specify '0' for UCP Provider, '2' for TAP provider, 'no request' for LineFeed and '3' for TAP provider, 'no request' for EOT.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. **Connection Sequences**

   This is the sequence in which the AVEVA Work Tasks Engine will choose the provider for sending SMS. For Example:



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Engine will try to send SMS using GSM Provider. If this fails, then it will try sending the SMS using other providers in this order: SMSPatternMatch and EmailProvider.

Provider	Priority
GSM	1
SMSPatternMatch	2
EmailProvider	3

SendSMS and WaitForSMS

The SMS Send and Receive actions communicate either through a serial port, Bluetooth connection, infrared port or any TAPI compliant device. When using the COM port directly, device properties must be setup properly before you can start to send or receive messages. When using TAPI, these settings are controlled by the TAPI telephone driver, and should be configured through the TAPI settings as described below:

- Open the Windows Control Panel;
- Double-click on 'Phone and Modem Options' and select the 'Modems' tab;
- Select the preferred modem and click on 'Properties'; from here, you can change the settings.

To select a TAPI device in the toolkit's object, you must assign the Device member:

- Device - the literal TAPI device string, for instance: 'Standard 9600 Modem'.

If you prefer not to use TAPI, you must assign COM port properties through the following member variables:

- Device - the actual COM port device, formatted as COM x, where x represents the COM port number;
- DeviceHwFlowControl - hardware flow control is used to regulate commands from PC to device and visa versa; default: On;
- DeviceSwFlowControl - software flow control is used to regulate data from device to device; default: Off;
- DeviceSpeed - the speed to be used to communicate with modem or GSM phone; default: 0 (which means that factory defaults are used for modem communications);
- DeviceSettings - the number of databits, paritybits and stopbits; default: 8 databits, no parity, 1 stopbit;
- DeviceInitString - the modem initialization string is used to initialize the modem; default value: "" (which means a generic initialization string will be used).

Before you can actually send an SMS or Pager message, the Toolkit needs to know a bit more about the sender, recipient and the actual message. Therefore, you must assign the following properties:

- Sender - the phone number of the sender of the message; only digits are allowed. The sender address will be displayed on the recipient's mobile phone. If not assigned, 00000000000 will be used as sender address;
- Recipient - the phone number of the recipient of the message; if not specified, the message cannot be sent;
- MessageText - the actual message body, containing any character. Maximum message length is 160 characters when using normal (non-Unicode) text. When Unicode is used, you can only send 70 characters;

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Dial-in number of the SMS/Paging service provider (not required if you send a message through a directly connected GSM phone).

If you send through an SMSC provider, you must tell the component which provider protocol you are going to use:

- ProviderType - the type of protocol requested by the provider. You can choose: UCP or TAP.

Important: Sending an SMS through a service provider requires special knowledge of the protocol and provider requirements of your equipment. A provider requires its own speed (1200 bps is most common, but also 300 and 2400 are also often required), data format (7 databits, no parity, 1 stopbit or 8 databits, even parity and 1 stopbit) and provider type (UCP or TAP).

Supported Phone Models

The following table depicts the supported phones by AVEVA Work Tasks:

Brand	Type	Send	Receive	Comments
Ericsson	Z520	Yes	Yes	
	Z525	Yes	Yes	
	Z530	Yes	Yes	
	Z550	Yes	Yes	
	Z800	Yes	Yes	
Motorola	V188	Yes	Yes	
Nokia	6100	Yes	Yes	CA42 Datacable tested
	6210	Yes	Yes	
	6220	Yes	Yes	
	6310	Yes	Yes	
	6310i	Yes	Yes	
	6610	Yes	Yes	
	8250	Yes	Yes	
	8910	Yes	Yes	
Siemens	C35	Yes	Yes	
	C45	Yes	Yes	
	M35	Yes	Yes	

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Brand	Type	Send	Receive	Comments
	M45	Yes	Yes	
	S35	Yes	Yes	
Sony-Ericsson	F500	Yes	Yes	
	K310	Yes	Yes	
	K320	Yes	Yes	
	K500	Yes	Yes	
	K510	Yes	Yes	
	K600	Yes	Yes	
	K700	Yes	Yes	
	K750i	Yes	Yes	
	K800i	Yes	Yes	
	S700	Yes	Yes	
	S710	Yes	Yes	
	T226	Yes	Yes	
	T230	Yes	Yes	
	T238	Yes	Yes	
	T290	Yes	Yes	
	T310	Yes	Yes	
	T610	Yes	Yes	
	T630	Yes	Yes	
	T637	Yes	Yes	
	T68	Yes	Yes	
	T68i	Yes	Yes	
	V800	Yes	Yes	
	W300	Yes	Yes	
	W550	Yes	Yes	

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Brand	Type	Send	Receive	Comments
	W600	Yes	Yes	
	W700	Yes	Yes	
	W800i	Yes	Yes	
	W810	Yes	Yes	
	W900	Yes	Yes	
	Z1010	Yes	Yes	
	Z500	Yes	Yes	
	Z600	Yes	Yes	

Disable Reply Email for Invalid Emails

Reply emails for invalid emails can be disabled. If any invalid email comes to the notification engine, an automated reply – **"System could not process your mail. Some of the keywords not found in your mail to process it. Please reply to the original message again."** will be send to the From address with the same subject. This action has been made configurable now. User can disable this reply if required.

This can be achieved by adding a registry string value called **SendReplyForInvalidMail** under **HKEY_LOCAL_MACHINE\SOFTWARE\AVEVA\WorkTasks** and set its value data as **NO**.

Notification Activity

Activity description:

The Notification Activity is incorporated in the workflow process design to send notifications to actors, in the form of email and SMS. The messages can be customized for each actor by appending the resource properties to the message.

Activity Properties:

The Notification activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Notification activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Notification Type - The Notification type can be selected from the drop down list provided for this property. Notifications can be sent in the form of email or SMS to cell phones. The 'All' option is provided to send messages through all media.

Property Type: Mandatory (This property must be set if the action is to be assigned to any resource(s).)

Actors

The property in the Actors group has been described below. You can use this property to specify the content of the notification.

ToResource- This property specifies the resource(s) to which the notification is to be sent.

Property Type: Mandatory (This property must be set if the action is to be assigned to any resource(s).)

Steps to set the To property

1. To specify the actors to receive the Notification message, click the button for the 'To' property to open the property window.
2. In this window, build an expression to select the actors using the actors properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop- down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Notification - ToResource for a detailed description of the To property window.

See the Using XML Variables section in the About XML Variables topic for details on how to use the XMLVariables to build the expression.

Notification Contents

The properties in the Notification Contents group have been described below. You can use these properties to specify the content of the notification.

From Email Address - This property specifies the email address from which the email notifications will be sent. If the Notification type is of the form 'All' or 'Email' then this property must be set. If it is not set, then the email notifications will not be sent.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if the notification type selected in the corresponding property is either 'All' or 'Email'.)

Message - This is the actual content of the notification sent to the resource(s). The notification can be customized for each resource.

Property Type: Optional (This property needs to be set only if required.)

Steps to set the Message property

1. To specify the Notification message, click the button for the 'Message' property to open the property window.
2. In this window, enter the message text directly in the text editor.
3. If you want to personalize the message you can do so using the **Field** controls to add resource property values. For example, to add a message that includes the name of the actor submitting the approval request, you can do as follows: First type in the text "Hello". Next select "Variable" and "Variable.SubmittedBy.name" from the two drop downs for **Field** and click **Append**. This will add the placeholder for the variable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the name, resulting in a personalized salutation. Type in a message in the next line to complete the message body.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the message, click **Update** to save it.
7. Click the **Close** button to close the window without saving the message.

See Notification: ToResource for a detailed description of the Message property window.

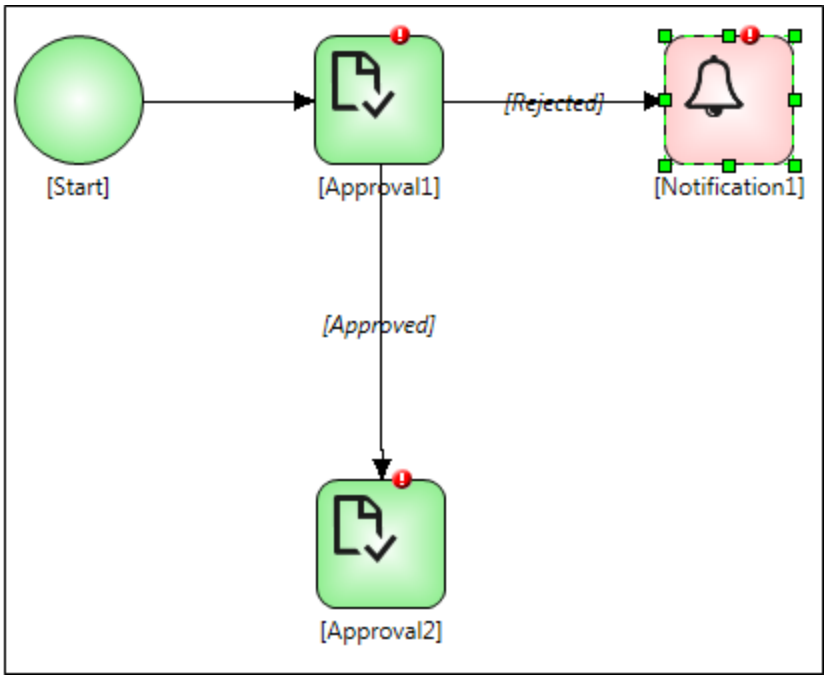
Activity outputs:

The Notification Activity does not return any value.

Example for Notification Activity

In an organization, whenever Approver1 rejects a Notification send to Manager, if manager approves it will go for next Approver, as shown in below workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



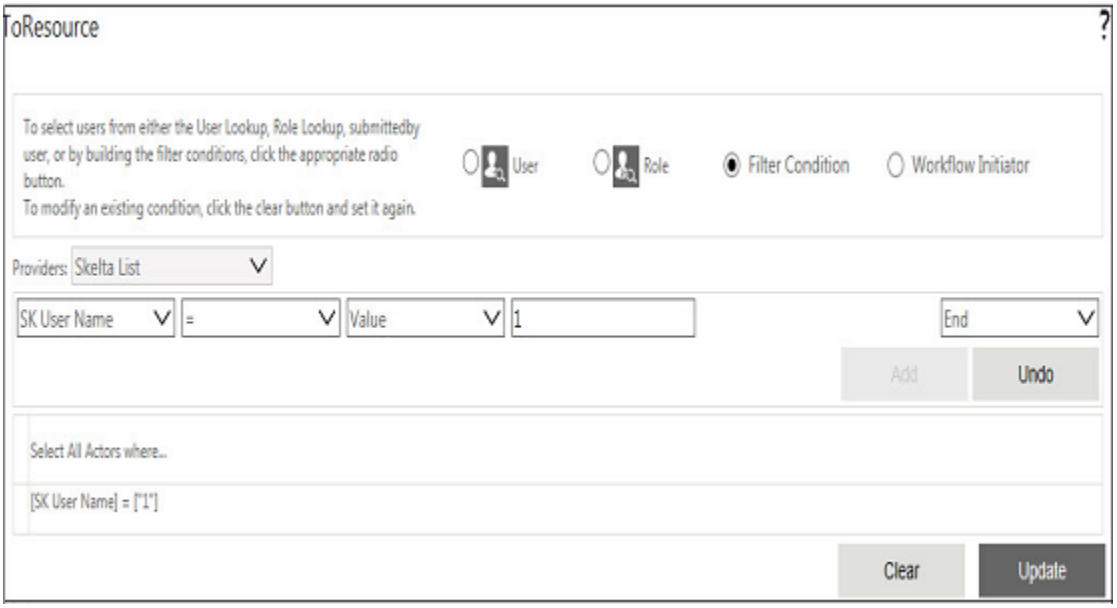
Whenever a workflow triggers a Request, it will send to Approval1. If Approval1 rejects, a Notification will send to Assign actor.

Settings For Notification Activity:

Notifications can be sent in the form of **Email** or **SMS** to cell phones.

Step 1: Select **Notification Type** as **Email** type from the drop down to send notification.

Step 2: Click **ToResource** property to specify the resource(s) to which the notification has to be sent.



Step 3: Select specified Actor and click **Update** to save the changes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Notification: ToResource

You can set the actors to whom the Notification is to be sent in this property window. To open this window, click the button for the 'To' property.

The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

Different Methods of Assigning an Actor

- **Assigning an Actor by building an expression**:

  Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.
- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

displayed depends on the option selected in the preceding drop down.

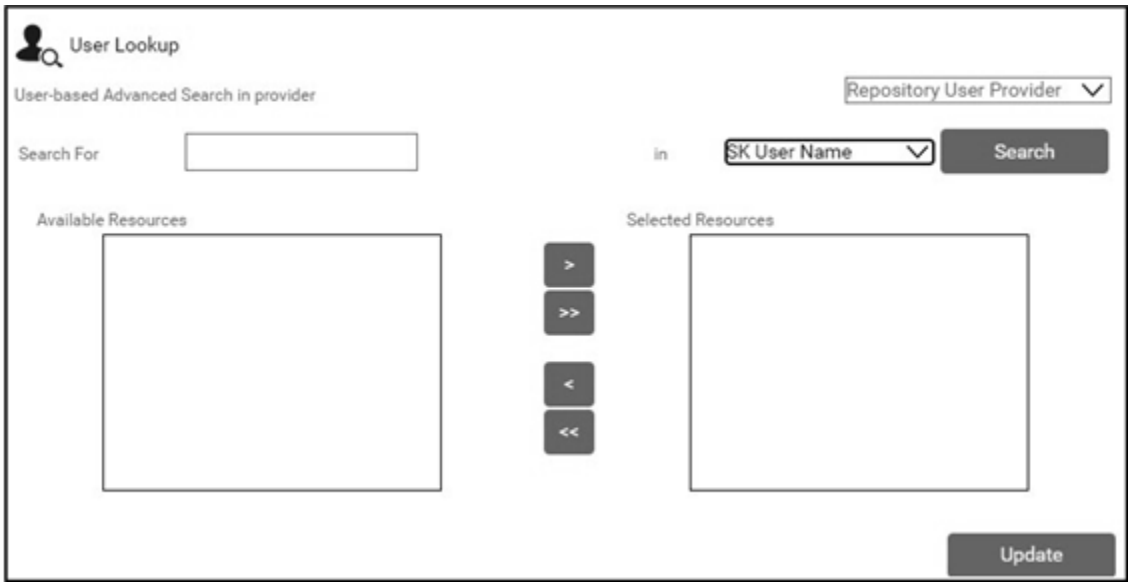
- If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
- If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.

- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

- **Assigning an Actor through User Lookup**
  Through the **User Lookup** you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup
  - Select the **User Lookup** option.
  - The **User Lookup** dialog is displayed, search and select the required actor.

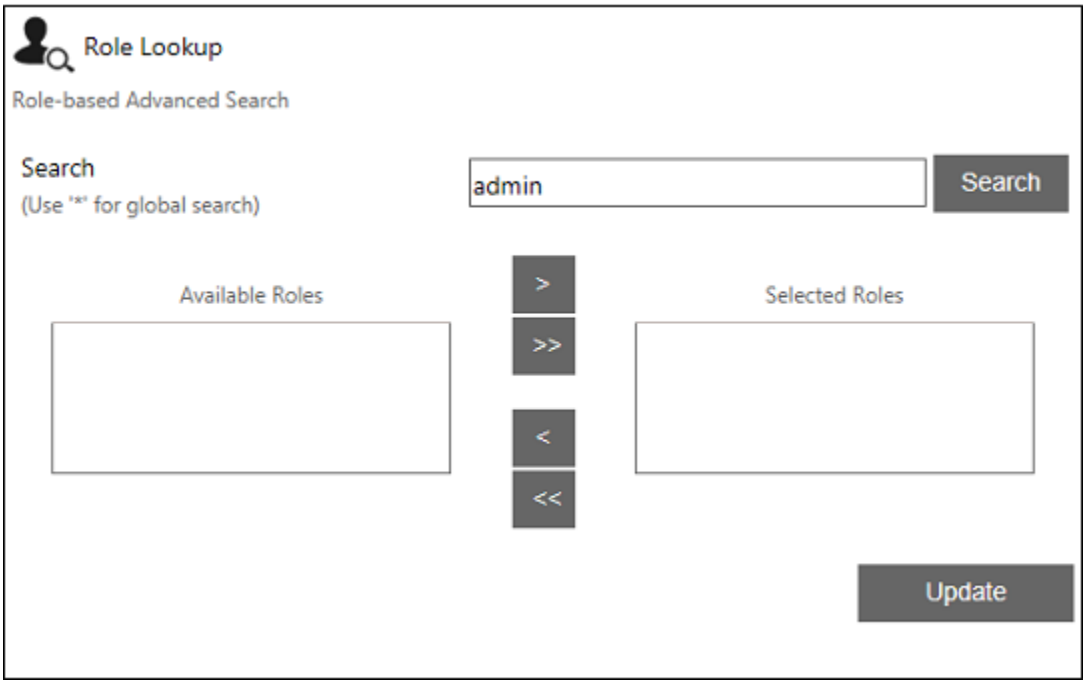


  - Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
    In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Click **Update** in the Assign Actor(s) window to update the selected actor(s) in **Assign Actor** property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

- **Assigning an Actor through Role Lookup**
  Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup
  - Select the **Role Lookup** option.
  - The **Role Lookup** dialog is displayed, search and select for the required role.

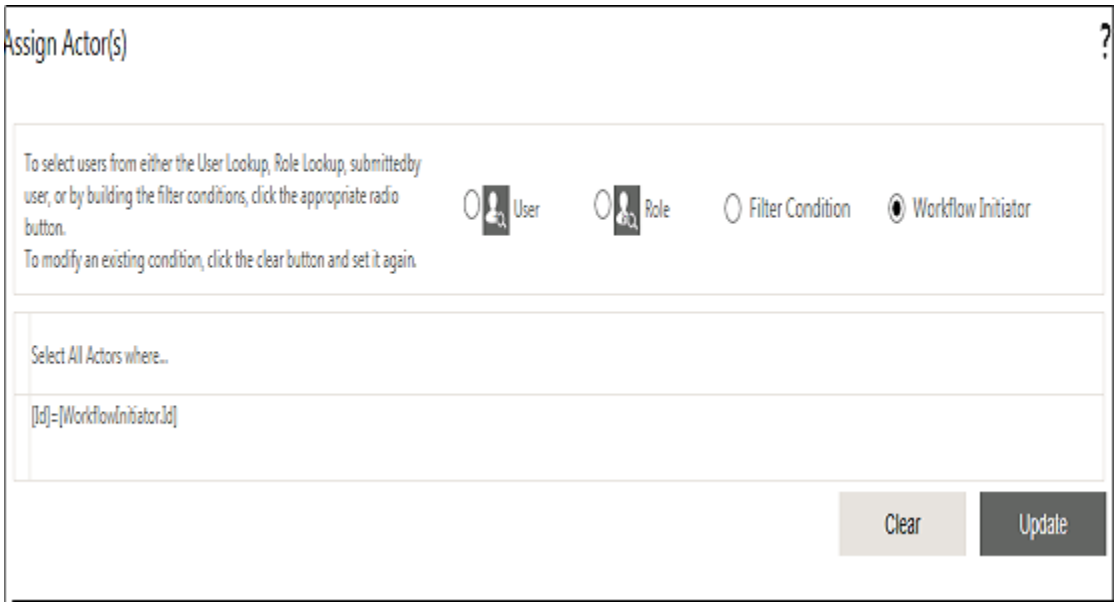


  - Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
    In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Click **Update** in the Assign Actor(s) window to update the Assign Actor property.

- **Assigning an Actor through Workflow Initiator**
  Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

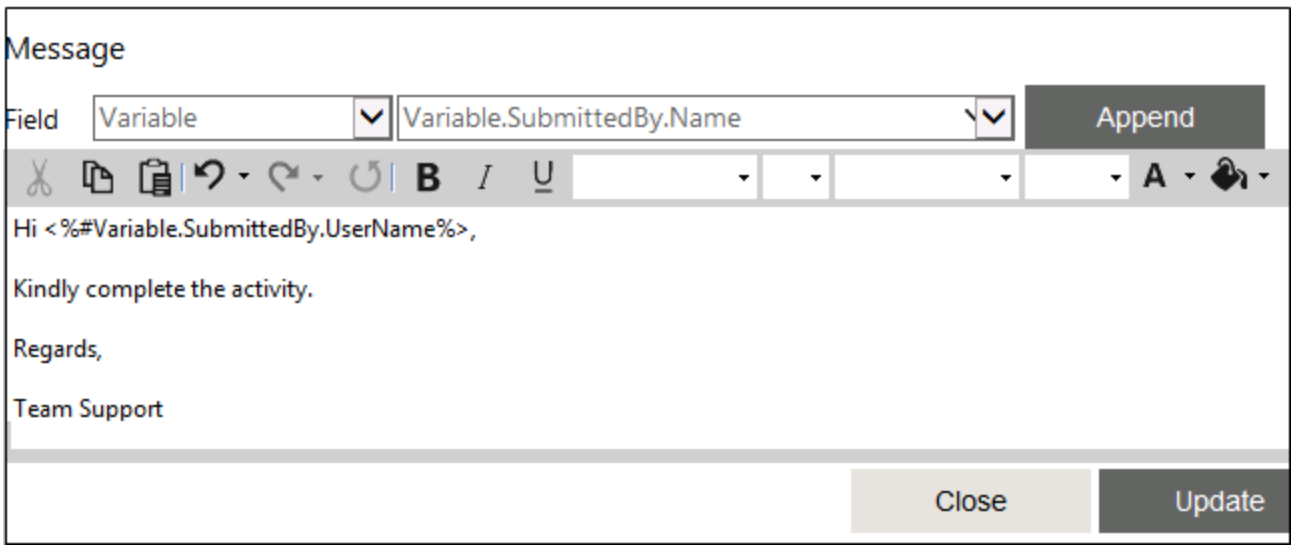


Notification: ToResource

Message Property window

You can specify a message for the Notification in this property window. To open this window, click the button for the 'Message' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the Notification message text. You can do this either by entering a discrete value directly in the text editor or by using the **Fi**eld controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the message. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the message text in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Notification message.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the Notification message using a combination of discrete value and variable or content. This helps in personalizing the message for each actor. For example, you can specify the salutation line in the message as "Hello <%#Variable.SubmittedBy.name%>,". Here, the string "Hello" must be typed directly in the text editor. The second half of the salutation is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the **Field** drop down and clicking **Append**. The rest of the message can be typed in the editor. This will result in the same message going to different actors with their respective names in the salutation line.

The text tool bar above the text editor allows you to manage and format the Notification message text. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the message
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window, you will find the following buttons:

- **Update** - You can click this button to save the message.
- **Close** - You can click this button to close this window.

Send SMS Activity

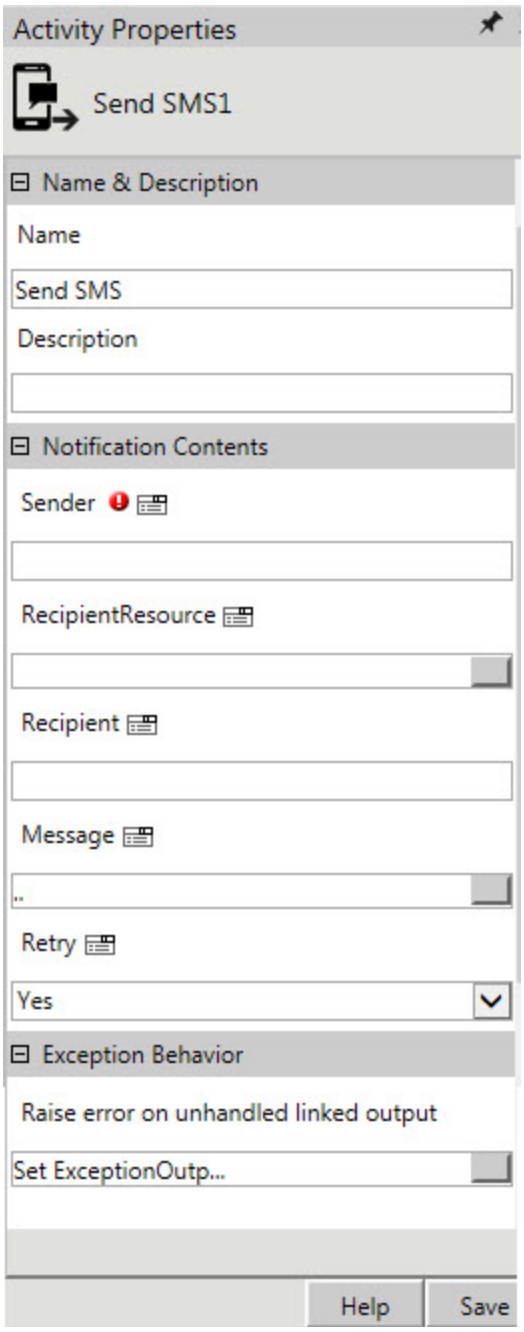
Activity description:

The SendSMS Activity can be used to send SMS to a selected number of actor(s) and/or to a specified phone number. The actor(s) can be selected by building a query. The workflow engine evaluates this query and sends an SMS to all the actors resulting from the query.

Activity Properties:

The SendSMS activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the tool bar or by selecting the appropriate option from the right-click menu for the SendSMS activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Note: In scenarios where different resource providers like Active Directory, Forms, List and ASP.NET Membership are used for different repositories, the resource fields like Email-Id, Mobile Provider, and Mobile No used for SMS configurations should have same field name across all resource providers. For example, let's say fields like Email, Mobile phone provider and SMS Number is used in Active Director provider as Email-Id, Mobile Provider, and Mobile No respectively, then In Forms provider/SkeltaList provider/ASP.NET membership provider also should have these fields like Email-Id, Mobile Provider, and Mobile No to support SMS functionality.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Notification Contents

The properties in the Notification Contents group have been described below. You can use these properties to specify the content of the notification.

Sender - This property is used to set the sender's phone number.

Property Type: Optional (This property needs to be set only if required.)

RecipientResource - This property is used to build a query for selecting the resources for sending SMS. The phone number of the resource will be taken from the resource table. The phone number fields should be configured in the CommunicationConfig.xml.

Property Type: Mandatory (This property alternates with the 'Recipient' property. One of these properties must be set if the SMS is to be sent to any resource(s).)

Steps to set the RecipientResource property

1. To specify the actors to receive the SMS, click on the button for the 'RecipientResource' property to open the property window.
2. In this window build an expression to select the actors using the resource properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Send SMS: RecipientResource for a detailed description of the RecipientResource property window.

Recipient - This property is used to specify any Phone Number for sending the SMS.

Property Type: Mandatory (This property alternates with the 'RecipientResource' property. One of these properties must be set if the SMS is to be sent to any resource(s).)

Message - This property is used to enter the message of the SMS.

Property Type: Mandatory (This property needs to be set.)

Steps to set the Message property

1. To specify the message, click the button for the 'Message' property to open the property window.
2. In this window, enter the message text directly in the text editor.
3. If you want to customize the message you can do so using the **Field** controls to add resource property values.
4. Click **Append** add the expression to the text editor.
5. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

undo and redo changes, zoom in and zoom out the view as well as print the text.

6. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
7. After adding the subject, click **Update** to save it.
8. Click **Close** to close the window without saving the message.

See Send SMS: Message for a detailed description of the Message property window.

Retry - This property allows the user to enable the activity to retry up to three times if the first attempt to execute fails. By default it is set to "Yes."

Property Type: Optional (This property needs to be set.)

Activity outputs:

The SendSMS Activity does not return any value.

Exception Behavior

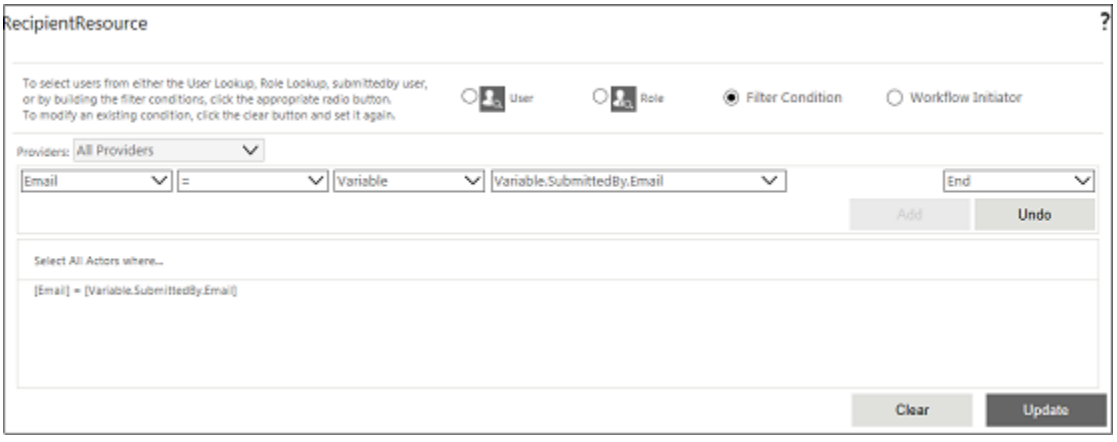
Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Note: Setting value for Set Variables and Contents from SMS property in Wait for SMS activity, through XML variable is not supported.

Send SMS: RecipientResource

You can set the actors to whom the SMS is to be sent in this property window. To open this window, click the button for the 'RecipientResource' property.



The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Different Methods of Assigning an Actor

- **Assigning an Actor by building an expression:**

Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.

- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

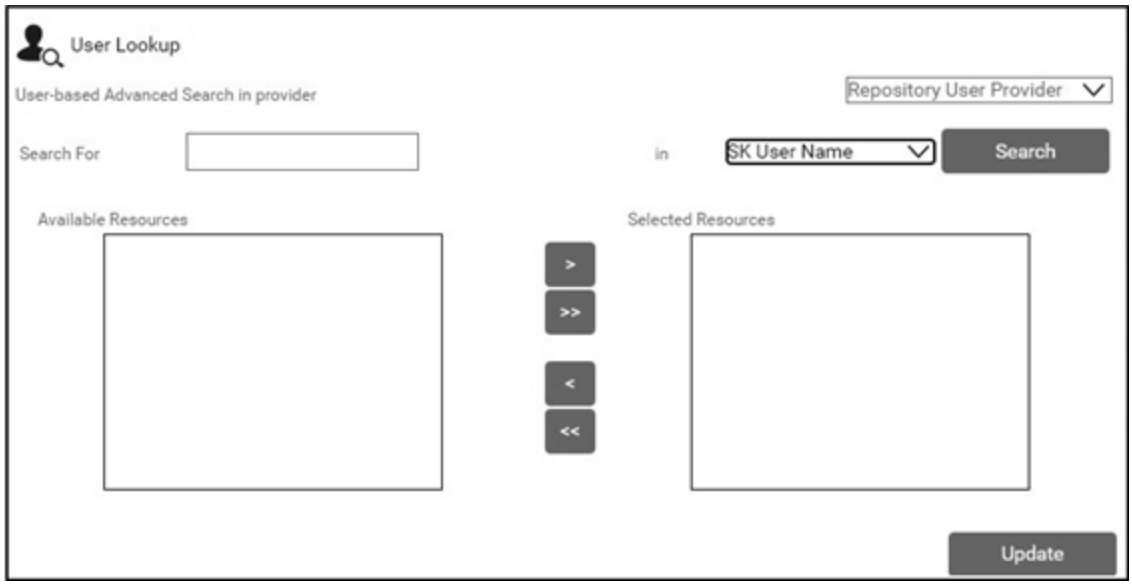
- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

- **Assigning an Actor through User Lookup**

Through the User Lookup you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup

1. Select the **User Lookup** option.
2. The **User Lookup** dialog is displayed, search and select the required actor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.

   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

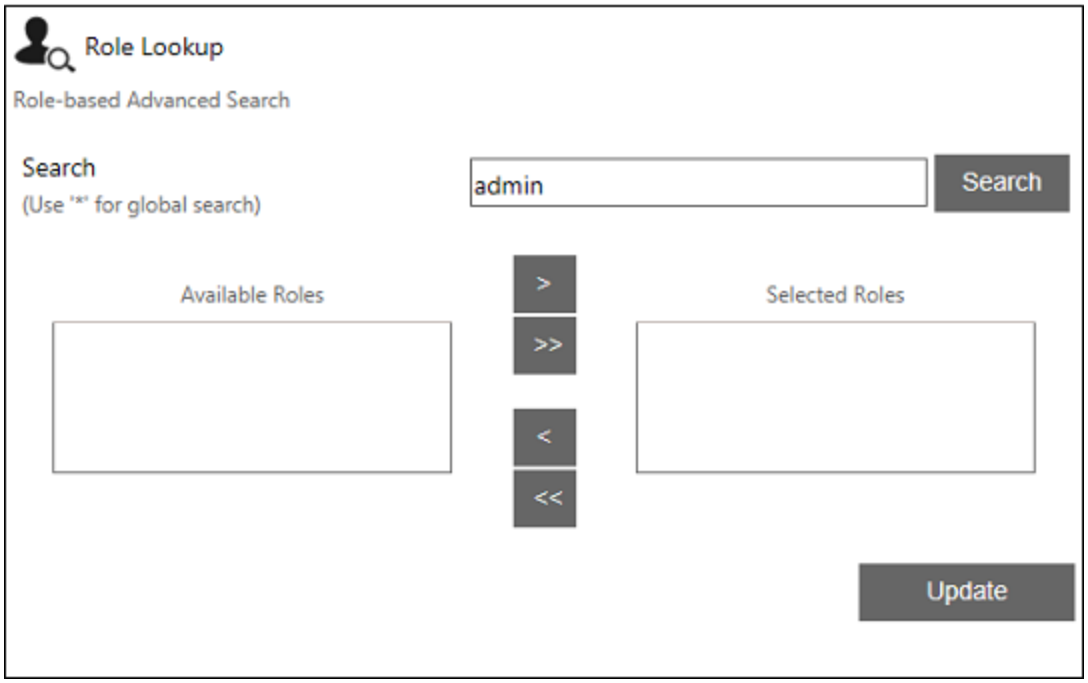
4. Click **Update** in the Assign Actor(s) window to update the selected actor(s) in the **Assign Actor** property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

- **Assigning an Actor through Role Lookup**

Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup

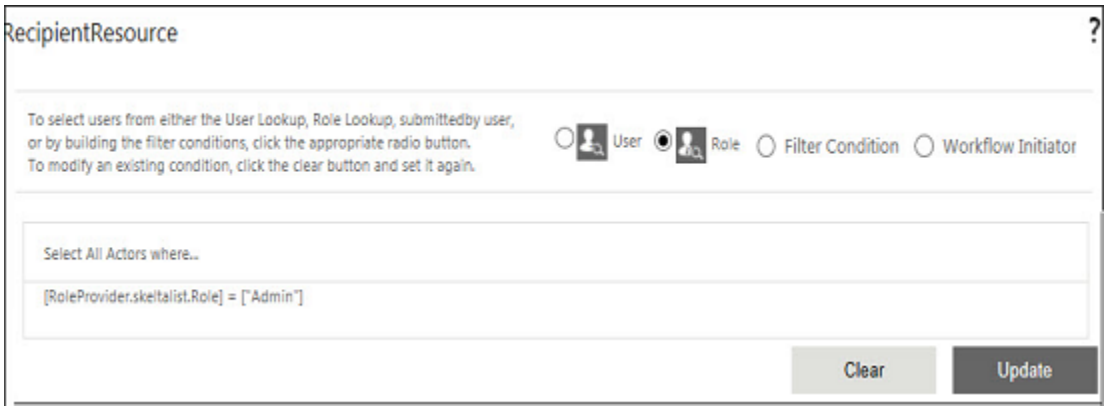
1. Select the **Role Lookup** option.
2. The **Role Lookup** dialog is displayed, search and select for the required role.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.

   In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:



4. Click **Update** in the Assign Actor(s) window to update the Assign Actor property.

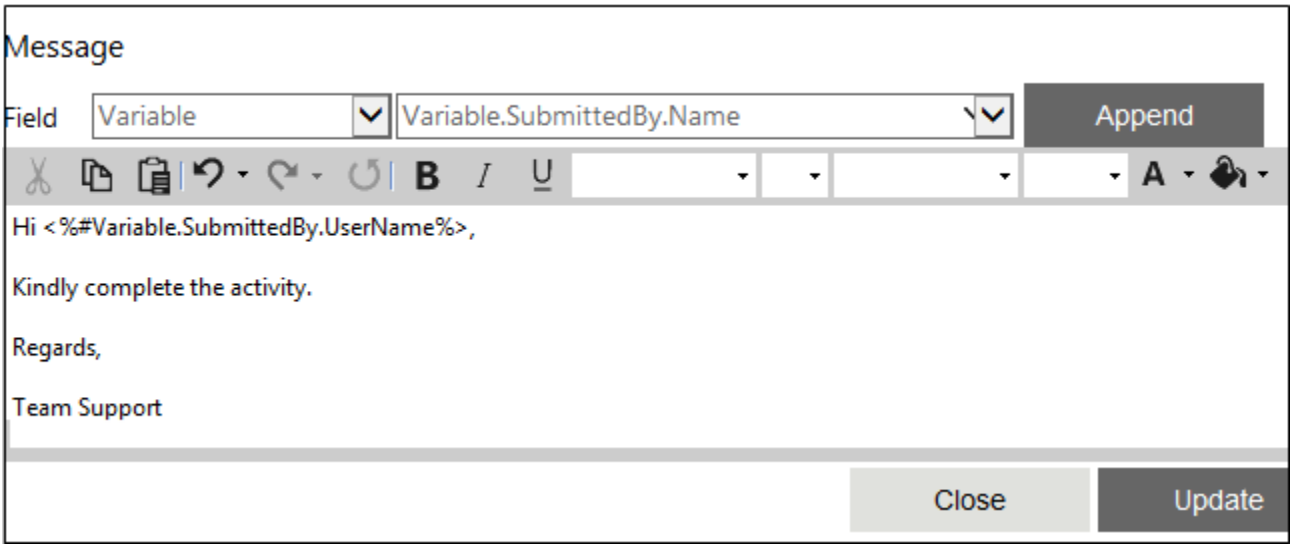
- **Assigning an Actor through Workflow Initiator**

  Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Send SMS: Message

You can specify a message for the SMS in this property window. To open this window, click the button for the 'Message' property.



The window has a text editor to enter the message text. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the message. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the message text in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the message.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The text tool bar above the text editor allows you to manage and format the message text. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the body
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the message.
- **Close** - You can click this button to close this window.

Wait for Email Activity

Activity description:

The Wait for Email Activity can be used to wait for a number of emails for a particular time period, both of which can be set in the activity's properties. It can also be used to filter incoming mails using specific conditions set in the properties. It can store emails (completely or partially) into specified variables. Options to save attachments, including the location to save, can also be specified. Depending on different property settings the activity will give different outputs.

See Localizing the response sent for an invalid email to the Communication Engine for more information on localizing an email.

Variables for capturing EmailIds

To get Email Id of the last received mail, use the AVEVA Work Tasks System variable. The variable name has to be prefixed with **"@"+CurrActivity.Name + "^MailID".**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For example, If activity name is Waitforemail1, then variable name should be "@ Waitforemail1^MailID".

Note: The use of AVEVA Work Tasks System variable CurrActivity.Name + "^EMailID" has been retained to ensure backward compatibility. For example, If activity name is Waitforemail1, then variable name should be "Waitforemail1^EMailID".

Activity Properties:

The Wait for Email activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Wait for Email activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Filter Expression - This property is used to filter the incoming email. It will add an email filter. On execution, the SMTP Notification Engine will be notified about the newly added filter so that it will search for this pattern in the next email.

Property Type: Mandatory (This property must be set if the action is to be assigned to any resource(s).)

Steps to set the Filter Expression property

1. To set the condition to filter the incoming emails, click the button for the 'Filter Expression' property to open the property window.
2. In this window build an expression to set the filter. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression in the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

See Wait for Email: Filter Expression for a detailed description of the Filter Expression property window.

Valid for Emails - This property is used to specify the number of emails for which the action should wait. It indicates that the action should wait till the specified number of emails are received.

Property Type: Optional (This property needs to be set only if required.)

Set Variables And Contents from Email - This property is used for assigning the email partly/ fully to a variable or content.

Property Type: Optional (This property needs to be set only if required.)

Steps to set the Set Variables And Contents from Email property

1. To set the filtered email can be to variables and contents, click the button for the 'Set Variables And Contents from Email' property to open the property window.
2. In this window, first select the variable from the **Variable** drop down.
3. Next specify the **search criteria** in the Search criteria field. You can either enter the regular expression directly or click on the **Generate RegExp for Selected Row** button to generate a regular expression.
4. If you click the **Generate RegExp for Selected Row** button it displays a set of fields and buttons at the bottom of the property window.
5. Specify the start string for the regular expression in the **Start String** field and **Start** and **End** delimiters in the corresponding text boxes of the **Delimiters** fields.
6. After specifying these values, click **Generate** to generate the regular expression.
7. Click **Hide** to hide these fields.
8. Specify where to search in by selecting an appropriate option from the Search in drop down. You can set the action to search in the entire email, the subject, and the attachments among other options.
9. Specify the name of a group to which the variable should belong in the **Set variable in the named group** field.
10. Click the **New row** tab to add the current search expression and display a blank row to add another expression.
11. If you need to remove any search expressions, you can select them and click **Remove selected**.
12. After specifying all the search parameters, click **Update** to save the new settings.

See Wait for Email: Set Variables And Contents from Email from Email for a detailed description of the Set Variables And Contents from Email property window.

Attachment Save Options - This property is used for setting the attachment saving options. The mail attachments can saved according to specified options. Filter attachments to save by searching for a string in the file name or file content.

Note:
- Attaching an email message as an attachment is not supported.
- Supported file types for attachment include .au,.avi,.bas,.bmp,.cdr,.drw,.dwg,.dxf,.eml,.eps,.gif,.ico,.img,.jpeg,.jpg,.log,.mid,.midi,.mov,.movie,.mp1,.mp2,.mp3,.mpeg, and .txt.

Property Type: Mandatory (This property needs to be set.)

Steps to set the Attachment Save Options property

1. To specify the attachment save options, click the button for the 'Attachment Save Options' property to open the property window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Select the save option from the first drop down.
   a. If you select "All Files", it will save all the attachment files
   b. If you select "File Names" or "File Contents", it will save either the specified files or the specified contents of the attachment file. This will display the corresponding fields/drop downs to specify the parameters.
3. Select the type of value (Value, Variable, Content) from the second drop down, and specify the corresponding value for the property from the next text box (if "Value" is selected) or drop down.
4. Click **Update** to save the new option.

Note: In user specified location, you have to append the mail id i.e, if user specified location is "C:\SkeltaMailAttachments\", then actual location will be "C:\SkeltaMailAttachments\<mailId>" .

Example

Location : c:\Temp\

Actual: c:\Temp\123 –(123 is last mailid)

Location : c:\Temp

Actual: c:\Temp123 –(123 is last mailid)

See Wait for Email: Attachment Save Options for a detailed description of the Attachment Save Options property window.

Attachment Save Location - This property is used to specify the location in which to save the attachments.

Property Type: Optional (This property needs to be set only if necessary.)

Steps to set the Attachment Save Location property

1. To specify the location to save the attachment, click the button for the 'Attachment Save Location' property to open the property window.
2. Select the protocol to be used to save the file in the location from the 'Protocol' property.
3. Specify the path of the location in the Location property. Build an expression to assign the location details by selecting the operator, type of value (Value, Variable, Content, XMLVariables), and the value for the property from the corresponding drop downs.
4. In the same way, you can assign a search pattern to filter the type of file to be copied or transferred as the value for the **Pattern** field. For example, you can specify "*.doc" as the filter pattern. This will select only files with the extension .doc for copying or transferring.
5. You can also assign the user name and password of the source file/directory in the **User Name** and **Password** fields respectively, by following the same procedure that is used in step 3.
6. In addition, you can assign the domain name and port number of the source in the **DomainName** and **PortNumber** fields respectively, by following the same procedure that is used in step 3.
7. After adding the parameter, click **Submit** to save it.

See Wait for Email: Attachment Save Location for a detailed description of the Attachment Save Location property window.

Escalation & Timeout

The properties in the Escalation & Timeout group have been described below. You can use these properties to specify escalation and timeout settings for the activity.

Timeout Warning - This property is used for setting the Timeout warning.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if required.)

Timeout for Action - This property is used for setting timeouts for this action. If specified, the action will be timed out after the specified time span.

Property Type: Optional (This property needs to be set only if required.)

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Wait for Email Activity returns the following values -

Timeout Warning: Displayed when a warning for timeout is to be generated

Email Received(Waiting): Displayed when an email matching the specified filter condition is received

Timeout: Displayed when the activity has timed out

Blank Output: Displayed when the activity is complete

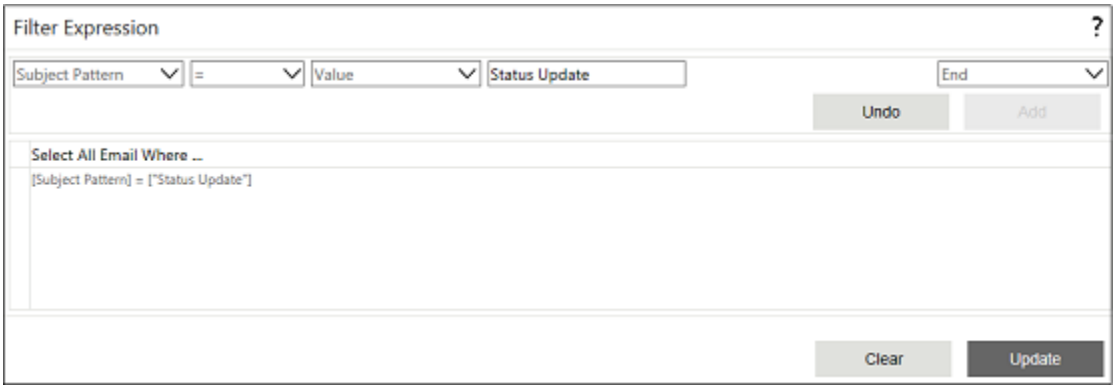
Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Wait for Email: Filter Expression

You can set the filter expression to filter the incoming email in this property window. To open this window, click the button for the **'Filter Expression'** property.



The window has a set of drop downs at the top to select the different parameters required to build the filter expression. You can also find related buttons next to the drop downs.

- The Attribute drop down - You can select the attribute to be used to filter the email, e.g., "Subject Pattern",

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

in this drop down.

- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down. If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click on this button to add the new expression to the text area.
- **Undo** - You can click on this button to remove a condition you have added to the expression.

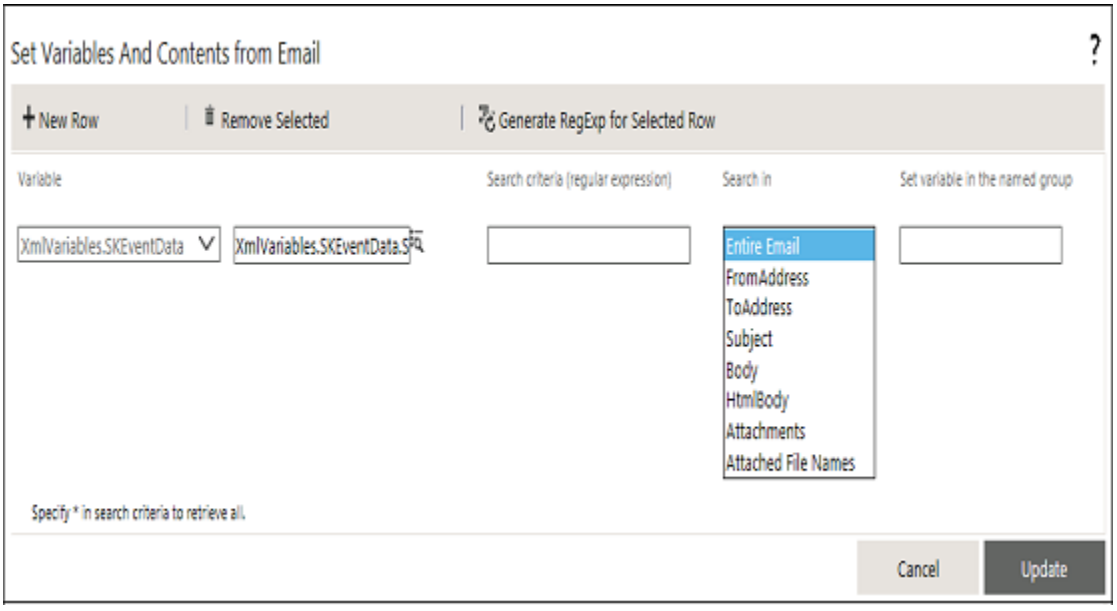
At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

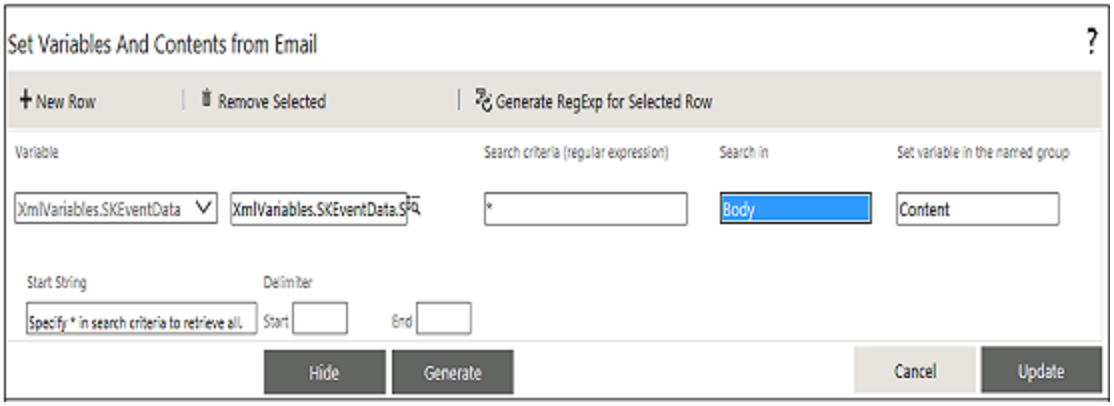
Wait for Email: Set Variables And Contents from Email

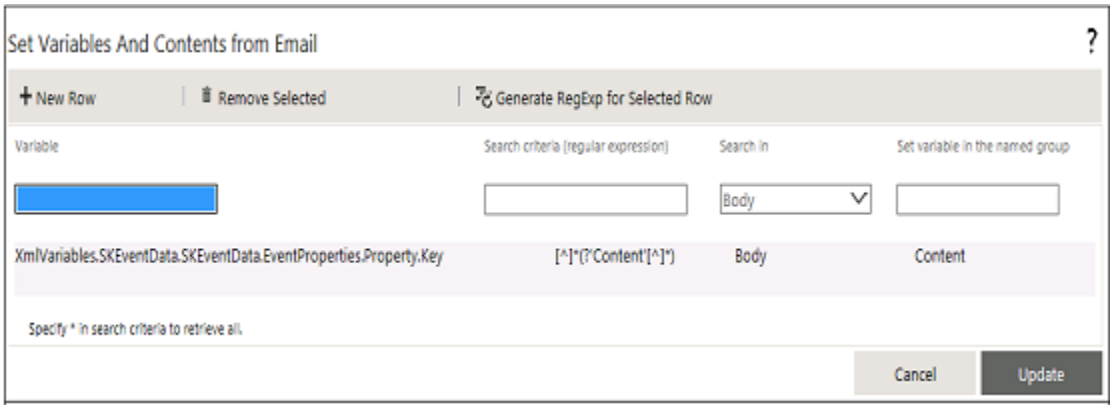
Set Variables And Contents from Email Property window

You can set the filter expression to filter the incoming email in this property window. To open this window, click the button for the 'Set Variables And Contents from Email' property.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.





The window has two tabs at the top to manage the rows of variable settings in the grid.

- **New Row** - You can click this tab to add the newly entered variable/content settings and display a new row to the grid where you can add another setting
- **Remove Selected** - You can click this tab to remove the variable/content selected in the grid
- **Generate RegExp for Selected Row** - You can click this tab to generate a regular expression for the search criteria. This displays the fields shown in the bottom half of the property window image. These fields are described later in the topic.

The grid has the following parameters that you need to specify for the new variable.

- **Variable** - You can select the variable, to which you wish to assign the data in the attachment, from the drop down for this field.
- **Search criteria (regular expression)** - You can enter a regular expression in this field to be used to search the email for data to be assigned to the variable/ content. Alternatively, you can click on the Generate RegExp for Selected Row tab to generate a regular expression.
- **Search in** - You can specify which part of the email should be searched by selecting an appropriate option from the drop down for this field. You can set the activity to search in the entire email, the subject, and the attachments among other options.
- **Set variable in the named group** - You can enter the name of a group, to which you can add the variable as a member, in this field.

Tip: You can specify "*" in the Search Criteria field to retrieve all the data.

When you click on the Generate RegExp for Selected Row it displays the following controls to generate the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

regular expression.

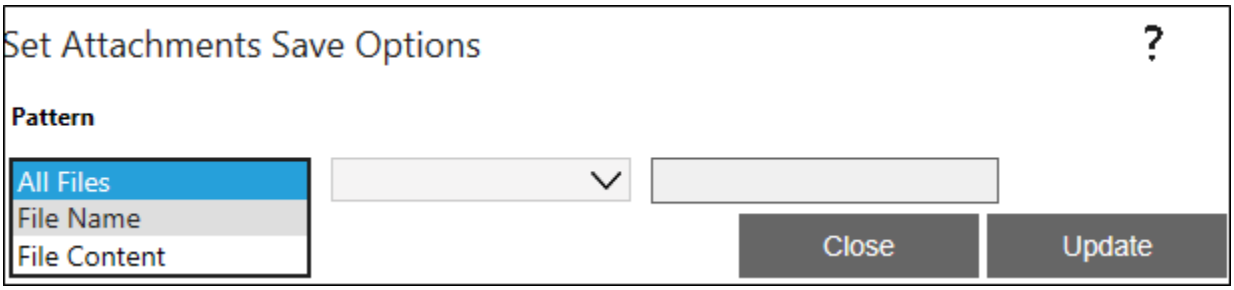
- **Start String** - You can specify the start string for the regular expression in this field.
- **Delimiter** - Start - You can enter the start delimiter for the regular expression in this field.
- **Delimiter** - End - You can enter the end delimiter for the regular expression in this field.
- **Generate** - You can click this button, after specifying the above values, to generate the regular expression.
- **Hide** - You can click this button to hide these fields.

There are two buttons at the bottom of the screen.

- **Update** - You can click this button to save the newly added variable/content settings.
- **Cancel** - You can click this to close the property window without saving the newly added settings.

Wait for Email: Attachment Save Options

You can set the options to save the email attachments in this property window. To open this window, click the button for the 'Attachment Save Options' property.



In user specified location, you have to append the mail id i.e, if user specified location is "C:\SkeltaMailAttachements\", then actual location will be "C:\SkeltaMailAttachements\<mailId>" .

Example

Location : c:\Temp\

Actual: c:\Temp\123 –(123 is last mailid)

This window has the following fields to specify the pattern to save attachments.

- The attachment drop-down field can be used to specify the attachment option by selecting the appropriate property from the drop-down field. You can choose to save all attachment files, specific files or file contents.
- The value type drop-down field can be used to specify the value type to assign to the option selected in the attachment drop-down field.
- The value text box/ drop-down field can be used to specify the pattern which matches with the "File Name" or "File Content" property selected in the attachment drop-down field. The control displayed here depends on the option selected in the value drop-down field. If you select "Value", a text box is displayed here, where you can enter the pattern directly. If you select Variable or Content, then a drop-down is displayed with a list of related collections. You can select the appropriate option from this drop-down field.

There are two buttons at the bottom of the screen.

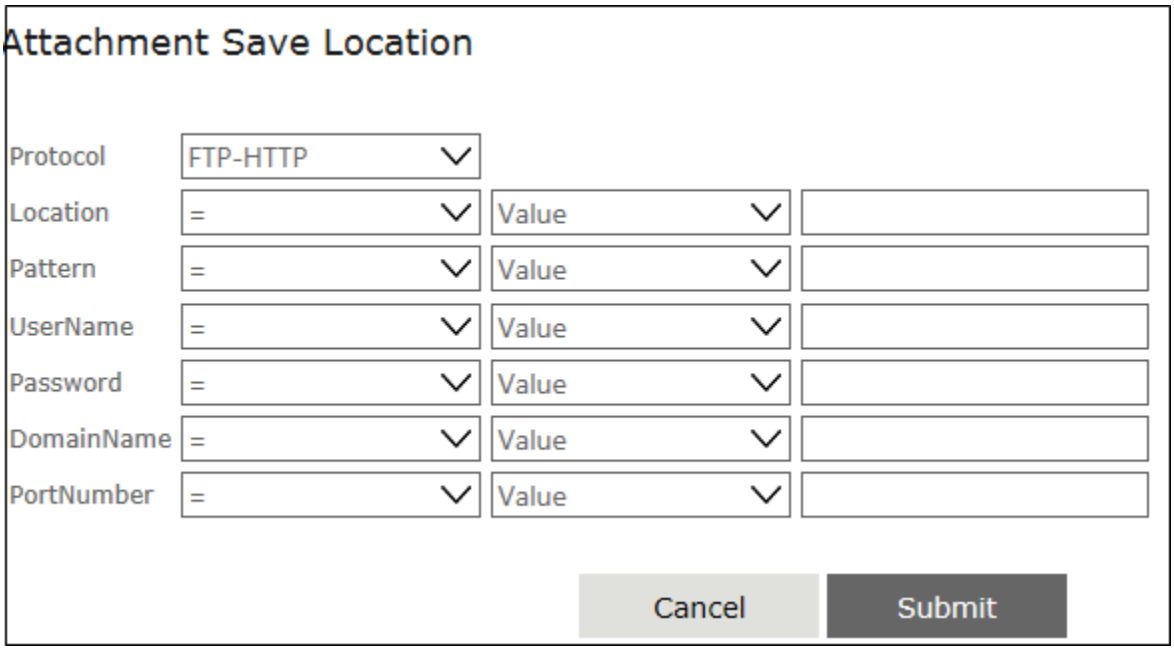
- **Update** - You can click this button to save the newly added pattern settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Close** - You can click this to close the property window without saving the newly added settings.

Wait for Email: Attachment Save Location

You can specify the details of the location to which the email attachment should be stored in this property window. To open this window, click the button for the **'Attachment Save Location'** property.



This window has the **Protocol** drop down where you can select the file transfer protocol.

Note: Email attachments can be stored only in *File* and *FTP* paths.

Below the **Protocol** drop down, it has controls to specify the following details.

- **Location** - You can specify the URI/UNC of the file or directory here.
- **Pattern** - You can specify the pattern to decide the file types to be saved like *.doc,*.xls, etc. in this field.
- **UserName** - You can specify the user name for the attachment save location here.
- **Password** - You can specify the password for the attachment save location here.
- **Domain** - You can specify the domain/machine of the attachment save location here.
- **PortNumber** - You can specify the port number of the attachment save location server.

Each of the above parameter fields has the following controls.

- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the parameter in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the parameter. The control displayed depends on the option selected in the preceding drop down. If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

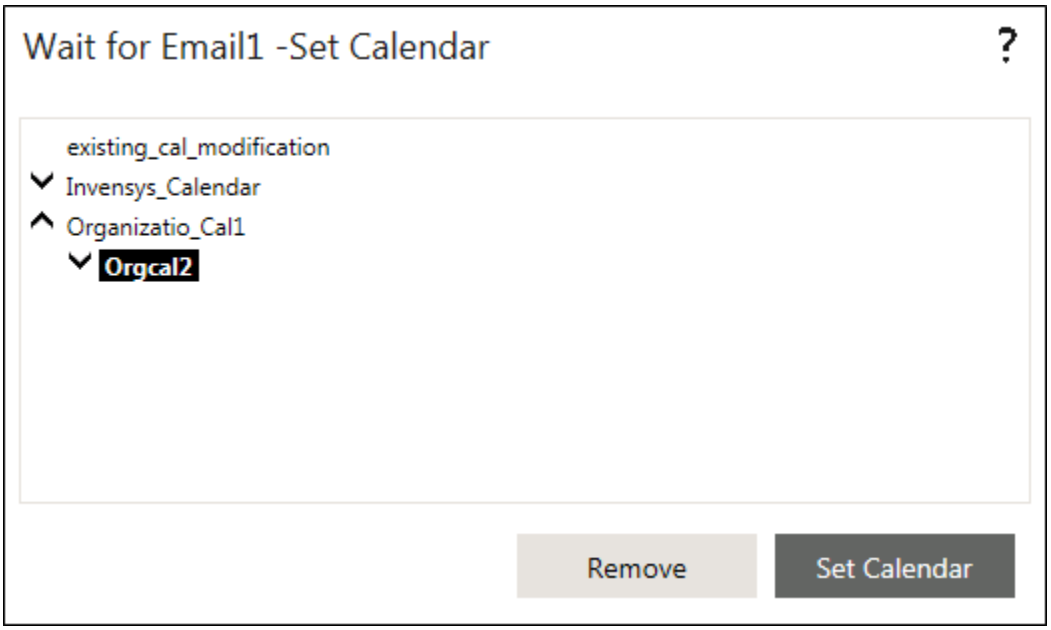
At the bottom of the window you will find the following buttons:

- **Submit** - You can click this button to save the parameters that have been specified.
- **Cancel** - You can click this button to cancel the settings and close this window.

Wait for Email: Set Variables And Contents from Email

Set Calendar for Timeout Property Window

You can set a Calendar to calculate the timeout for the Wait for Email activity in this property window. To access this window, click the button for the 'Set Calendar for Timeout' property.



This window lists the available calendars in a tree structure, with lower-level calendars nested within the respective top-level calendar. You can click a calendar to select it for setting.

There are two buttons in this window:

- **Set Calendar**: Set the selected calendar.
- **Remove**: Remove the selected calendar.

Wait for SMS Activity

Activity description:

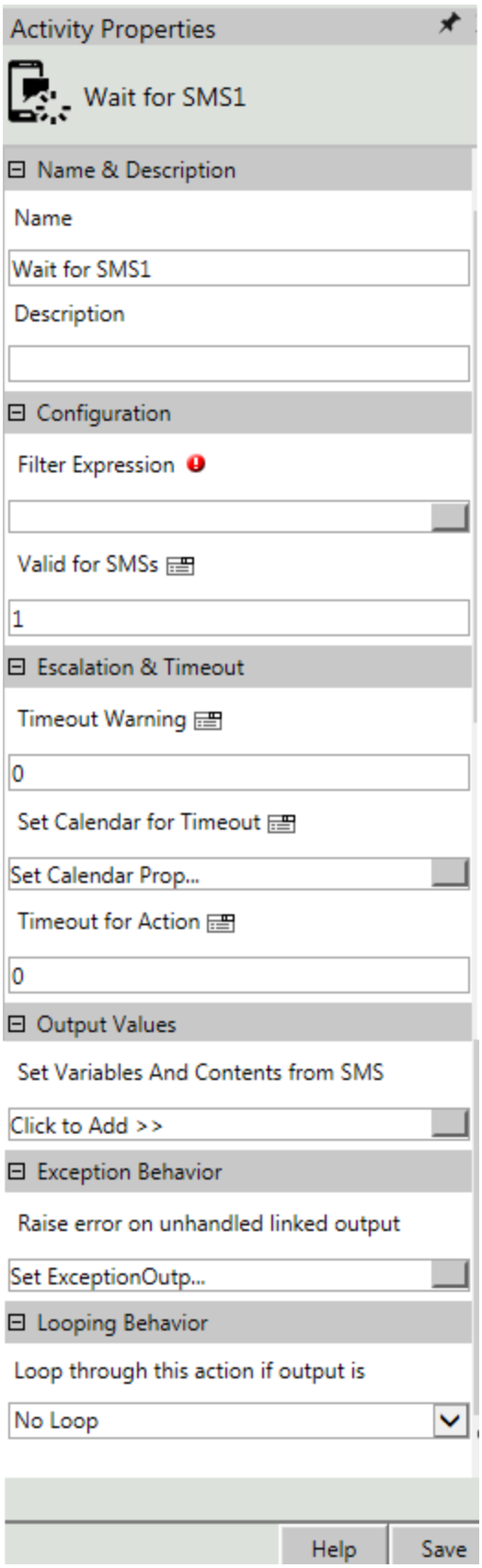
The Wait for SMS Activity can be used to wait for a number of SMSs for a particular time period, both of which can be set in the activity's properties. It can store SMSs (completely or partially) into specified variables. Depending on different property settings the activity will give different outputs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties:

The Wait for SMS activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the tool bar or by selecting the appropriate option from the right click menu for the Wait for SMS activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Filter Expression - This property is used to filter the incoming SMS. It will add an SMS filter. On execution, the SMTP Notification Engine will be notified about the newly added filter so that it will search for this pattern in the next SMS.

Property Type: Mandatory (This property must be set if the activity is to be assigned to any resource(s).)

Steps to set the Filter Expression property

1. To set the condition to filter the incoming SMS, click the button for the 'Filter Expression' property to open the property window.
2. In this window, build an expression to set the filter. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression in the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Wait for SMS: Filter Expression for a detailed description of the Filter Expression property window.

Valid for SMSs - This property is used to specify the number of SMSs for which the activity should wait. It indicates that the activity should wait till the specified number of SMSs are received.

Property Type: Optional (This property needs to be set only if required.)

Escalation & Timeout

The properties in the Escalation & Timeout group have been described below. You can use these properties to specify escalation and timeout settings for the activity.

Timeout Warning - This property is used for setting the Timeout warning.

Property Type: Optional (This property needs to be set only if required.)

Timeout for Action - This property is used for setting timeouts for this activity. If specified, the activity will be timed out after the specified time span.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if required.)

Output Values

The properties in the Output Values group have been described below. You can use these properties to specify the input values for the activity.

Set Variables And Contents from SMS - This property is used for assigning the SMS partly/fully to a variable or content.

Property Type: Optional (This property needs to be set only if required.)

Steps to set the Set Variables And Contents from SMS property

1. To set the filtered email can be to variables and contents, click the button for the 'Set Variables And Contents from Email' property to open the property window.
2. In this window, first select the variable from the **Variable** drop down.
3. Next specify the search criteria in the **Search criteria** field. You can either enter the regular expression directly or click on the **Generate RegExp for Selected Row** button to generate a regular expression.
4. If you click the **Generate RegExp for Selected Row** button it displays a set of fields and buttons at the bottom of the property window.
5. Specify the start string for the regular expression in the **Start String** field and **Start** and **End** delimiters in the corresponding text boxes of the **Delimiters** fields.
6. After specifying these values, click **Generate** to generate the regular expression.
7. Click **Hide** to hide these fields.
8. Specify where to search in by selecting an appropriate option from the **Search in** drop down. You can set the action to search in the entire email, the subject, and the attachments among other options.
9. Specify the name of a group to which the variable should belong in the **Set variable in the named group** field.
10. Click the **New row** tab to add the current search expression and display a blank row to add another expression.
11. If you need to remove any search expressions, you can select them and click **Remove selected**.
12. After specifying all the search parameters, click **Update** to save the new settings.

See Wait for SMS: Set Variables and Contents from SMS for a detailed description of the Set Variables And Contents from SMS property window.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Wait for SMS Activity returns the following values -

Timeout Warning: Displayed when a warning for timeout is to be generated.

Email Received(Waiting): Displayed when an SMS matching the specified filter condition is received

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Timeout: Displayed when the activity has timed out

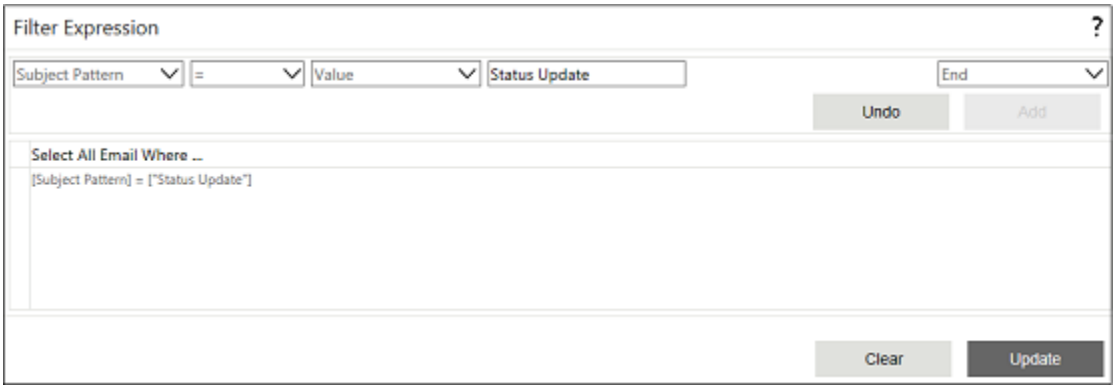
Exception Behavior

Each activity has the **Raise Error on Unhandled Linked Output** property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Wait for SMS: Filter Expression

You can set the filter expression to filter the incoming SMS in this property window. To open this window, click the button for the 'Filter Expression' property.



The window has a set of drop downs at the top to select the different parameters required to build the filter expression. You can also find related buttons next to the drop downs.

- The Attribute drop down - You can select the attribute to be used to filter the SMS, e.g., "Subject Pattern", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down. If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

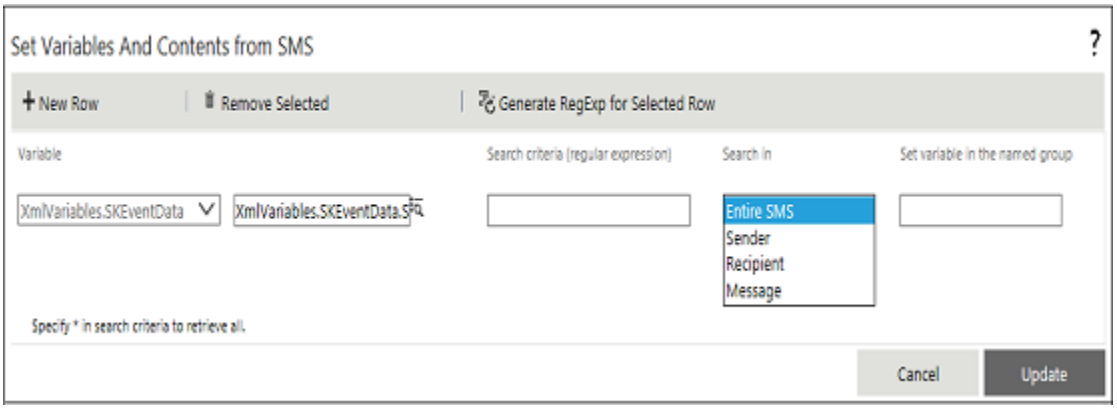
to build a new expression.

- **Update** - You can click this button to save the expression that has been added to the text area.

Wait for SMS: Set Variables and Contents from SMS

Set Variables And Contents from SMS Property window

You can set the filter expression to filter the incoming SMS in this property window. To open this window, click the button for the **'Set Variables And Contents from SMS'** property.



The window has two tabs at the top to manage the rows of variable settings in the grid.

- **New Row** - You can click this tab to add the newly entered variable/content settings and display a new row to the grid where you can add another setting.
- **Remove Selected** - You can click on this tab to remove the variable/content selected in the grid.
- **Generate RegExp for Selected Row** - You can click on this tab to generate a regular expression for the search criteria. This displays the fields shown in the bottom half of the property window image. These fields are described later in the topic.

The grid has the following parameters that you need to specify for the new variable.

- **Variable** - You can select the variable, to which you wish to assign the data in the attachment, from the drop down for this field.
- **Search criteria (regular expression)** - You can enter a regular expression in this field to be used to search the SMS for data to be assigned to the variable/ content. Alternatively, you can click on the **Generate RegExp for Selected Row** tab to generate a regular expression.
- **Search in** - You can specify which part of the SMS should be searched by selecting an appropriate option from the drop down for this field. You can set the activity to search in the entire SMS, Sender, Recipient, and the Message.
- **Set variable in the named group** - You can enter the name of a group, to which you can add the variable as a member, in this field.

Tip: You can specify "*" in the Search Criteria field to retrieve all the data.

When you click on the **Generate RegExp for Selected Row** it displays the following controls to generate the regular expression.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Start String** - You can specify the start string for the regular expression in this field.
- **Delimiter** - **Start** - You can enter the start delimiter for the regular expression in this field.
- **Delimiter - End** - You can enter the end delimiter for the regular expression in this field.
- **Generate** - You can click this button, after specifying the above values, to generate the regular expression.
- **Hide** - You can click this button to hide these fields.

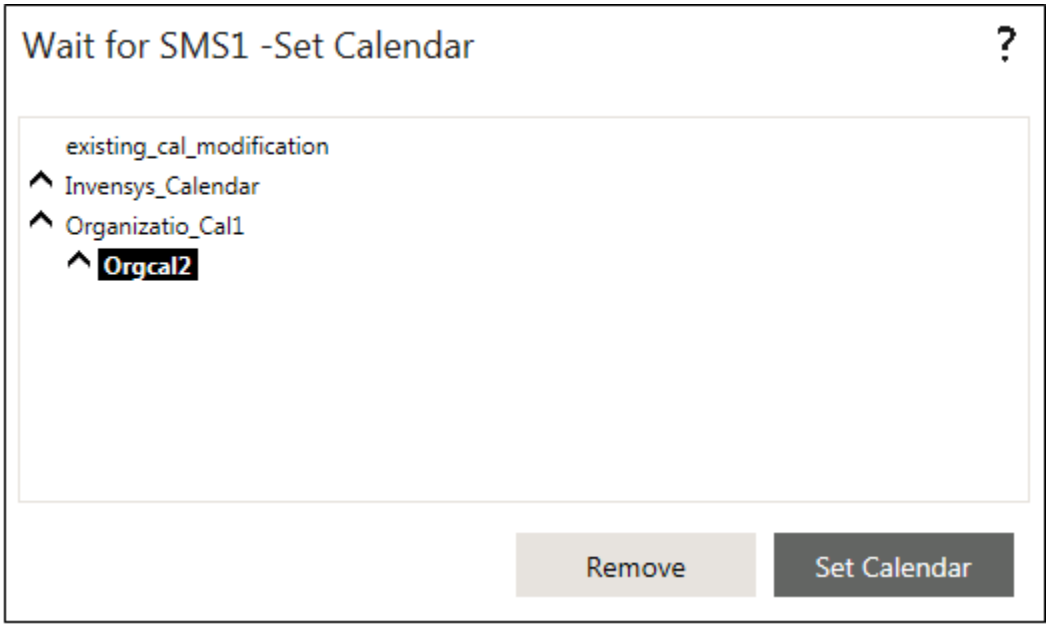
There are two buttons at the bottom of the screen.

- **Update** - You can click this button to save the newly added variable/ content settings.
- **Cancel** - You can click this to close the property window without saving the newly added settings.

Wait for SMS: Set Calendar for Timeout

Set Calendar for Timeout Property Window

You can set a Calendar to calculate the timeout for the Wait for SMS activity in this property window. To access this window, click the button for the 'Set Calendar for Timeout' property.



This window lists the available calendars in a tree structure, with lower-level calendars nested within the respective top-level calendar. You can click a calendar to select it for setting.

There are two buttons in this window:

- **Set Calendar**: Set the selected calendar.
- **Remove**: Remove the selected calendar

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

AVEVA Work Tasks Pro Push Notification Activity

Activity Description

The AVEVA Work Tasks Pro Push Notification activity is used to send notifications to the AVEVA Work Tasks Pro application. When this activity is added and configured in a workflow along with other activities which need human interaction, a notification is sent to the actor intimating the actor that action is required on the assigned task.

Activity Properties

Name and Description

The properties in the Name & Description are used to specify the name and description for the activity.

Name: This property is used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description: This property is used to specify a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group allows you to configure the functionality of the activity.

Recipient Resource: This property is used to configure the resources who have to receive the notification. A maximum of 16 resources can be configured as recipients to receive notification. If more than 16 resources are configured, then the 16 active resources will receive the notifications.

You can also build an expression to configure the recipient, this expression will be evaluated at runtime and the recipient will be configured.

Property Type: Mandatory (This property needs be set to add the recipients.)

For more information on Recipient Resource, see Configure Recipient Resource.

Notification Type: This property allows you to configure custom notification type based on the device and operating system used.

Property Type: Mandatory (This property needs to be set to configure the type of notification specific to a device.)

For more information on Notification Type Setting, see Setting Notification Type.

Message: This property allows you to configure the content of the notification message.

You can also build an expression to configure the notification message. The message will be updated dynamically at runtime.

Property Type: Mandatory (This property must be set to configure the content of the notification.)

Retry on Failure?: When this property is set to **Yes** and the notification is not sent, attempts to re-send the notification will be made based on the value defined in the Retry Count property.

By default it is set to **Yes**.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Retry Count: Number of re-tries that needs to be attempted to send the notification. The value specified for this property should be from 1 to 10 only.

Property Type: Optional (This property needs to be set only if necessary.)

Activity output is successful: This property lets you decide when the activity is successful. The available options are **If notification is sent to at least one device** and **If notification is sent to all devices**.

- When **If notification is sent to at least one device** is selected, the activity will be successful only when the notification is sent to at least one device.
- When **If notification is sent to all devices** is selected, the activity will be successful only when the notification is sent to all the devices

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property. Selecting one of the property shows all the default mapped error outputs that are configured in red with the respective check boxes selected.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Reference Topics

- Set Notification Type
- Mapped Error Outputs
- AVEVA Work Tasks Pro Push Notification Activity Scenarios

Configure Recipient Resource

The **Recipient Resource** property allows you to configure the users who have to receive the notification. A maximum of 16 resources can be configured as recipients to receive notification. If more than 16 resources are configured, then the 16 active resources will receive the notifications.

Different Methods of Assigning an Recipient Resource

Assigning an Recipient by building an expression:

Filter Condition is the default option that is selected when the Recipient Resource is accessed. This option allows you to build an expression to identify the Recipient. The section below explains how you can choose different parameters required to build and expression.

- The Providers list - You can select the required provider from the Providers list.
- The Attribute list - You can select the attribute to be used to identify the Recipient. For example, "AdministratorId", in this list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The Operator list - You can select the operator for the expression here. E.g., "="
- The Value type list - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/list - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding list.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a list instead of the field. This list will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator list - You can select logical operators from this list to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

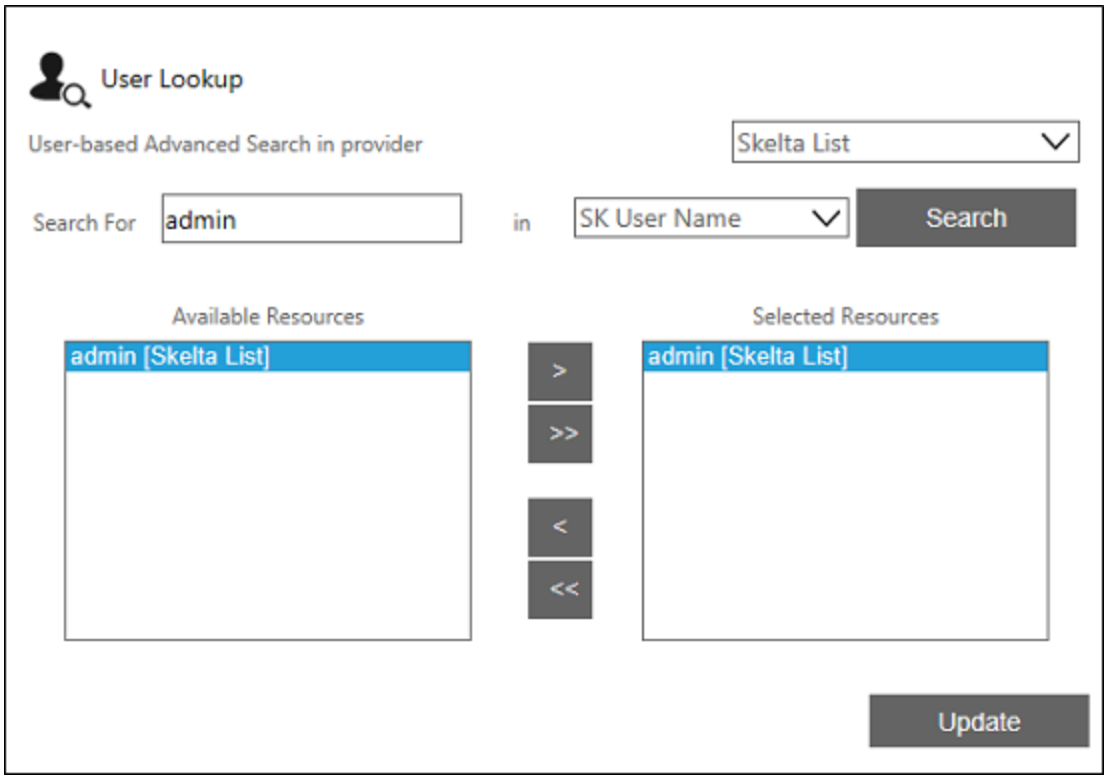
- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

Assign an Recipient Resource through User Lookup

Through the **User Lookup** you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an Recipient Resource through User Lookup

1. Select the **User Lookup** option. The **User Lookup** dialog is displayed.
2. Search and select the required user.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



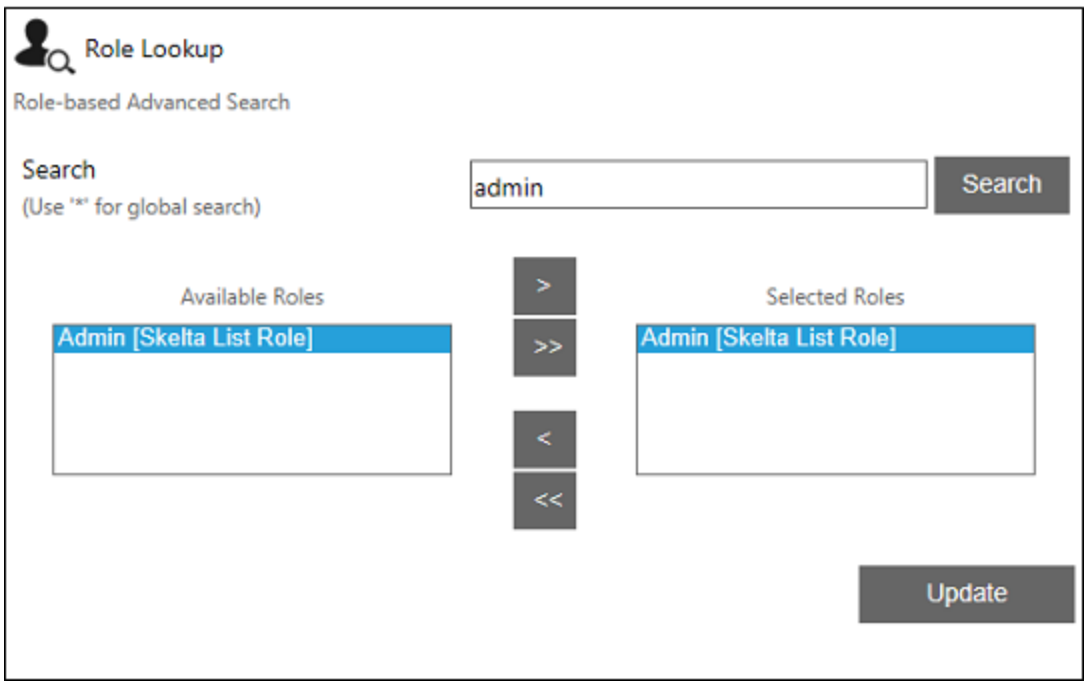
3. Click **Update** in the **User Lookup** dialog. In **Recipient Resource**, you will see that an expression is displayed.
4. Click **Update** in **Recipient Resource** to update the selected user in the **Recipient Resource** property.

Assign an Recipient Resource through Role Lookup

Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area.

1. Select the **Role Lookup** option. The **Role Lookup** dialog is displayed.
2. Search and select for the required role.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click **Update** in the **Role Lookup** dialog and verify the Recipient Resource. In **Recipient Resource**, you will see that an expression is displayed.
4. Click **Update** in **Recipient Resource** to update the **Recipient Resource** property.

Assign an Recipient Resource through Workflow Initiator

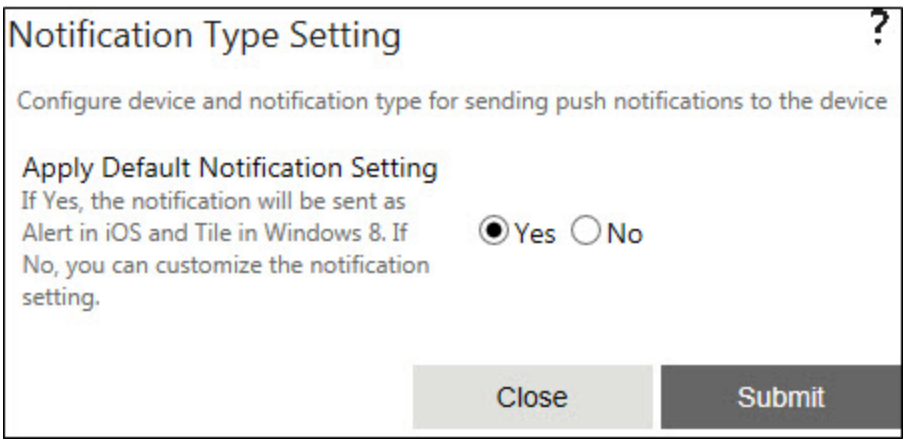
Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Recipient Resource.

Set Notification Type

In **AVEVA Work Tasks Pro Push Notification Activity**, you can configure different types of notification based on the device being used. You can choose to continue with the default notification type or customize the notification type.

To use the default notification type, set the option **'Apply Default Notification Setting'** as **Yes**. Default notification for iOS is **Alert** and for Windows 8.1 it is **Tile**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Configuring Custom Notification Type

1. Click the edit icon in **Notification Type** property. The **Notification Type Setting** is displayed.
2. Select **No** for the option **'Apply Default Notification Setting'**. The option of **Device and Notification Type** is now available for selection.

   The default selection for the option **'Apply Default Setting'** is **Yes**.
3. Select the operating system used on your device and then select the **Notification Type**.
   - iOS operating system, supports only the default notification specific to iOS.
   - For Windows 8.1 operating system, the available notification types are **Tile** and **Toast**
     - When **Tile** is selected the notifications are displayed on tile of the application
     - When **Toast** is selected the notifications are displayed as pop-up notifications

Note: Push Notification activity will continue to execute successfully even after unregistering for notification from the **Farm Configuration Wizard**. In such cases, unpublish the workflow, unregister for notification from the **Farm Configuration Wizard**, and then republish the workflow.

AVEVA Work Tasks Pro Push Notification Activity Scenarios

Activity in waiting state

The activity will be in waiting state, when the first attempt to send notification fails and the property **Retry on Failure** is set to **Yes**.

When the first attempt by the activity to send the notification fails, an entry is made into the database. This entry is read by the **ArchestrA Mobile Notification Service** which in turn attempts to re-send the notification based on the value entered in the **Retry Count** property.

The activity will be completed only after all the attempts to retry the notifications are made.

Activity output is successful when Recipient Resource is inactive/blocked/not logged in

When multiple recipient resources are configured in the AVEVA Work Tasks Pro Push Notification Activity and one or more recipient is inactive/blocked/not logged in, the activity will be considered as successful if the notification is sent to all other active recipients.

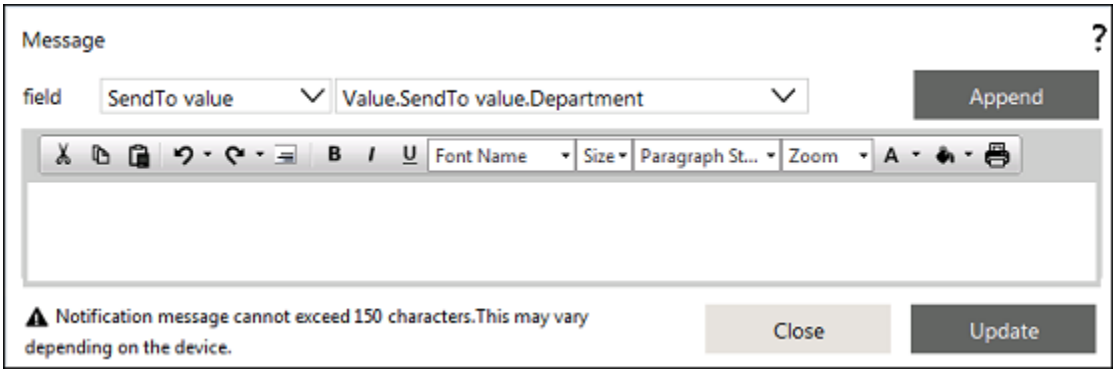
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Unsuccessful activity output scenarios:

- First attempt to send notification fails and **Retry on Failure** is set to **No**.
- When **Retry on Failure** is set to **Yes** and **Activity is successful** is set to **If notification is sent to at least one device:** If notification is not sent to any device even after consecutive attempts to resend notification based on the value specified in **Retry Count**.
- When **Retry on Failure** is set to **Yes** and **Activity is successful** is set to **If notification is sent to all devices:** If notification is not sent to one or more devices even after consecutive attempts to resend the notification based on the value specified in **Retry Count**.

Configure Notification Message

Use the **Message** property to configure the content of the notification message.



You can also build an expression to configure the notification message. The message will be updated dynamically at runtime.

For iOS devices, the notification message can have a maximum of 150 characters. In Windows 8.1 devices, if the notification message exceeds 208 characters, then the notification message is truncated. In both iOS and Windows 8.1 devices, the notification message is displayed in plain text, hence all html tags are removed.

BPMN Elements

The following activities are found under BPMN Elements:

- **Placeholder**: The Placeholder Activity is used as a placeholder for the BPMN events.
- **Gateway Placeholder**: The Gateway Placeholder Activity is used as a placeholder for the ccccN Gateway event.
- **Group Box**: The Group Box Activity is used as a marker for activity groups.
- **Intermediate Event Placeholder**: The Intermediate Event Placeholder Activity is used as a placeholder for the BPMN Intermediate events.
- **Lane**: The Lane Activity is used to create a swim-lane in the process model.
- **Pool**: The Pool Activity is used to create a Pool in the process model.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Gateway Placeholder

Activity description

The Gateway Placeholder Activity is used as a placeholder for the ccccN Gateway event.

Activity Properties

The Gateway Placeholder activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the context menu for the Gateway Placeholder activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Gateway Placeholder Activity does not return any value.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Group Box

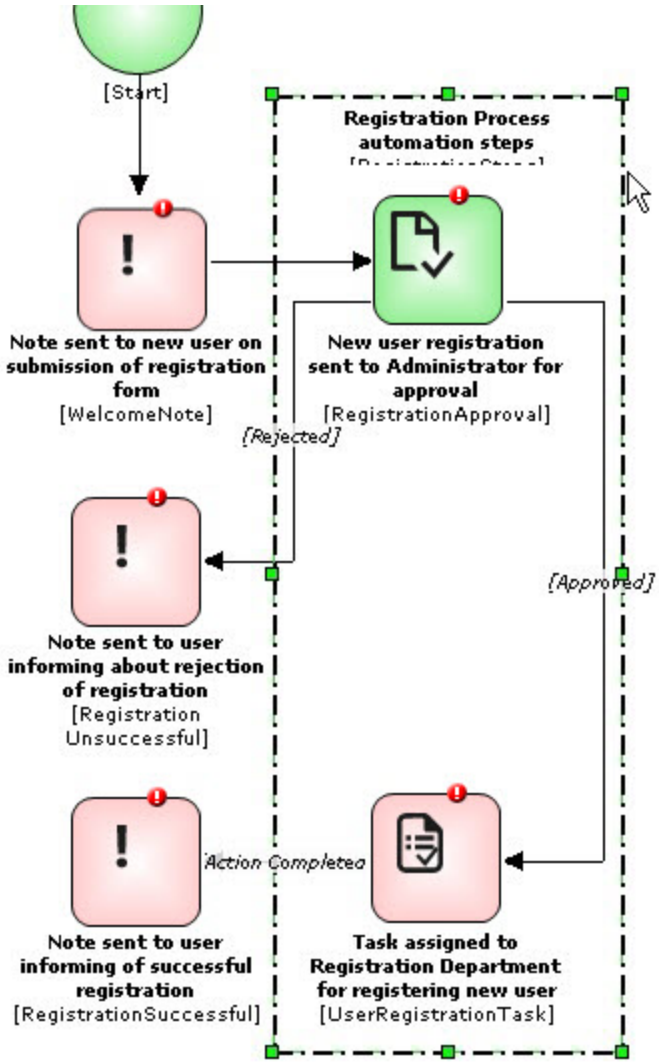
Activity description

The Group Box Activity is used as a marker for activity groups.

Application Scenarios

The Group Box activity is a useful tool for process modeling. It can be placed around a set of activities in the process to denote a group of related activities. This helps to streamline the process diagram. In the image shown below the Group Box identifies the registration processing steps in the diagram.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Sample Group Box

Activity Properties

The Group Box activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking the Activity Properties tool in the tool bar or by selecting the appropriate option from the context menu for the Group Box activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Group Box Activity does not return any value.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Intermediate Event Placeholder

Activity description

The Intermediate Event Placeholder Activity is used as a placeholder for the BPMN Intermediate events.

Activity Properties

The Intermediate Event Placeholder activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the tool bar or by selecting the appropriate option from the context menu for the Intermediate Event Placeholder activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Intermediate Event Placeholder Activity does not return any value.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

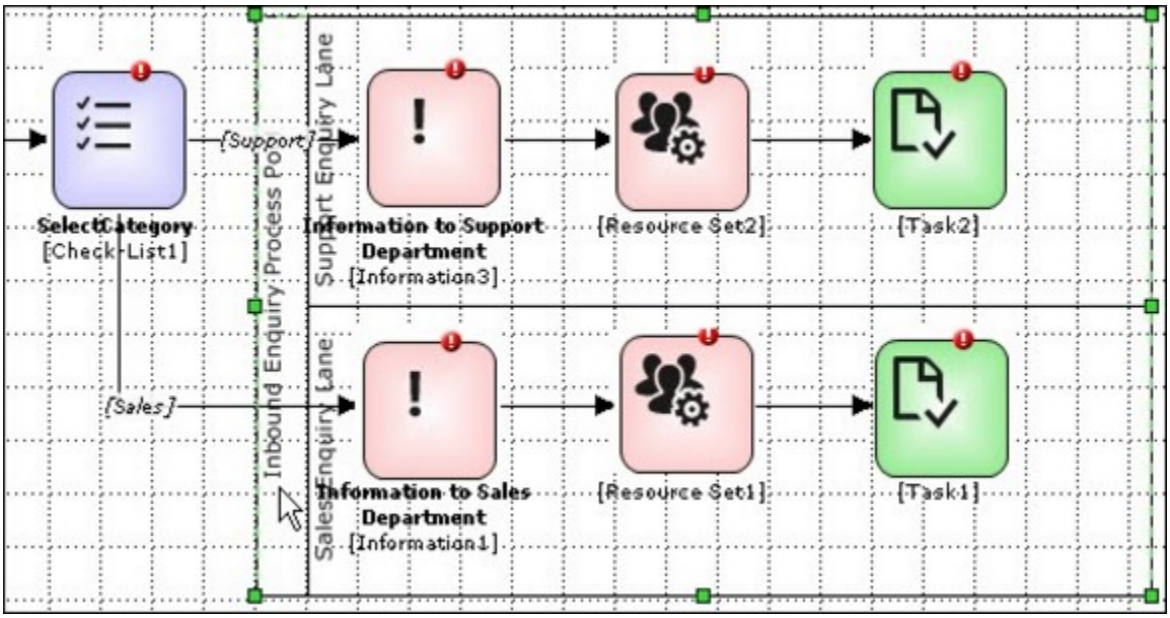
Lane

Activity description

The Lane Activity is used to create a swim-lane in the process model.

Application Scenarios

The Lane activity is a useful tool for process modeling. It can be placed around a set of activities in the process to denote a swim-lane in a process execution pool. This helps to streamline the process diagram. In the image shown below there are two lanes representing two types of enquiries handled in the process, i.e. Support Enquiry and Sales Enquiry.



Sample Lanes in a Pool

Activity Properties

The Lane activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking the Activity Properties tool in the tool bar or by selecting the appropriate option from the context menu for the Lane activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Lane Activity does not return any value.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Placeholder

Activity description

The Placeholder Activity is used as a placeholder for the BPMN events.

Activity Properties

The Placeholder activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking the Activity Properties tool in the tool bar or by selecting the appropriate option from the context menu for the Placeholder activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Miscellaneous

The properties in the Miscellaneous group have been described below. You can use these properties to specify the web service details.

Action Output - This property allows the user to specify the action output(s) for the BPMN Placeholder activity.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Action Outputs property

1. To specify the action output, click button for the 'Action Output' property to open the property window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. In this window, enter the action output value in the **Output String** field.
3. After entering the details, click **Add Row** to add this action output setting and display a new row.
4. Remove rows if necessary by selecting them and clicking **Remove Selected**.
5. After setting the action output settings, click **Update**.

See Placeholder - Action Outputs for a detailed description of the Action Outputs property window.

Activity outputs:

The Intermediate Event Placeholder Activity does not return any value.

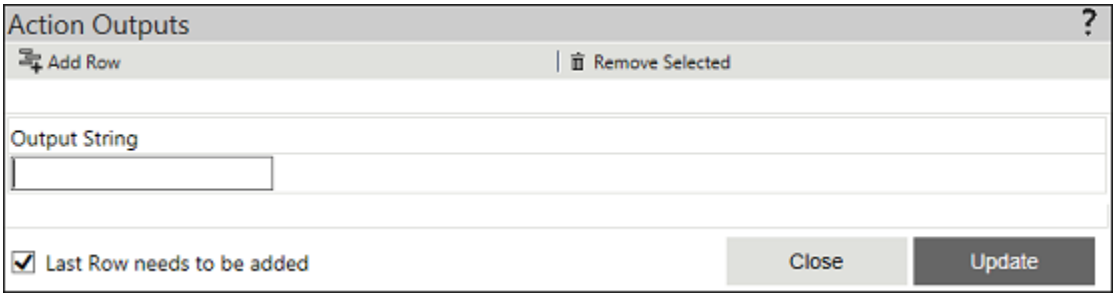
Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Placeholder: Action Outputs

You can set the Action Outputs settings in this property window. To open this window, click the button for the 'Action Outputs' property. It displays the timer settings in a grid, which also has a blank row where you can enter the settings for new Action Outputs.



The window has two tabs at the top to manage the rows of Action Outputs settings in the grid.

- **Add Row** - You can click this tab to add the newly entered settings and display a new row to the grid where you can add another setting.
- **Remove Selected** - You can click this tab to remove the Action Outputs settings selected in the grid.

The grid has the following parameters that you need to specify for the new Action Outputs setting.

- **Action Outputs** - You can enter the output message for this activity in this field.

There are two buttons and a check box at the bottom of the screen.

- **Update** - You can click this button to save the newly added Action Outputs settings.
- **Close** - You can click this to close the property window without saving the newly added Action Outputs settings.
- **Last Row needs to be added** - You need to select this check box to save the last row added.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

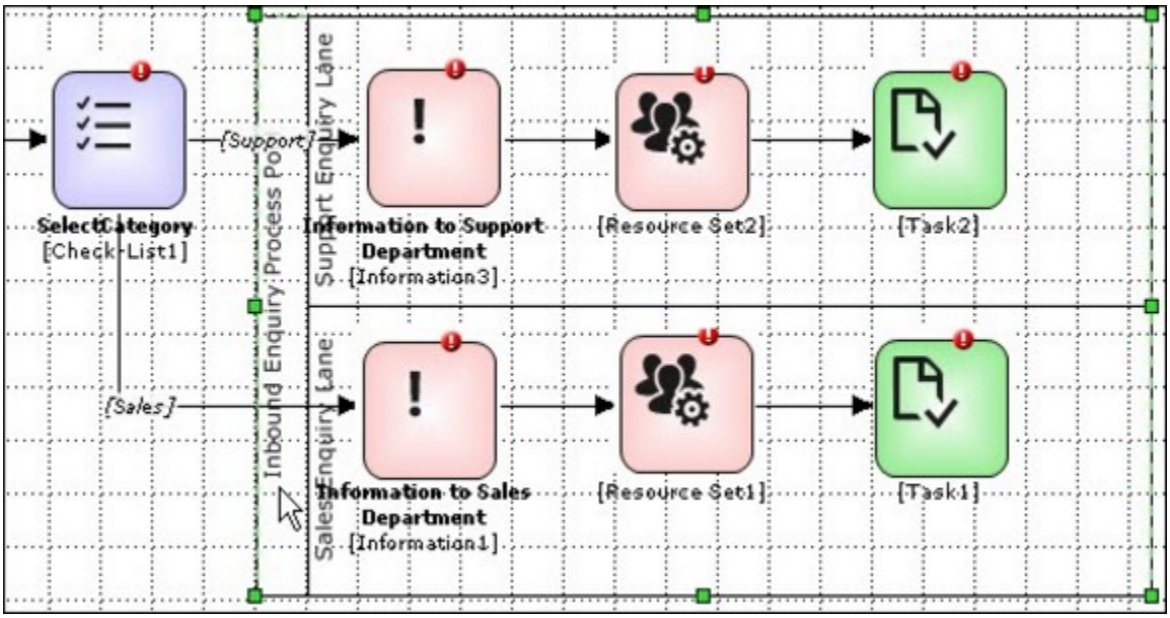
Pool

Activity description

The Pool Activity is used to create a Pool in the process model.

Application Scenarios

The Pool activity is a useful tool for process modeling. It can be placed around a set of activities in the process to denote a process execution pool. This helps to streamline the process diagram. In the image shown below, the Pool represents the enquiry handling process with two swim-lanes representing the two types of enquiries handled in the process, i.e. Support Enquiry and Sales Enquiry.



Sample Pool

Activity Properties

The Pool activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the context menu for the Pool activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Pool Activity does not return any value.

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

List Activities

There are two activities under List Activities:

- **List Navigator Activity**: The List Navigator activity is used to change the scope of the XML List XML variable during the workflow execution.
- **Wait for Event Activity**: The Wait for Event Activity is used to wait for list events and list item events like creating workflows, publishing workflows, creating users etc.

List Navigator Activity

Activity Description

The List Navigator activity is used to change the scope of the XML List XML variable during the workflow execution.

Usage Scenario

Telematics Project Managers can submit ad hoc requests for list of overdue tasks assigned to all team members. The request triggers a workflow that goes through every item in the Task List to build a report of tasks with overdue status.

Activity Properties

The List Navigator activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking the Activity Properties tool in the tool bar or by selecting the appropriate option from the right-click menu for the List Navigator activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Display Name - This property specifies the name to be displayed for the activity in the Web Work Item List.

Property Type: Optional (This property need not be set if the display name is already configured in the Actions.xml. The name entered here overrides any display name that is configured for this activity in the Actions.xml file.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

XMLVariable of List Type - This property is used to select the XML variable of List type (XML List).

Property Type: Mandatory

Steps to set the XMLVariable of List Type

Select an XML variable from the XMLVariable of List Type drop-down list.

See XML List Navigator Activity - XMLVariable of List Type for a detailed description of the XMLVariable of List Type property window

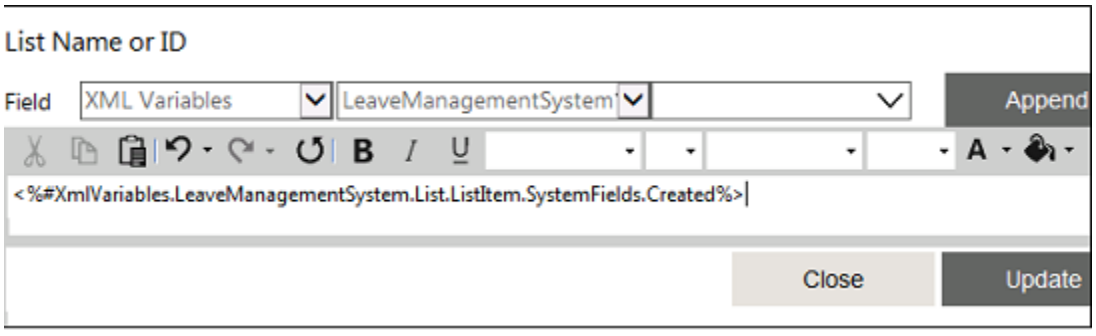
.

List Name or ID - This property is used to specify the list name.

Property Type: Optional

Steps to set the List Name

1. To specify the list name, click edit button for the 'ListName' property to open the property window.
2. In this window, enter the list name directly in the text editor.
3. If you want to customize the list name, you can do so using the **Field** controls to add workflow property values. For example, to select the XMLVariable. you can do as follows: Select "XML Variables". Select the XML variable from where you want to fetch the list name and then select respective node which is expected to have the values from the drop downs for **Field** and click **Append**. This will add the placeholder for the XMLvariable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the list name.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the list name, click **Update** to save it.
7. Click **Close** to close the window without saving the list name.
8. Click **Save** to save the XML variable in the **Activity Properties** window.

See List Navigator: List Name or ID for a detailed description of the List Name property window.

Folder Name - Enter the folder name.

Property Type: Optional (This property needs to be set only if necessary.)

Scope - This property allows you to change the scope of the XML variable during the workflow execution.

Property Type: Optional

Steps to set the Scope

1. Select the option from the **Scope** drop-down list.
2. If **Current** option is selected, you can fetch the current item which was alerted in the workflow.
3. If **New** option is selected, you can add new item to the list.
4. If **List Query** option is selected, you can build the list query to fetch the items.

List Query - This property is used to build the query for the list.

Property Type: Optional

Steps to set the List Query

1. To build the list, click the edit button for the 'List Query' property to open the property window.
2. You can also enter the query directly in the text editor.

   Note: While building query to fetch the items, you need to specify the table name with column name in a Where clause. For example, SKEUsers.Title="AVEVAWorkTasks";
3. You can also build the query using dynamic Query builder.
4. Click the query builder icon to build the query. See List Navigator Activity - Query Builder for a detailed description of the List Query builder window.
5. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
6. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
7. After adding the list name, click **Update** to save it.
8. Click **Close** to close the window without saving the query.
9. Click **Save** to save the XML variable in the **Activity Properties** window.

See List Navigator: List Query for a detailed description of the List Query property window.

Note:
- For executing the List Query following user information is considered for security:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

-- If the Workitem Context is available for the List Item, it will consider the workitemactorid as the user id.
-- If the Workitem Context user id is unavailable, then the System user configured for the Repository is considered.
-- If the System user is not configured for the Repository, then it will consider the currently logged-in user id.
- Make sure the user has administrator rights in the AVEVA

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The List Navigator Activity returns the following values -

Action completed: Triggers this output when action is completed.

List Item Not Found: This output will be given for the activity, when the list query specified does not return any items.

Note: Enable Errors Encountered Output property of **List Navigator** should be set as No to capture this output.

Exception Behavior

These options of Wonderware Skelta BPM 2009 have now been integrated with **Raise Error on Unhandled Linked Output** and **Exception Email Template Configuration** in the latest version of the product.

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Outputs for more details.

Property Type: Optional (This property needs to be set only if necessary.)

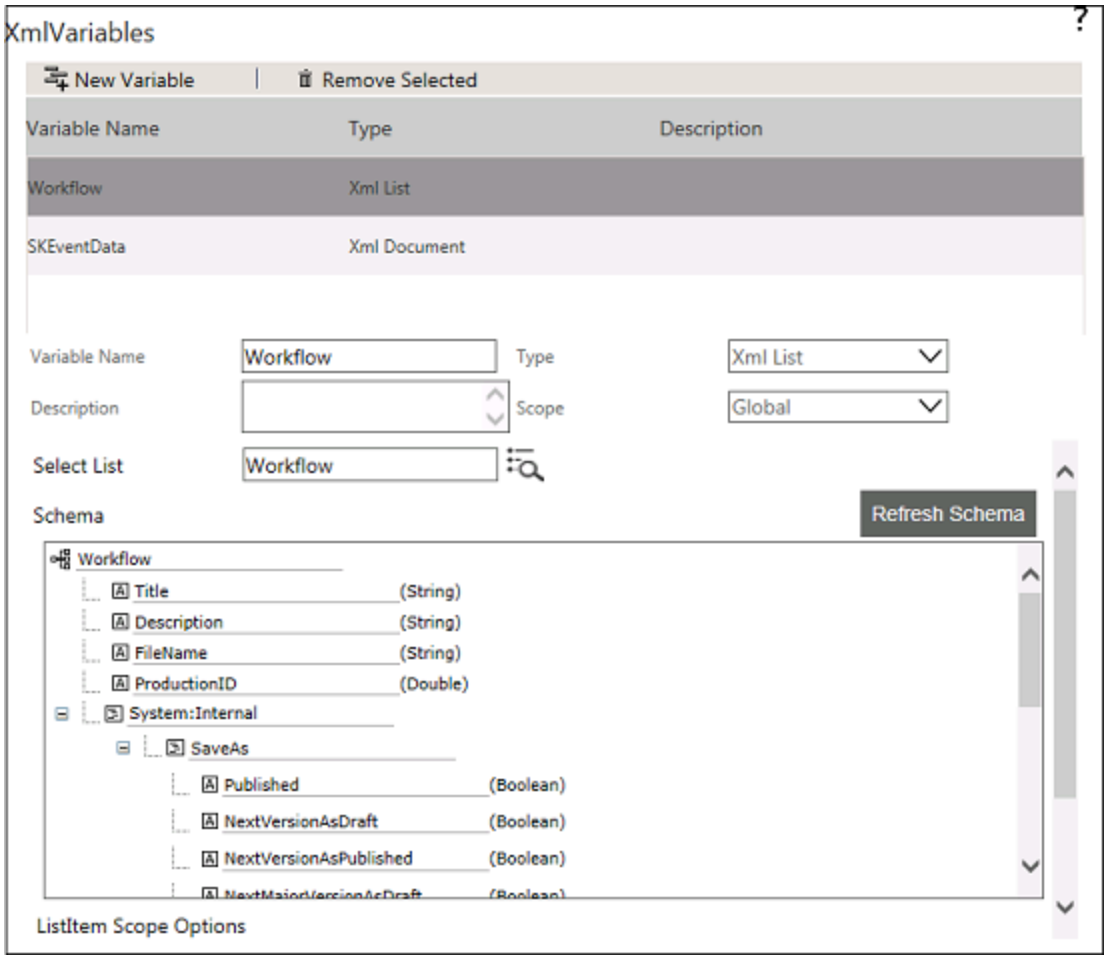
XML List Navigator Activity - XMLVariable of List Type

XML Variable of List type can be configured in the Start activity through the XML Variable property. XMLVariable of List typed can be mapped to any list schema i.e. either to lists provided out of the box or to lists created dynamically using create list option, to perform retrieval of data of a list item.

To get XMLVariable of List Type in the drop-down list, do the following steps:

1. Log on to the **Enterprise Console**.
2. Select **Workflows** from the Enterprise Console menu.
3. Click **New**, and then click **List** to design a list workflow.
4. Create the workflow.
5. In the Process Designer, open **Activity Properties** of the **Start** activity.
6. Click **xmlVariables** property to configure XML Variable of List type. The following screen appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



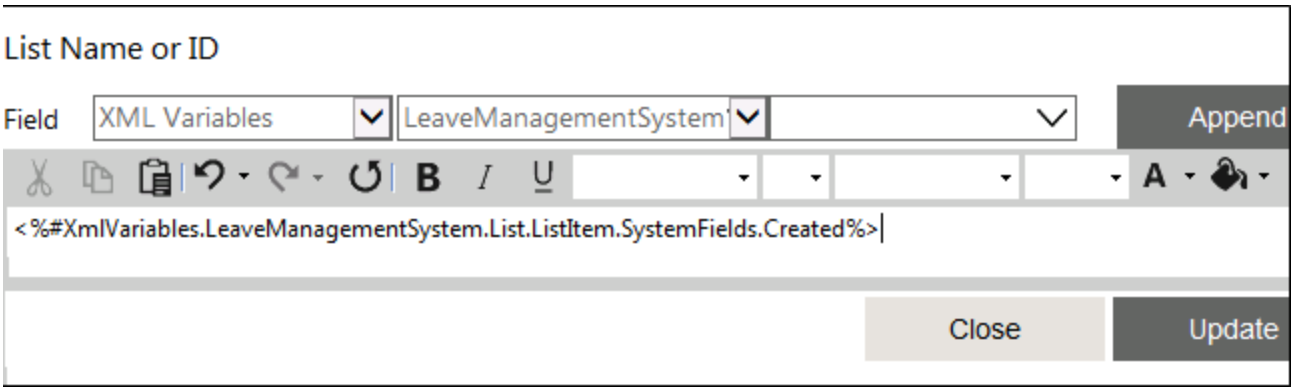
7. Click **New Variable** to create a XMLVariable of type XML List.
8. Enter variable name in the **Variable Name** textbox.
9. Select **XML List** from the Type drop-down list.
10. Enter the List name or click the icon to select the list.
11. Select **Current** option to work on current item which alerted the workflow or Select **New** option to add new item to the list.
12. Click Save Variable to save the created XML variable of List type.
13. Now Right-click on List Navigator activity and select the **Activity Properties**.
14. The above created XML variables will be listed in the XMLVariable of List Type. Now you can select the required **XML variable from the XMLVariable of List Type** drop-down list.

Note: The XML List XML variable can have work on following scopes
Current - This scope allows you to fetch the current item which was alerted in the workflow.
New - This scope allows you to add new item to the list.
List Query - This scope allows you to fetch items as per the query.

List Navigator: List Name or ID

You can specify a list name for the list in this property window. To open this window, click the edit button for the 'List Name' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the list name. You can do this either by entering a discrete value directly in the text editor or by using the Field controls at the top of the window. These controls are described below:

- The List value type drop down - You can select the type of list value here. You can specify SendTo, Variable, Content, or XMLVariables here.
- The Value text box/drop down - You can specify the actual name of the list. You can select the list name from the drop-down list.
- **Append** - You can click this button to add the specified list variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the list name or ID. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

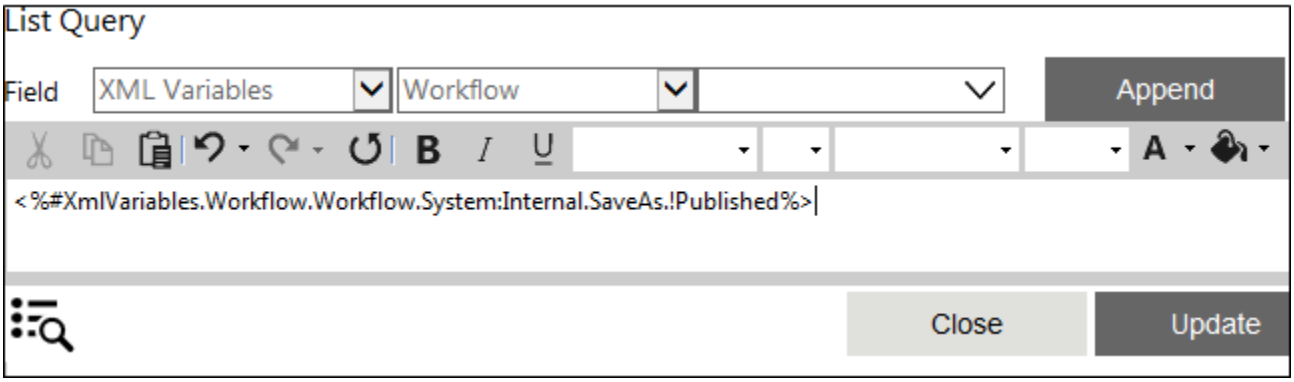
At the bottom of the window you will find the following buttons:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Update** - You can click this button to save the list name or ID
- **Close** - You can click this button to close this window

List Navigator: List Query

You can build query for the list in this property window. To open this window, click the edit button for the 'List Query' property.



The window has a text editor to build the query for the list. You can do this either by entering a discrete value directly in the text editor or by using the Field controls at the top of the window. These controls are described below:

- The List value type drop down - You can select the type of list value here. You can specify SendTo, Variable, Content, or XMLVariables here.
- The Value text box/drop down - You can specify the actual name of the list. You can select the list name from the drop-down list.
- **Append** - You can click this button to add the specified list variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the list name or ID. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

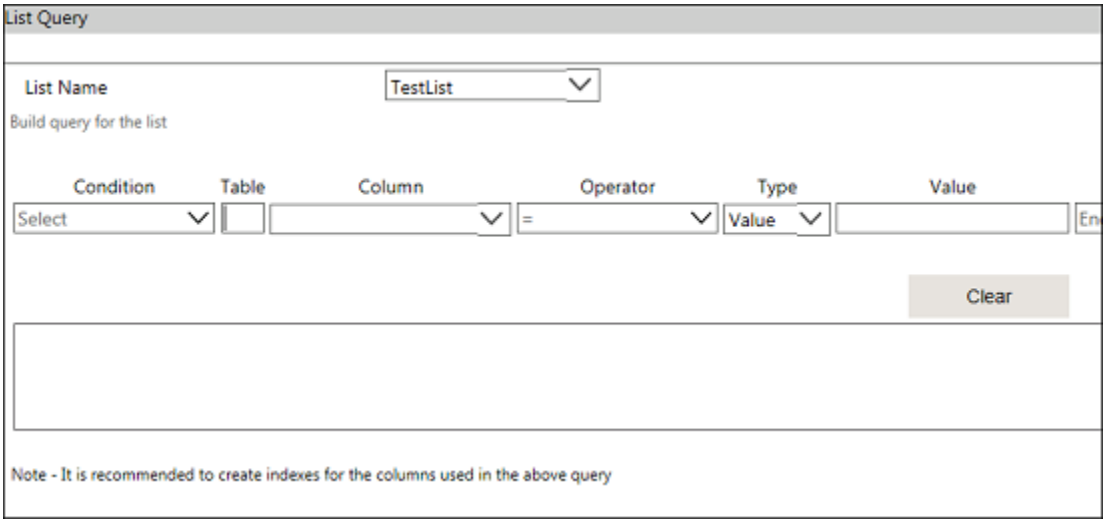
At the bottom of the window you will find the following buttons:

- **Query Builder**- You can click the query builder icon to build the query. See List Navigator: Query Builder for building query. To replace or change the existing query, delete the existing query, and then click on the **Query Builder** icon to enter the new query.
- **Update** - You can click this button to save the list name or ID.
- **Close** - You can click this button to close this window.

List Navigator: Query Builder

You can build query in this window.

To open this window, click the icon for the 'List Query' property.



The following are the field values for List Query window:

- **List Name** - Select the list from List Name drop-down list.
- **Table** - By clicking on this textbox, a pop up allows you to select the corresponding table for query. The respective table name will display in the pop up window based on the list selected by the user.
- **Column** - You can select this drop-down list to select the column or field of the corresponding list.
- **Operator** - You can select this drop-down list to select an operator (either one of =,>, <,>=, <=, <>, IN, Not IN, Like) to build the query.
- **Type** - You can select this drop-down list to select a type. The value is automatically displayed in the textbox.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

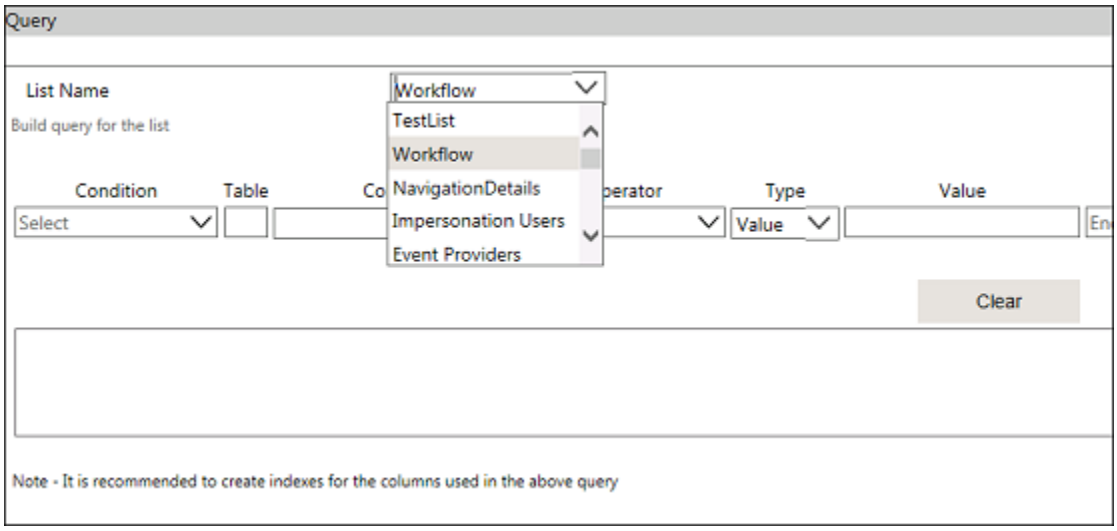
- **Condition** - You can select this drop-down list to select the condition [either of End, And, Or, '),' '('].
- **Add** - You can click this button to add a query in the field.
- **Clear** - You can click this button to clear the query contents.
- **OK** - You can click this button to save this query.
- **Close** - You can click this button to close this window.

Example:

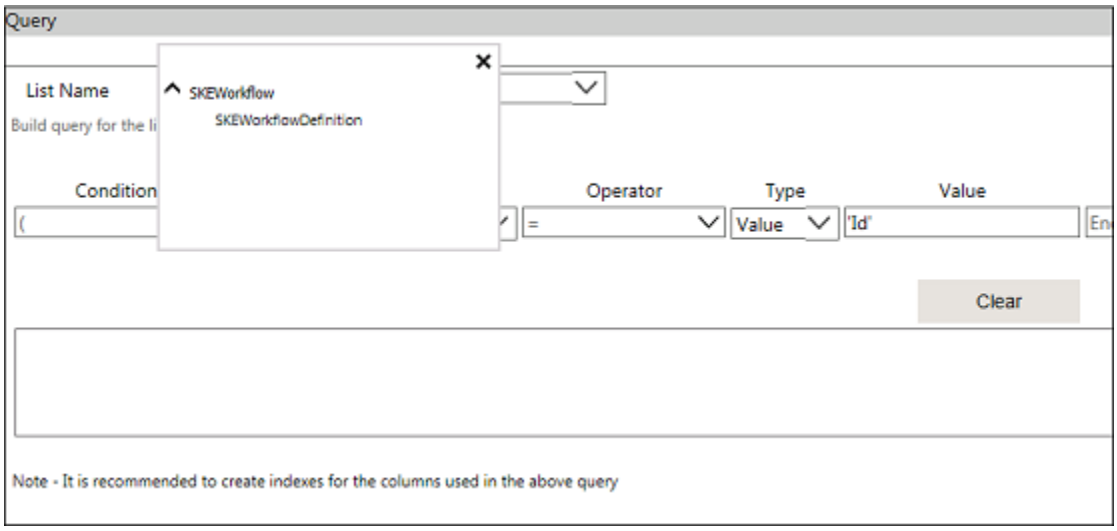
Retrieve the records whose title is starts from 'C' and 'B' in Product Categories list.

The following steps to be performed:

1. Click the icon to build the query for the list 'Product Categories'.
2. Select a list name from the **List Name** drop-down list.

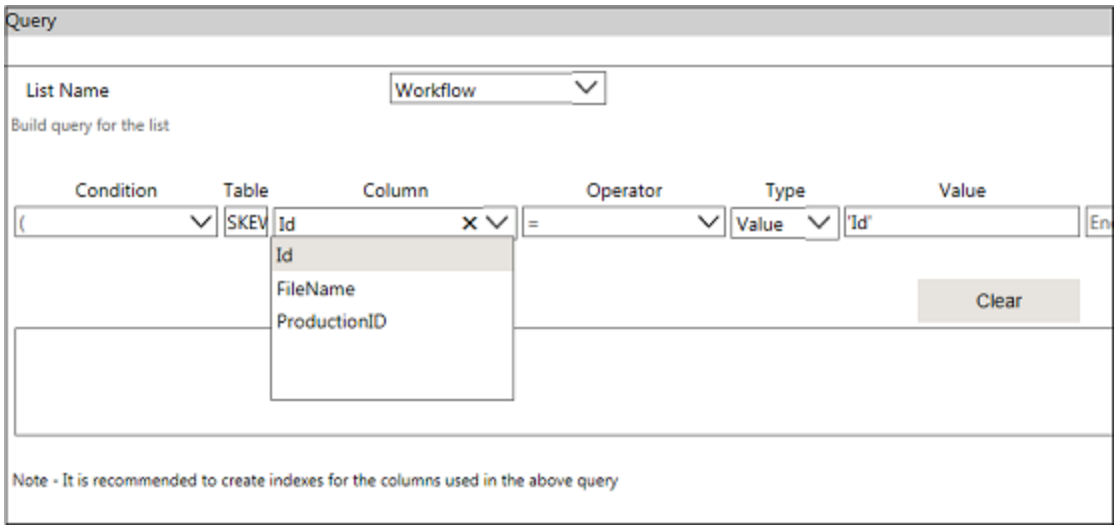


3. Click on **Table**, the pop up window allows to select the table from the available list.

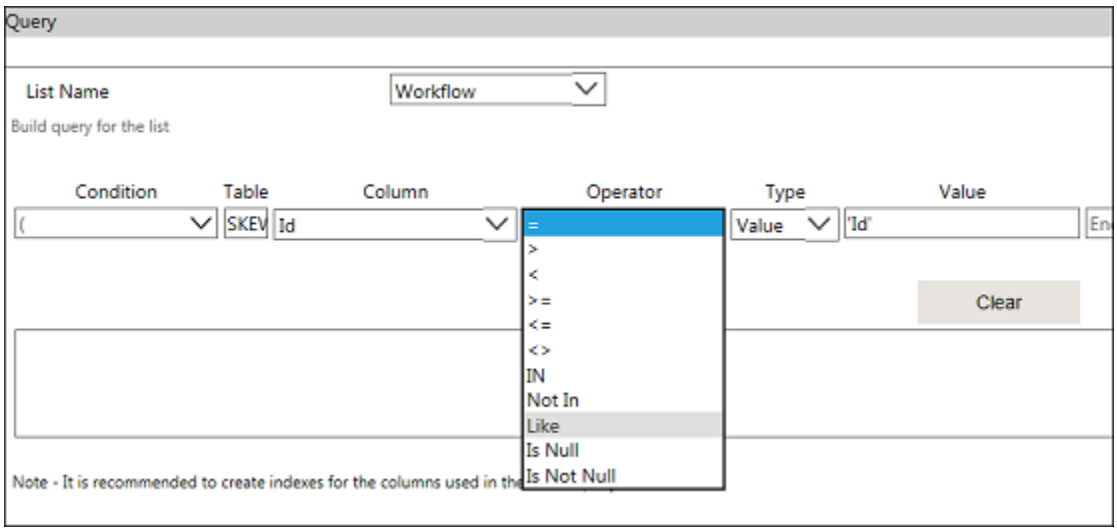


4. Select the table name. The table name appears in the textbox.
5. Select the column name from the **Column** drop-down list.

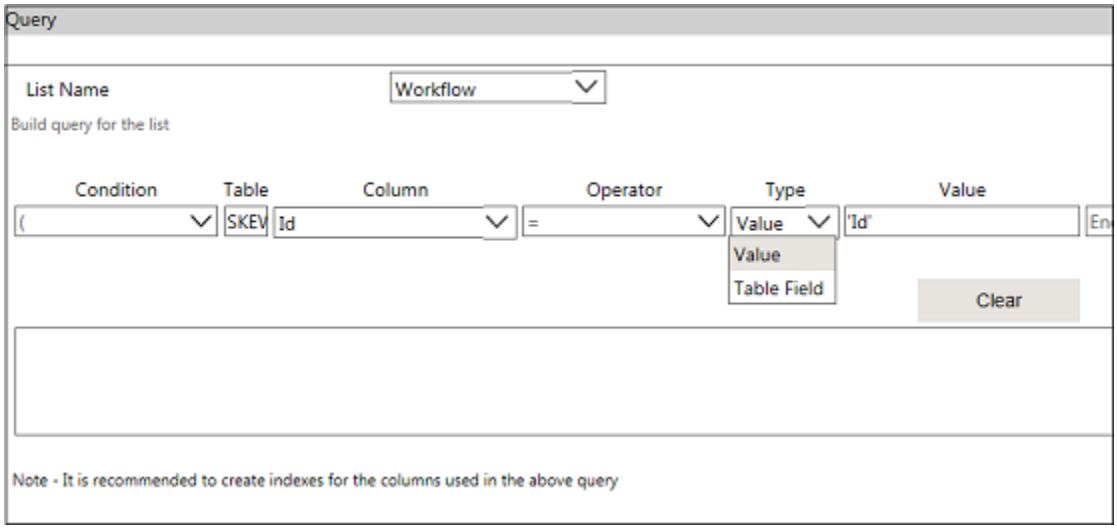
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



6. Select an operator from **Operator** drop-down list.



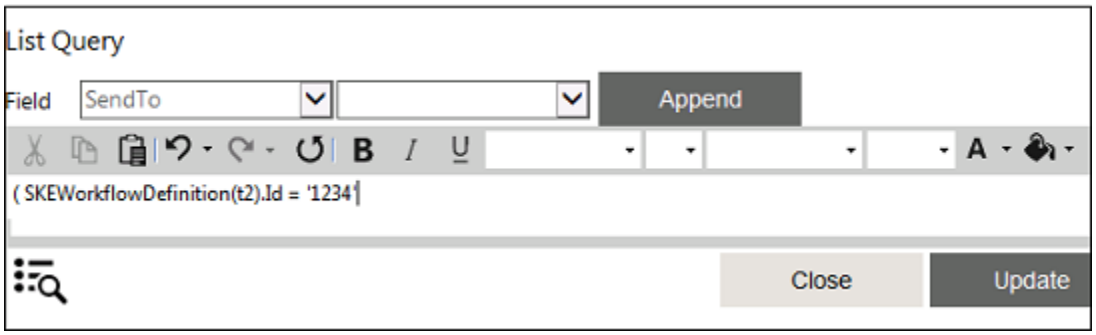
7. Select a type from the Type drop-down list.



8. Type a value in the **Value** textbox.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Select a condition operator from the **Condition** drop-down list.
10. Click **Add** to add the condition in textfield.
11. Repeat the steps if required.
12. Click **OK** to save the query. The build query appears in the List Query window.



13. You can replace the value for the query, with variables. To replace, select the text/value to be replaced and then select the respective field.
14. Click **Update**.
15. Click **Save** in the **Activity Properties** window of the List Navigator activity.

Note: If the query returns more than one item of list for the XML variable, then you must have XML variable of type XML List Iterator for the list to loop through.

Wait for Event Activity

Activity Description

The Wait for Event Activity is used to wait for list events and list item events like creating workflows, publishing workflows, creating users etc. This activity waits for one or more event and once the event is triggered, the workflow proceeds. When the activity is waiting for an event it can be paused,resumed or killed through process execution view.

Activity Properties

The Wait for Event activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking the Activity Properties tool in the tool bar or by selecting the appropriate option from the right-click menu for the Wait for Event activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if necessary.)

Display Name - This property specifies the name to be displayed for the activity.

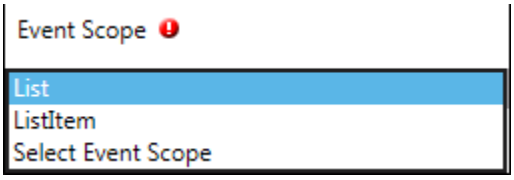
Property Type: Optional (This property need not be set if the display name is already configured in the Actions.xml. The name entered here overrides any display name that is configured for this activity in the Actions.xml file.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Event Scope - This property is used to select the scope to which event association is required. Available options are - List and ListItem.

Property Type: Mandatory



Event Scope Explanation:

1. List: Select this scope option only when List Id is required.
2. List Item: Select this scope option when List Id and Listitem Id is required.

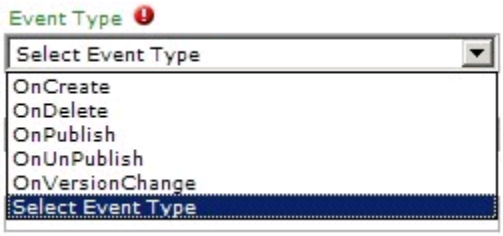
Steps to select the Event Scope property

1. Select scope from the Event Scope drop-down list.
2. If you select List as event scope, then you need to specify only list name.
3. If you select ListItem as event scope, then you need to specify list name and list item under the list.
4. Click Save in the Activity Properties window.

Event Type - This property is used to select the event type. Items should change depending on Event Scope property selection.

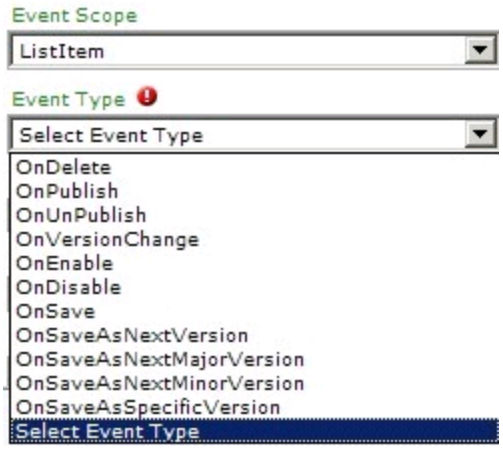
Property Type: Mandatory

If you select Event Scope property as List, then the following are the available event types:



If you select Event Scope property as ListItem, then the following are the available event types:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The options shown for Event type are dependent on Event Scope property selection.

Note:

The following Event types are not supported:

- OnSaveAsNextVersion
- OnSaveAsNextMajorVersion
- OnSaveAsNextMinorVersion
- OnSaveAsSpecificVersion

Steps for selecting Event Type property

1. Select event type from the Event Type drop-down list.
2. Click Save in the Activity Properties window.

List ID - This property is for taking list name or list item to which the event to be associated and waited. List ID property window displays the items depending on the Event Scope property selection. If you select Event Scope property as 'List', then the List ID property window will ask only List name to select. If you select Event Scope property as 'ListItem', then the List ID property window will ask List name as well as ListItem name.

Property Type: Mandatory

Note: The options shown for List ID are dependent on Event Scope property selection.

Steps to set the List ID property

1. To select the List ID property, click the edit button for the 'List ID' property to open the property window.
2. If you select List as event scope, then select list name from the List Name drop-down list.
3. If you select ListItem as event scope, select list name from the List Name drop-down list and select list item from the ListItem drop-down list.
4. Click Save in the Activity Properties window.

See Wait for Event: List ID for a detailed description of the List ID property window.

List Item ID - This property displays the list item id which was selected in List ID property.

Note: List Item ID are dependent on Event Scope property selection.

No of Events - This property is used to execute the wait for event activity by specified number of events. If you

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

specified No Of Events as '0' then this activity will wait for 'n' number of times. If you specified No Of Events as '4' then this activity will wait for four times and next time it will not wait for the event.

Property Type: Optional

Steps for specifying No Of Events property

To select the No Of Events property, click the edit button for the 'No Of Events' property to open the property window.

1. You can directly type in the text editor.
2. Or you can specify the No of Events by using field controls and then click Append.
3. Click Update.
4. Click Save in the Activity Properties window.

See Wait for Event for a detailed description of the No Of Events property window.

Effective From - This property is used to specify the effective from date for the association, i.e the date starting from which the activity will wait for the events. If left blank, the activity will wait for events starting from the day the association is created.

Property Type: Optional

Steps to set Effective From property

1. To select the Effective From property, click the edit button for the Effective Form property to open the property window.
2. You can directly type in the text editor, or you can specify the date by using field controls and then click Append.
3. Click Update.
4. Click Save in the Activity Properties window.

See Wait for Event: Effective From From for a detailed description of the Effective From window.

Effective To - This property is used to set the Effective To Date. Enter the effective to date for the association, i.e the date till which the activity will wait for the events. If left blank, it will be wait till the activity is paused/killed or the workflow is aborted.

Property Type: Optional

Steps to set Effective To property

1. To select the Effective To property, click the edit button for the Effective To property to open the property window.
2. You can directly type in the text editor, or you can specify the date by using field controls and then click Append.
3. Click Update.
4. Click Save in the Activity Properties window.

See Wait for Event: Effective To for a detailed description of the Effective To window.

Output Values

Event XML Data Variable. - This property is used to capture the Output details of the wait for event activity.

Property Type: Optional (This property needs to be set only if necessary).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

See Wait for Event: Workflow Event Data XML Variable for details on Event XML Data Variable and the output details.

Looping Behavior

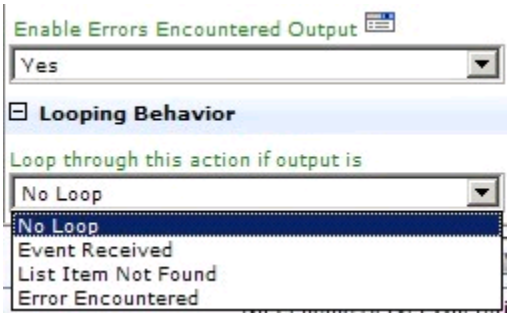
The properties in the Looping Behavior group have been described below. You can use these properties to specify the looping Behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down. Available options are - Event Received and List Item Not Found.

Property Type: Optional

If you select Enable Errors Encountered Output property as No, then the following are the available outputs:

If you select Enable Errors Encountered Output property as Yes, then the following are the available outputs:



Note: The options shown for Looping through this action if output is are dependent on Enable Errors Encountered Output property selection.

Activity outputs:

The Wait For Event Activity returns the following values -

Event Received: Displayed when the event is received.

List Item Not Found: Displayed when the list item is not found.

Error Encountered: Displayed when an error encountered.

Action Completed: Displayed when no event is received.

Exception Behavior

These options of Wonderware Skelta BPM 2009 have now been integrated with Raise Error on Unhandled Linked Output and Exception Email Template Configuration in the latest version of the product.

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

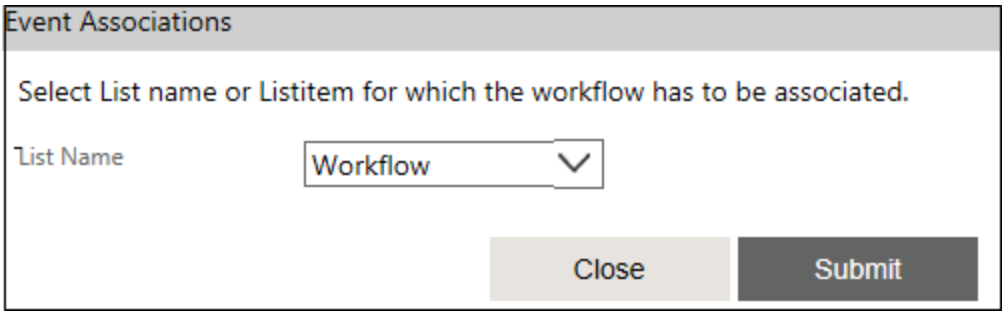
See Mapped Error Outputs for more details.

Property Type: Optional (This property needs to be set only if necessary.)

Wait for Event: List ID

You can select the list name and list item for which the workflow has to be associated in this property window. To open this window, click the edit button for the 'List ID' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has following fields:

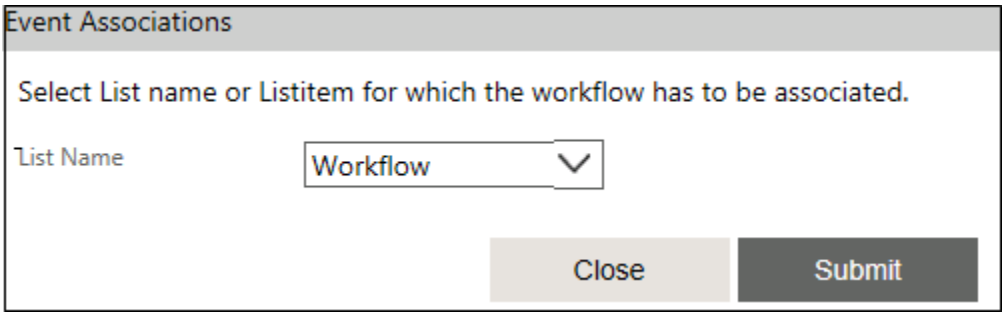
List Name - This allows you to select list name from the drop-down list.

Submit - Clicking this will save the List ID and List Item ID in the respective properties.

Close - Clicking this will close the window.

To set the List ID or List Item ID:

1. Select event scope as 'List', if you want only List Id.
2. Click **Save** in the **Activity Properties** window.
3. Click the edit button to open the List ID property window. The following screen is displayed.



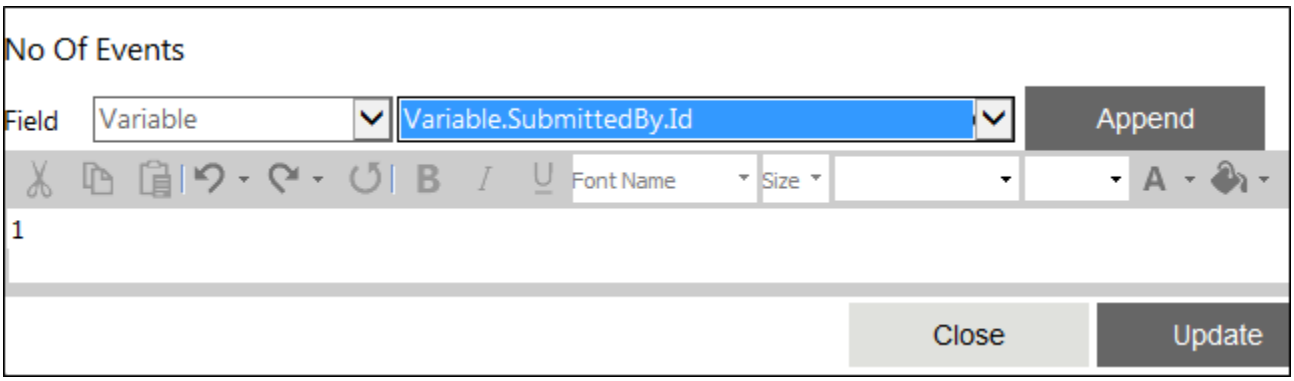
4. Select list name from the drop-down list.
5. Click **Submit** to display list id of the selected list in the List ID property.
6. The List Id in the **List ID** fields respectively.
7. Select event scope as 'ListItem', if you want both List Id and List Item Id.
8. Click **Save** in the **Activity Properties** window.
9. Click the edit button to open the List ID property window.
10. Select list name from the drop-down list. For example, select Users from the List Name.
11. After selecting list name, List Item field will be displayed.
12. Select List item from the **List item** drop-down list. For example, select 'George' user from the available users.
13. Click **Submit**.
14. The List Id and List Item Id will be displayed in the **List ID** and **List Item ID** fields respectively.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Wait for Event: No of Events

You can specify number of events for which the workflow has to be associated in this property window. To open this window, click the edit button for the 'No of Events' property.

This property is used to execute the wait for event activity by specified number of events. If you specified **No Of Events** as '0' then this activity will wait for 'n' number of times. If you specified **No Of Events** as '4' then this activity will wait for four times and next time it will not wait for the event.



The window has a text editor to enter the number of events. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The value type drop down - You can select the type of value here. You can specify SendTo, Variable, Content, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Custom Document View URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Custom Document View URL.
- **Append** - You can click this button to add the specified variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the number of events specified.
- **Close** - You can click this button to close this window.

Wait for Event: Effective From

You can specify effective from date for the association in this property window. If left blank, the date on which the association was created will be considered as the start date to wait for an event. To open this window, click the edit button for the 'Effective From' property.

Note: You can also specify the time in the text editor. The time entered can be in either 24 hour or 12 hour format. If time is entered in 12 hour format then AM/PM has to be specified.

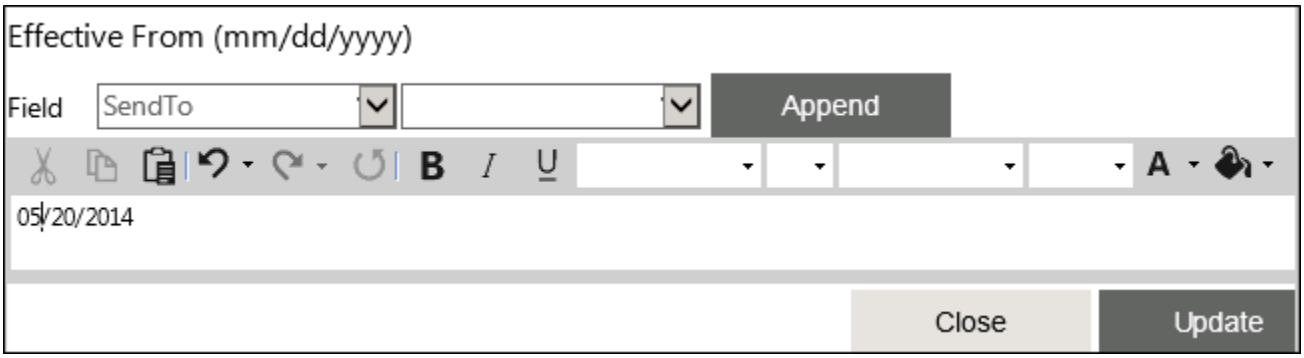
Example:

Date and time in 24 hour format

08/18/2010 22:30:00

Date and time in 12 hour format

08/18/2010 10:30:00 PM



The window has a text editor to enter the date. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The value type drop down - You can select the type of value here. You can specify SendTo, Variable, Content, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Custom Document View URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Custom Document View URL.

- **Append** - You can click on this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window, you will find the following buttons:

- **Update** - You can click this button to save the date specified.
- **Close** - You can click this button to close this window.

Wait for Event: Effective To

You can specify effective to date for the association in this property window. If left blank, will be valid always. To open this window, click the edit button for the 'Effective To' property.

Functionality

The activity will wait for events starting from the Date and Time specified in the Effective From field till the Date and Time specified in this field.

- If only date is specified in the Effective To field and time is not specified, then the activity will wait till the end of the day specified in Effective To field.
- If both date and time is not specified in the Effective To field, then the activity will continue to wait for an

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

event always, starting from the the Date and Time specified in the Effective From field.

Illustration 1

- Effective From field is 08/10/2010 00:00:00
- Effective To field is 08/18/2010 22:00:00

The activity will wait for events starting at 00:00:00 hours on 10th August 2010 till 22 hours (10:00 pm) on 18th August 2010.

Illustration 2

- Effective From field is 08/10/2010 00:00:00
- Effective To field is 08/18/2010

The activity will wait for events starting at 00:00:00 hours on 10th August 2010 till 23:59 hours (11:59 pm) on 18th August 2010.

Illustration 3

- Effective From field is 08/10/2010 00:00:00
- Effective To field is blank

The activity will continue to wait for events starting at 00:00:00 hours on 10th August 2010 till the workflow is aborted or activity is paused or killed.

Note: You can also specify the time in the text editor. The time entered can be in either 24 hour or 12 hour format. If time is entered in 12 hour format then AM/PM has to be specified.

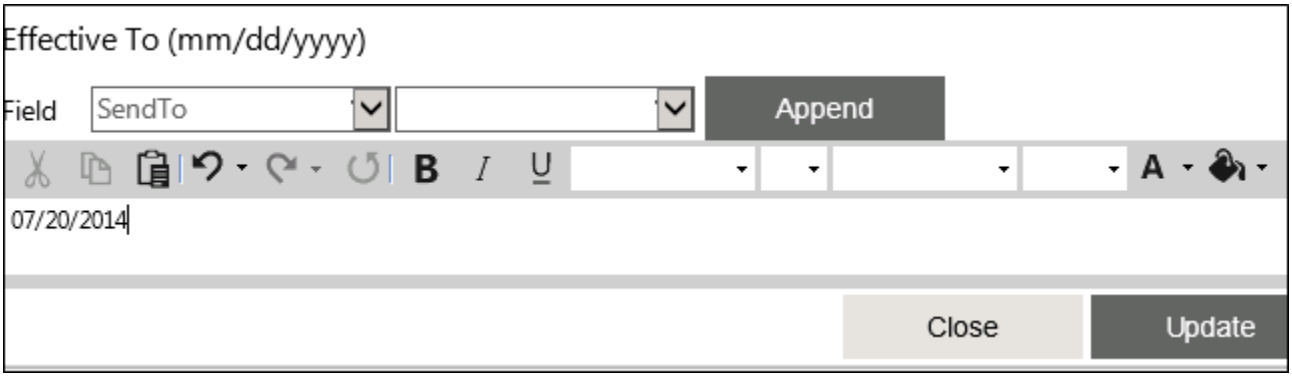
Example:

Date and time in 24 hour format

08/18/2010 22:30:00

Date and time in 12 hour format

08/18/2010 10:30:00 PM



The window has a text editor to enter the date. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The value type drop down - You can select the type of value here. You can specify SendTo, Variable, Content, or XMLVariables here.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The Value text box/drop down - You can specify the actual value of the Redirect URL. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Custom Document View URL in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Custom Document View URL.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the subject
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window, you will find the following buttons:

- **Update** - You can click this button to save the date specified.
- **Close** - You can click this button to close this window.

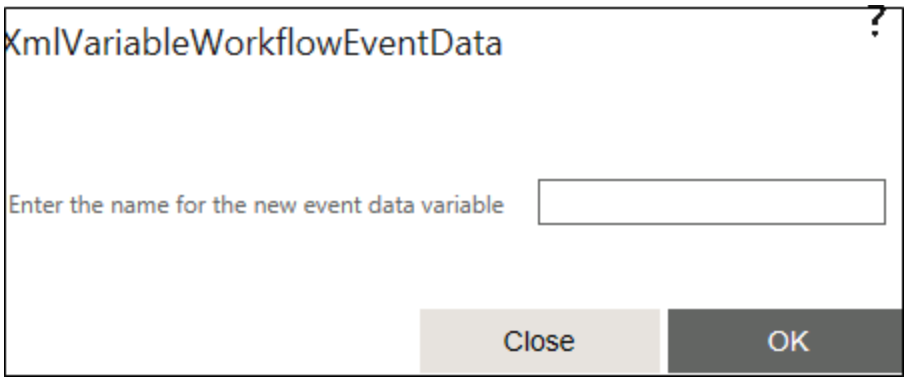
Wait for Event: Workflow Event Data XML Variable

You can create a new event variable in this property window. To open this window, click the edit button for the 'Event XML Data Variable' property.

To create event data XML variable:

1. To open this window, click create a new variable for event data edit button.
2. Enter the variable name in the text box.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click **OK**.
4. Click **Save** in the **Activity Properties** window.

Security Activities

There are two activities under Security Activities:

- **Cancel Approval Workflow**: The Cancel Approval Workflow activity is responsible for the canceling of an operation, and the terminating of the workflow instance.
- **Security Instruction Handler**: The Security Instruction Handler activity is responsible for the completion of an operation.

Cancel Approval Workflow

Activity Description

Cancel Approval Workflow activity is responsible for the canceling an operation (Save As, Delete, Disable, Enable, Publish) and terminating the workflow instance, if while setting the rights for a specific security group, user has selected the option as **"Requires Approval**" or "**Requires Approval with Prompt**" for that operation.

If the Request for approval is rejected or is terminated due to some reason, **Cancel Approval Workflow activity** is responsible for rolling back the operation.

Scenario

Consider a scenario, where while setting the security rights for Administrator, **Can Publish Item** right has been set as **Requires Approval** with the selected workflow as **Security Approval Workflow**.

Security Approval workflow ensures that Publish operation completes successfully only after approval from the concerned person. But if the request is rejected by the approver, then Security Approval workflow uses Cancel Approval Workflow activity internally to terminate the request.

For more details, see *Security Approval Workflow*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

The Cancel Approval Workflow activity has to be configured by specifying appropriate values for the different properties in the Action Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Approval activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

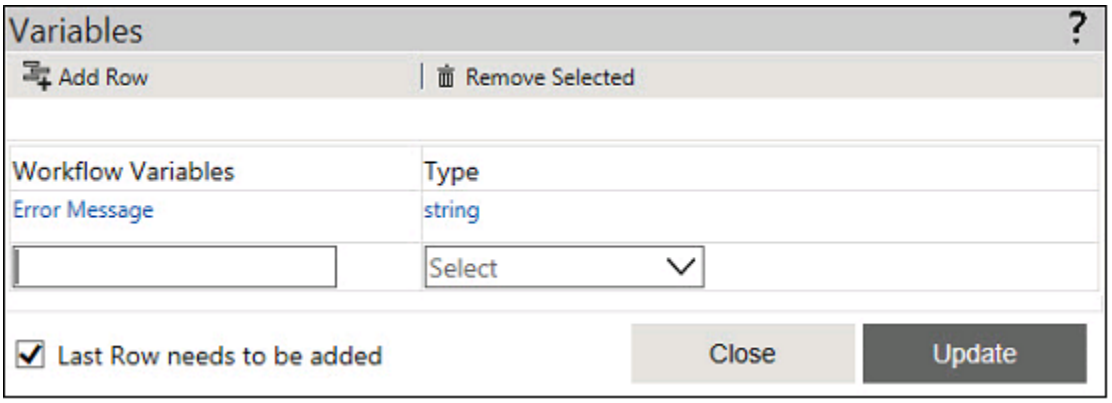
Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Note: The user needs to create a variable with the name **"Error Message"** in the Start activity of the workflow which is using the Cancel Approval Workflow activity. This variable is used to capture any errors which have might occur during the process of cancelation of the various operations (Save As, Delete, Disable, Enable, Publish).



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Security Instruction Handler

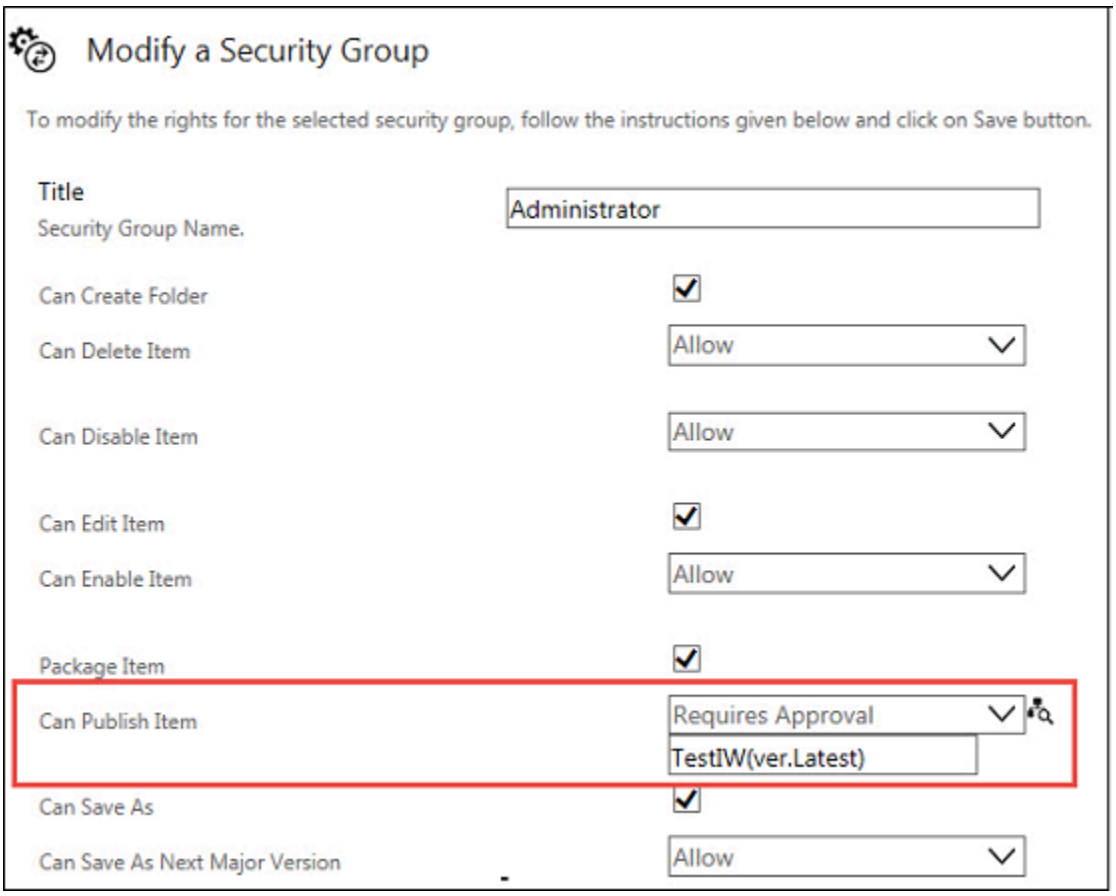
Activity Description

Security Instruction Handler activity is responsible for the completion of an operation (Say, Delete, Disable, Enable, Publish), if while setting the rights for a specific security group, user has selected the option as "Requires Approval" or "Requires Approval with Prompt" for that operation.

Scenario

Consider a scenario, where while setting the security rights for Administrator, **Can Publish Item** right has been set as **Requires Approval**, with the selected workflow as **Security Approval Workflow**.

Security Approval workflow ensures that the Publish operation completes successfully only after approval from the concerned person. Security Approval workflow internally uses Security Instruction Handler Activity to complete the Publish operation after the approval has been given.



Activity Properties

The Security Instruction Handler activity has to be configured by specifying appropriate values for the different properties in the Action Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Approval activity. The Activity Properties are organized under the following groups of related properties. The properties

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Miscellaneous

Error Description

This property is not being used currently. It is being reserved for future use.

Property Type: Optional (This property needs to be set only if necessary.)

Exception Behavior

The property in the Exception Behavior group has been described below. You can use these properties to raise exceptions if the output is not linked to the next activity in the workflow and also define and configure email templates and users for the exception mails.

Raise Error on Unhandled Linked Output? – This property can be configured to raise an error if the output is not linked to the next activity in the workflow.

Property type: Optional (This property needs to be set only if necessary)

See Mapped Error Output for more details.

Looping Behavior

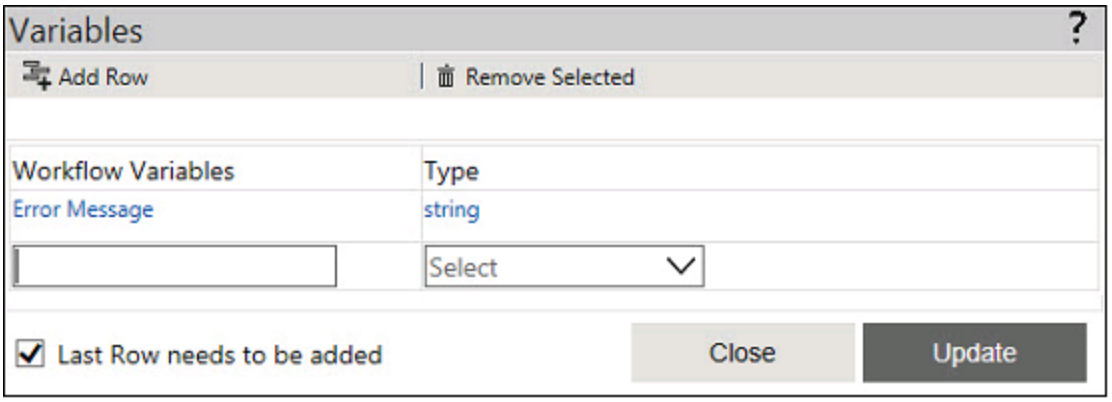
The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down.

Property Type: Optional (This property needs to be set only if necessary.)

Note: The user needs to add a variable with the name "Error Message" in the Start activity of the workflow which is using the Security Instruction Handler activity. This variable is used to capture any errors which have might occur during the process of completion of the various operations (Say, Delete, Disable, Enable, Publish).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Report Activities

This section provides information on the Report Generator Activity.

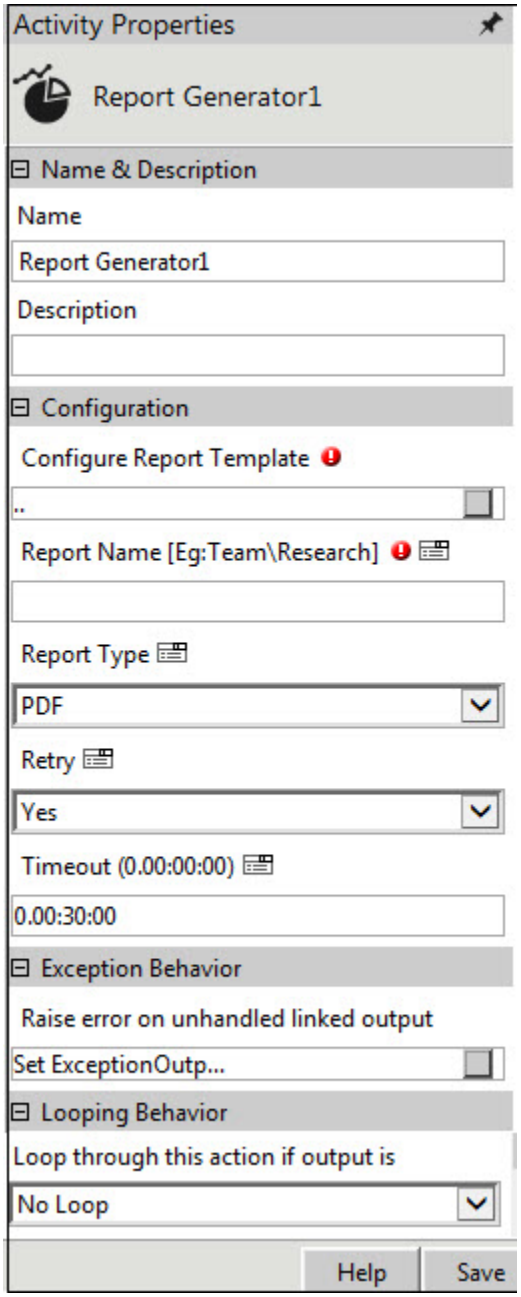
Report Generator Activity

Activity Description

You use the Report Generator activity to run a report template and save the report in the Reports list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties



Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property is to be used to specify a name for the activity.

Property Type: Mandatory (This property must be set.)

Description - This property can be used to specify a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration

The properties in this group have been described below. You can use these properties to configure the functionality of this activity.

Configure Report Template - This property is to be used to configure the Report Template from the list of available report templates.

Property Type: Mandatory (This property must be set for the configuration.)

Steps to set the Configure Report Template Property

1. To specify the **Configure Report Template** properties, click the edit button of the property and the property window opens.
2. Select the required report template from the **Report Template** drop-down field. All the parameters related to the selected report template will be listed in the Parameters field.
3. Select the required **Parameters**.
4. Select the "Generate Report Reference Variable automatically" option to populate the variable in the Report Reference Variable field.
5. Alternatively, select the "Create Report Reference Variable manually" option to create the variable manually.
6. Click **Submit**.

See Configure Report Template for a more detailed understanding of its properties.

Report Name - This property is used to specify a name for the report to be generated.

User can also pass the folder path in this field. For example, Team\Research\GroupPeriodReport. Here the root level folder name is Team under which the *Research* folder will be the destination for saving the report "GroupPeriodReport". If the given path is not present, then the Report Generator activity will create the required folders.

Property Type: Mandatory (This property must be set for the report to display a name.)

The default name for the selected Report Template is:

[Selected Report Template Name]_#workflow#_#year#_#month#_#day#_#hour#_#minutes#

The default name is a combination of the name of the selected Report Template Name and the place holders. The following table shows the place holders supported for the report name:

Place holders	Format	Description
#day#	dd	It will replace with the current day.
#month#	MM	It will replace with the current month.
#year#	yyyy	It will replace with the current year.
#hour#	HH	It will replace with the current hour.
#minutes#	mm	It will replace with the current minutes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Place holders	Format	Description
#seconds#	ss	It will replace with the current seconds.
#application#		It will replace with the repository name.
#workflow#		It will replace with the workflow name.
#activity#		It will replace with the current activity name.
#executionid#		It will replace with the workflow execution ID.
#executiondetailsid#		It will replace with the execution details ID.

Report Type - This property is used to select the type of the report to be exported from the reporting server.

Property Type: Optional

Default type is PDF. Other supported types are MS Excel and Image.

Retry - This property is used to specify if the activity must retry to connect to the reporting service.

Property Type: Optional

Yes: This is the default value. If you select Yes, the activity retries three times to connect to the SQL Server Reporting Service, if the web service does not respond at the first attempt.

No: If you select No, the activity does not retry to connect to the web service.

Timeout - This property is used to specify the web service (SQL Server Reporting Service) time out. By default, the timeout is 30 minutes. Use this property to modify the timeout.

Property Type: Optional

Exception Behavior

Each activity has the Raise Error on Unhandled Linked Output property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Looping Behavior

The property in the **Looping Behavior** group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop-down field.

Property Type: Optional (This property needs to be set only if necessary.)

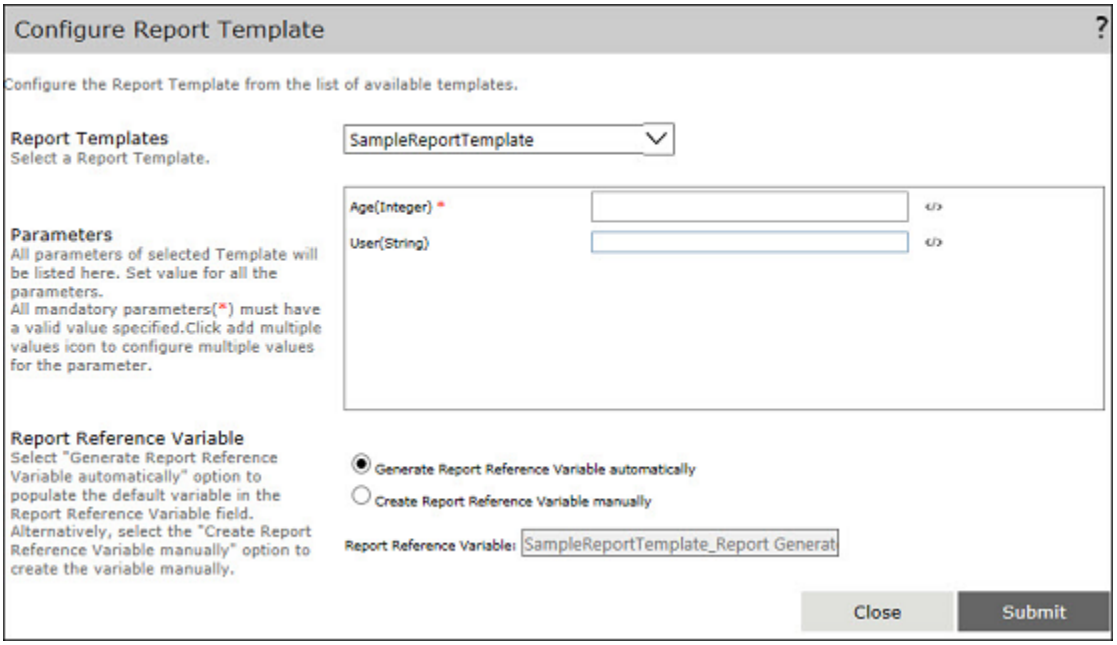
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure Report Template

You can use the **Configure Report Template** properties to perform the following tasks:

- Select the required Report Template from the list of available report templates.
- Set the value for the parameters for the selected report template.
- Generate the report reference variable or create the report reference variable manually.

To open the **Configure Report Template** window, click the **Configure Report Template** edit button.



The **Configure Report Template** window has the following properties:

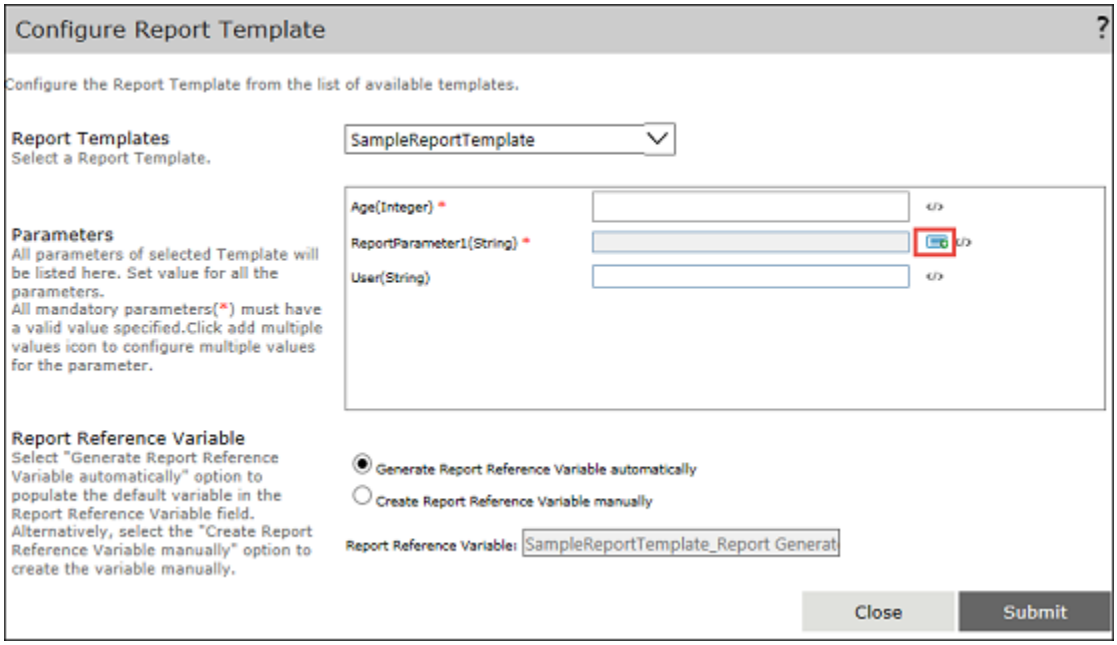
- **Report Template:** You can use this property to select the required report template from the list of available report templates. The available report templates are populated from the Report Template list.
- **Parameters:** You can use this property to set the values for the selected report template parameters. On selecting the required Report Template, the parameters defined for this particular report template appears. The user can type the value of the parameter in the respective parameter fields or use the expression editor to define the same.
- **Report Reference Variable:** You can use this property to generate a report reference variable. This default report reference variable is [Report_Template Name]_[activity Name]. You can automatically or manually generate this variable.
  This variable will be used while attaching the report to a mail channel in various Human activities within the same workflow.
- **Generate Report Reference Variable automatically:** When you select this option, you can generate the default variable in the Report Reference Variable field. You cannot edit the variable in the Report Reference Variable field.
- **Create Report Reference Variable manually:** When you select this option, you can change the name of the variable that is populated in the Report Reference Variable field. The default variable populated is same in both the options.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Multiple Value Parameter Configuration

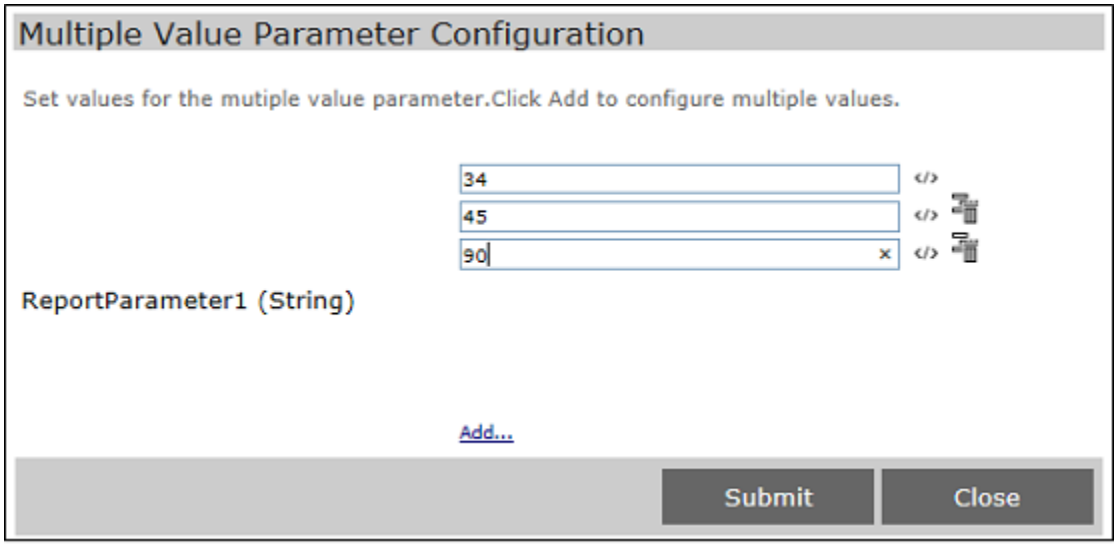
You can configure multiple values for the parameters. To configure multiple values for a parameter, complete the following steps:

1. In the **Configure Report Template** dialog box, select the template from the list of templates available in the **Report Templates** field.
2. Click the multi-value icon as shown in the following image:



3. The **Multiple Value Parameter Configuration** dialog box appears. The dialog box displays the parameter name and the data type of the parameter.

   For example, the following image shows the parameter with the name ReportParameter1 of the type Integer:

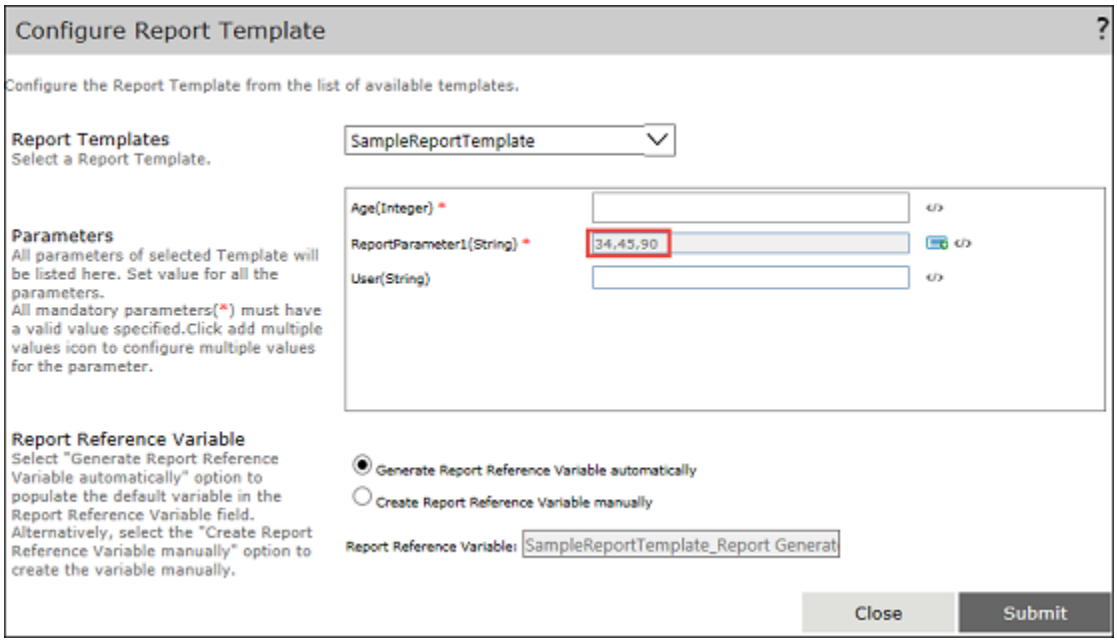


4. You can add multiple values to the parameter. To add multiple values, click Add... and then enter the values in each new row.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Click **Submit**.

   You can now see that the multiple values that you have entered appears as comma separated list in the *ReportParameter1TestMultiValue* field of the **Configure Report Template** dialog box as shown in the following image:

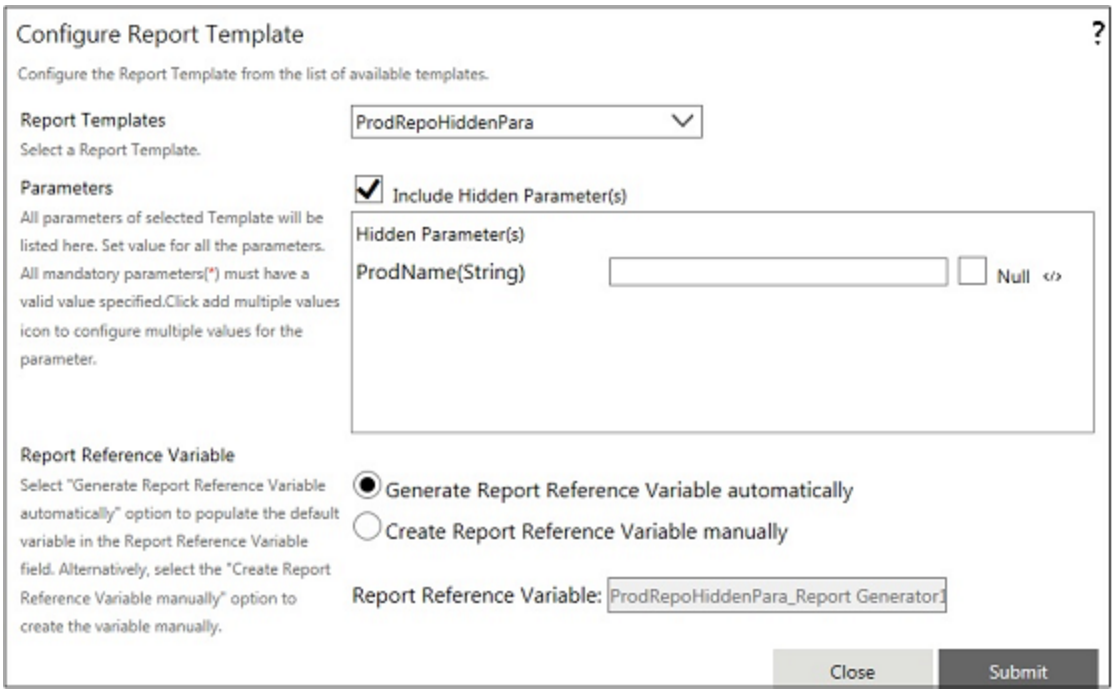


Note: To assign values to a date-time parameter, use a variable and assign a value to the variable, such as YYYY-MM-DDTHH:MM:SS.0000000. You cannot directly assign a hard coded date-time value from the expression editor.

Hidden Parameters Configuration

By default, you can see only the unhidden parameters in this section. You can check the I**nclude Hidden Parameters** option to view all the hidden parameters, as shown in the following image:

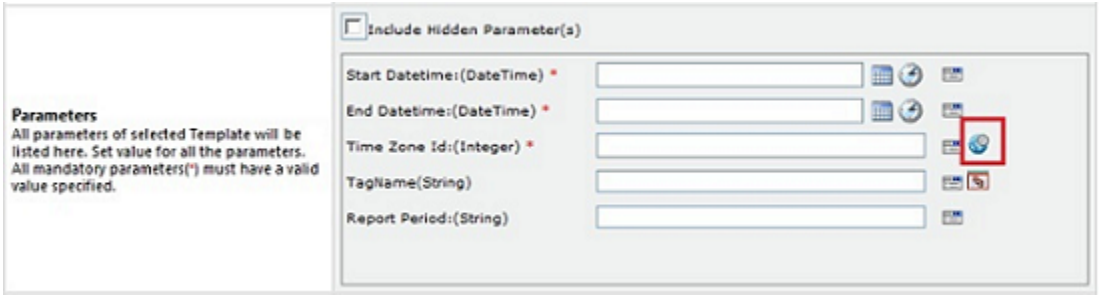
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Among the parameters displayed here, the Time Zone ID and the TagName parameters have specific icons, Time Zone and Historian Tag, displayed against them. Users should note that it is a prerequisite to have a Historian Connection established before selecting a Time Zone or a Historian Tag. For more information on how to establish a Historian Connection, see Create a Historian Connection.

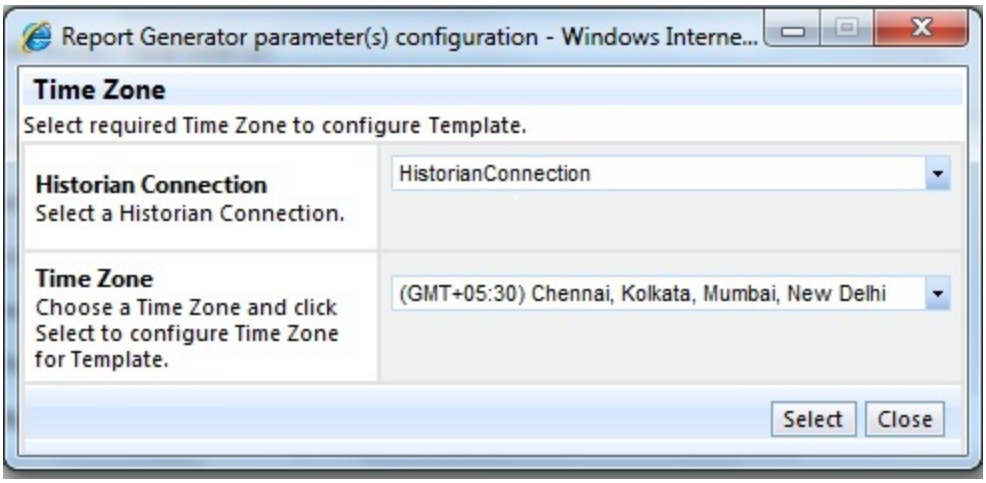
Time Zone Configuration

If a report parameter name is the Time Zone ID, then the Time Zone icon is displayed for that particular parameter as shown in the following image:



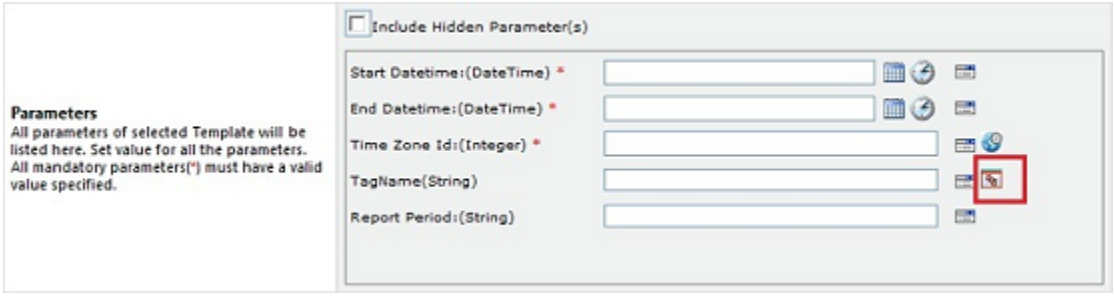
On clicking the Time Zone icon, the Time Zone dialog box appears. Here, you can select the required **Historian Connection**.When you select a connection, different time zones are populated in the Time Zone drop-down list. You can select an appropriate time zone and click **Select**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

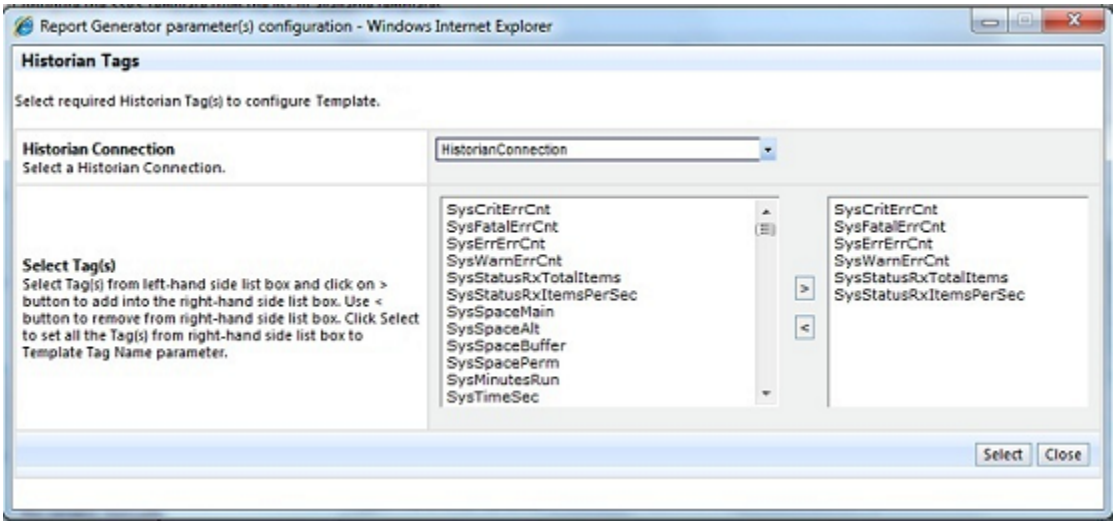


Historian Tag Configuration

If a report parameter name is TagName, then the Historian Tag icon is displayed for that particular parameter as shown in the following image:



On clicking the **Historian Tag** icon, the Historian Tags dialog box appears. Here, you can select the required **Historian Connection**. When you select a connection, all the Historian tags are displayed on the left-hand side list box of **Select Tag(s)** section.



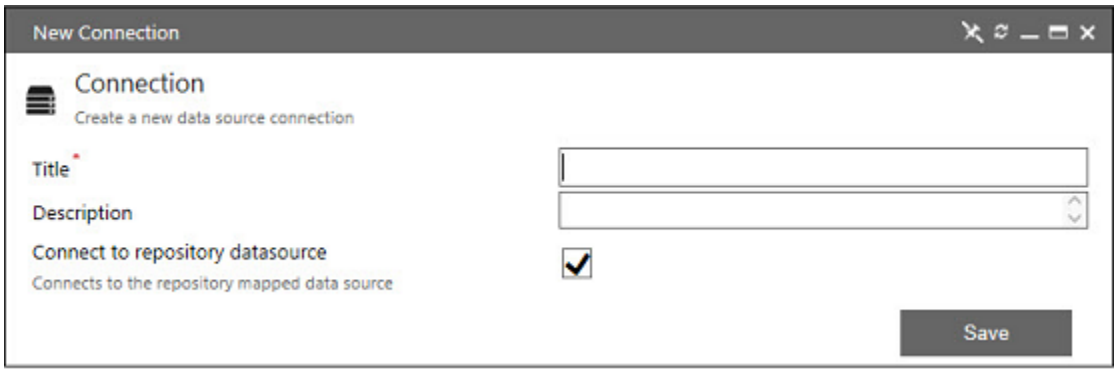
Select the required tags, click the > button to move the tags to the right-hand side of the **Select Tag(s)** section, and click **Select**. If you want to remove any tag from the right-hand side list box, then you must select the tag(s) and click the **<** button. The tag will be placed on the left-hand side list box.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

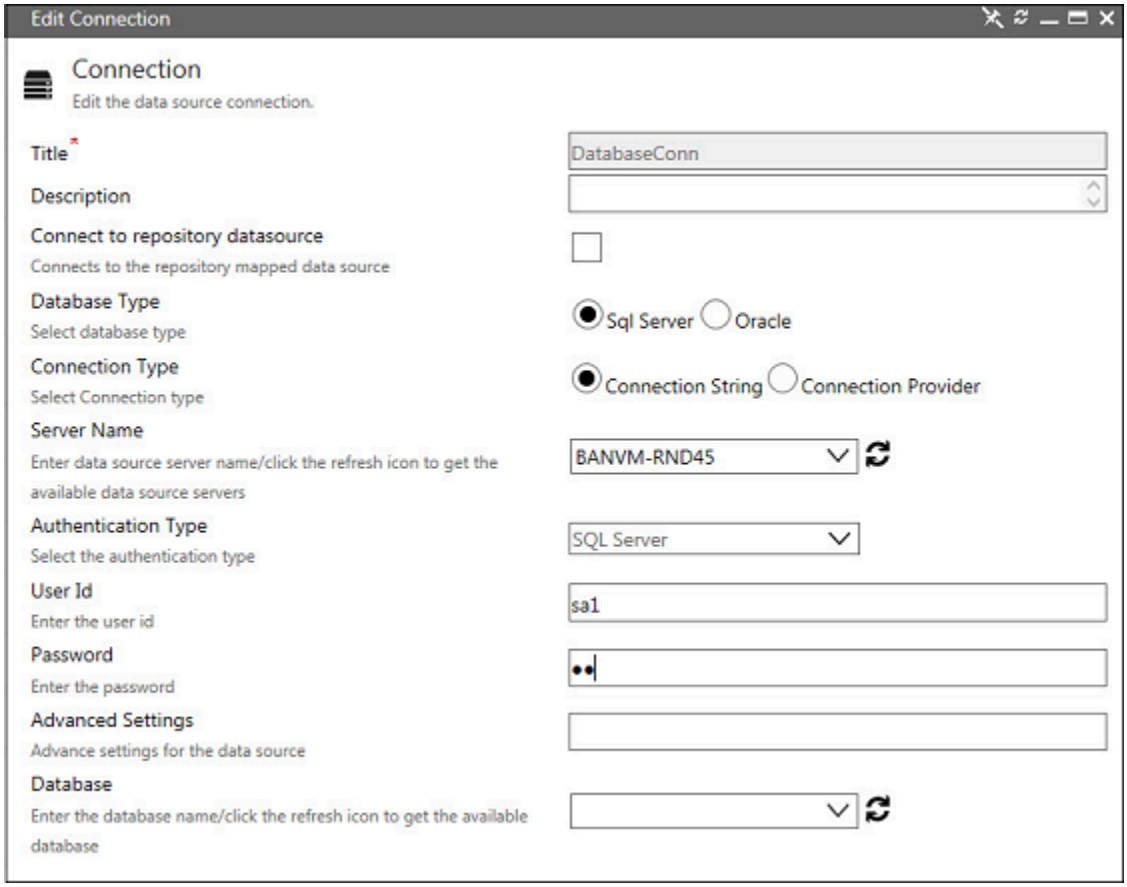
Create a Historian Connection

You can configure the Historian Connection information in the Database Connections Settings in the Enterprise Console.

1. On the **Enterprise Console**, select **Settings**, and then click **Database Connections**. The Database Connection page appears. This page lists the **Connection List**.
2. Click **New**. The **New Connection** dialog box appears.

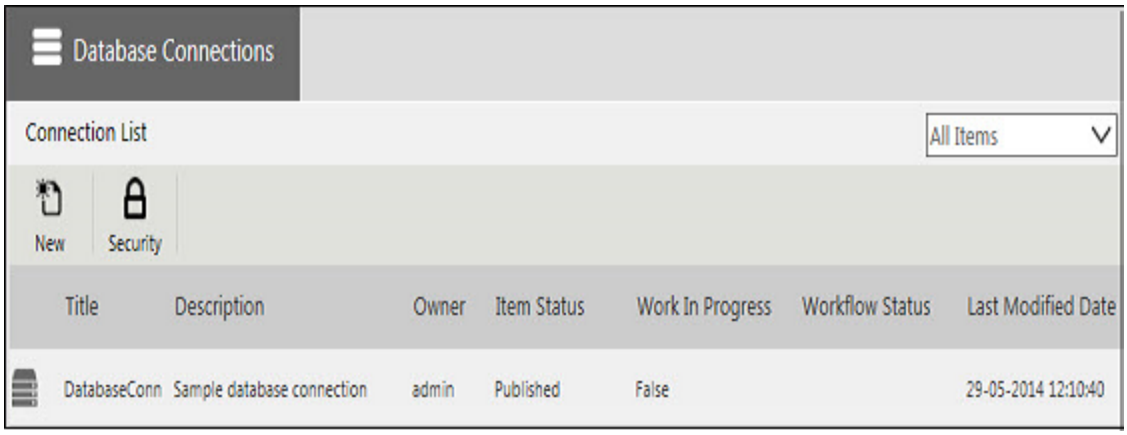


3. Enter the details as shown in the image given below:



4. Then click **Save**. The database connection is saved and will appear in the Connection List page.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



SOA Activities

This section provides information on the SOA Execute activity.

SOA Execute Activity

Under the SOA Execute activity, you can:

- Execute methods with the out parameters in SOA.
- Retrieve multiple records from the database using SOA.
- Iterate a collection object.
- Design a workflow.

Execute Methods with Out Parameters in SOA

Use Out Parameter in SOA

Out Parameter: The out parameter can be used to return more than one value in the same variable passed as a parameter of the method. Any changes made to the parameter will be reflected in the variable.

If an Assembly has a method with the Out Parameter, then you can make use of the Out Parameter to update the variables or for building the Expression through SOA.

To execute the methods with Out Parameter in SOA, follow the steps given below:

1. Prepare an Assembly with the following code.

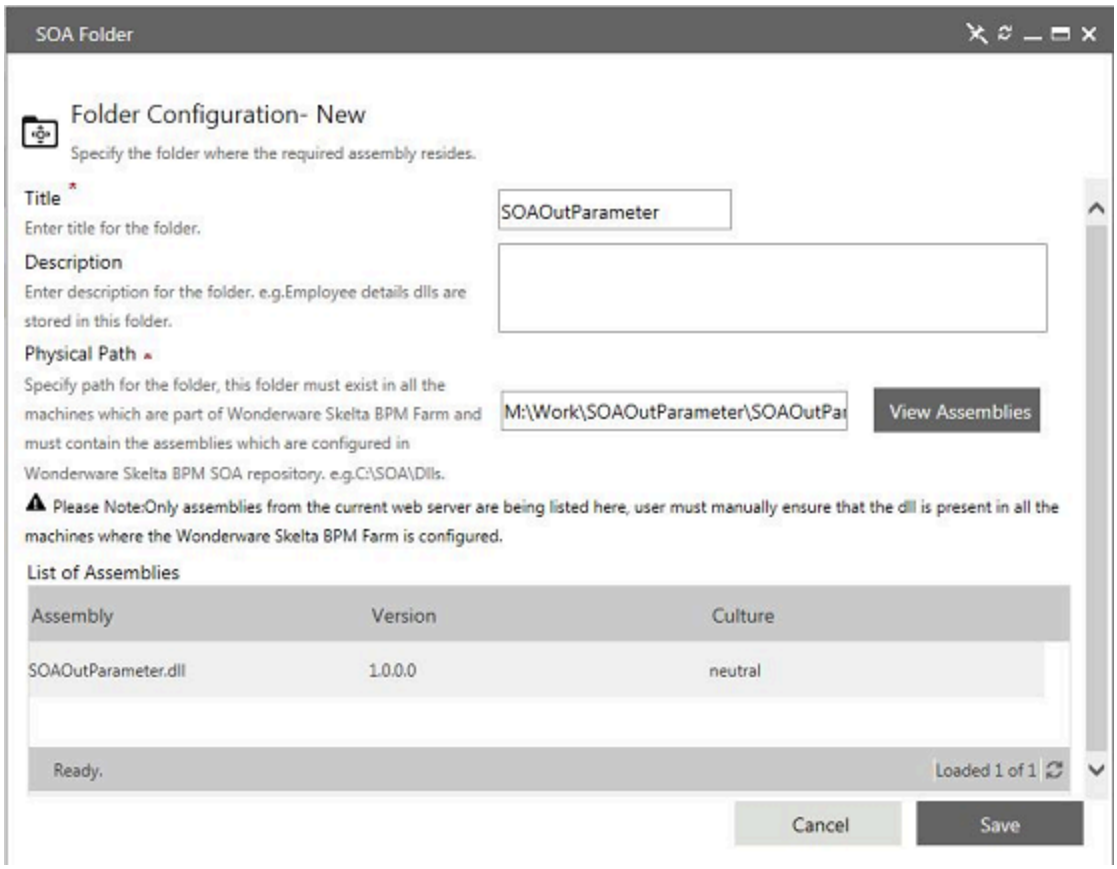
```
using System;
using System.Collections.Generic;
using System.Text;

namespace FinalTest
{
public class SOAOutParameter
{
private int _a;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

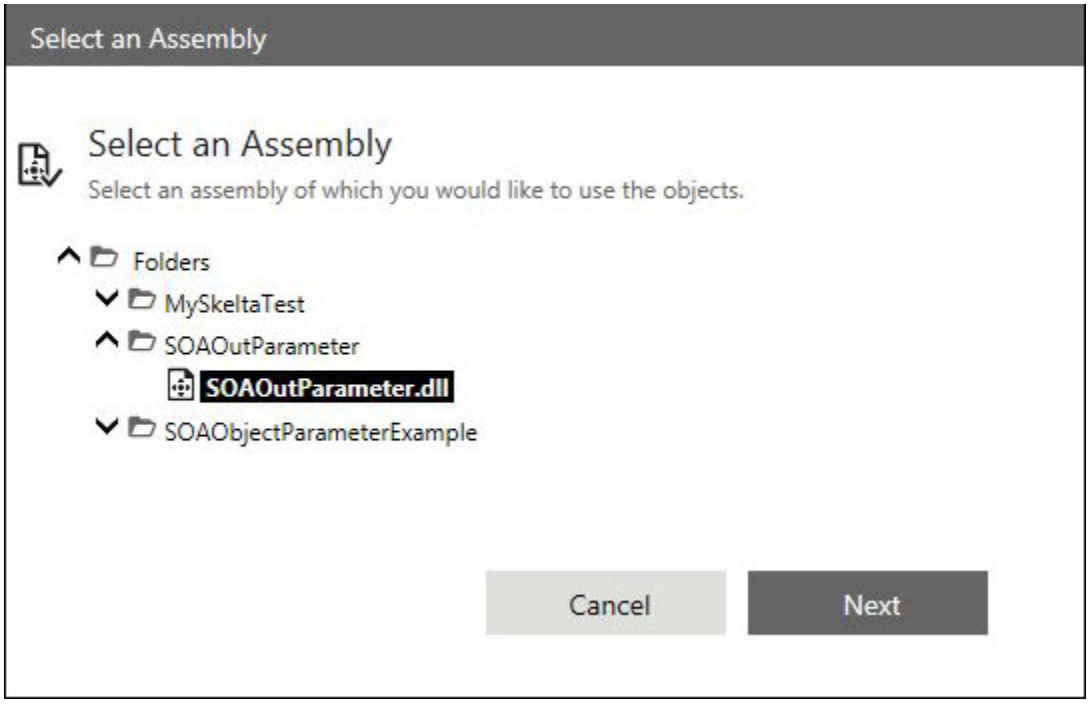
```
private int _b;
public int A
{
get
{
return _a;
}
set
{
_a = value;
}
}

public int B
{
get
{
return _b;
}
set
{
_b = value;
}
}

public SOAOutParameter(int a, int b)
{
this.A = a;
this.B = b;
}

public void AddTwoNumbers(out int Result)
{
Result = this.A + this.B;
}
}
}
```

2. You have to register the assembly in SOA Assembly to use members of the assembly in the workflow.
3. To register a assembly, select **Settings > SOA Folder > New Assembly Folder** from the **Enterprise Console** pull-down menu. Register an assembly called 'SOAOutParameter'.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

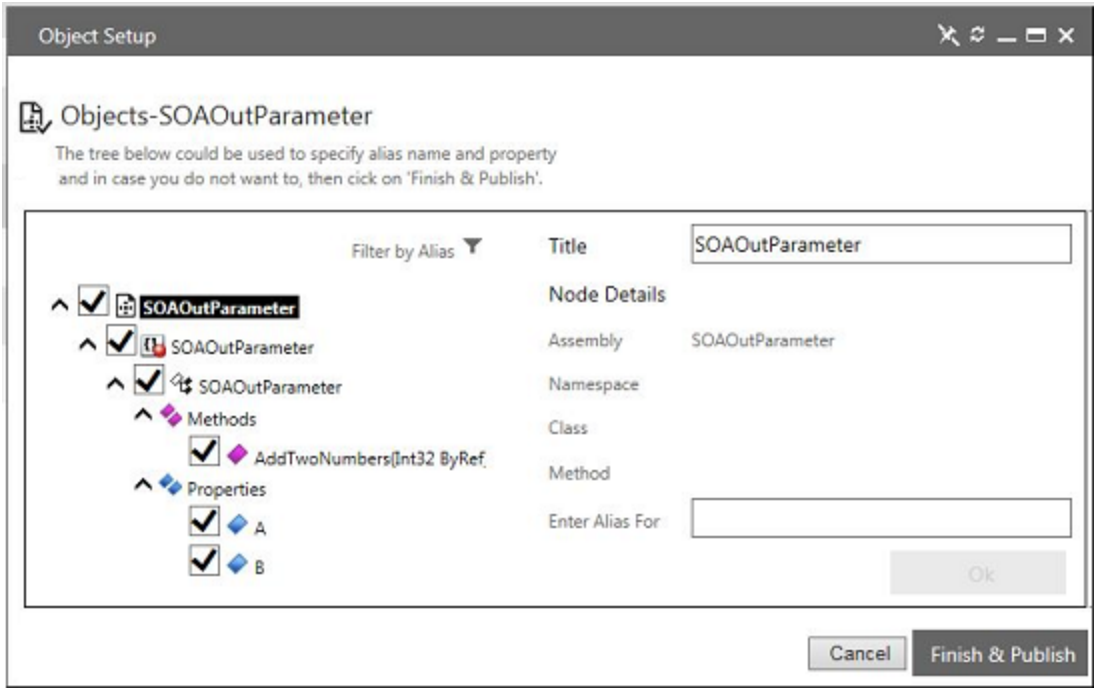


4. Click **Save**. Now the SOAOutParameter assembly has been registered as shown below.
5. Select **Tools > SOA Assembly > From Assembly Folder** from the **Enterprise Console** pull-down menu. Expand SOAOutParameter and select dll and click **Next**.



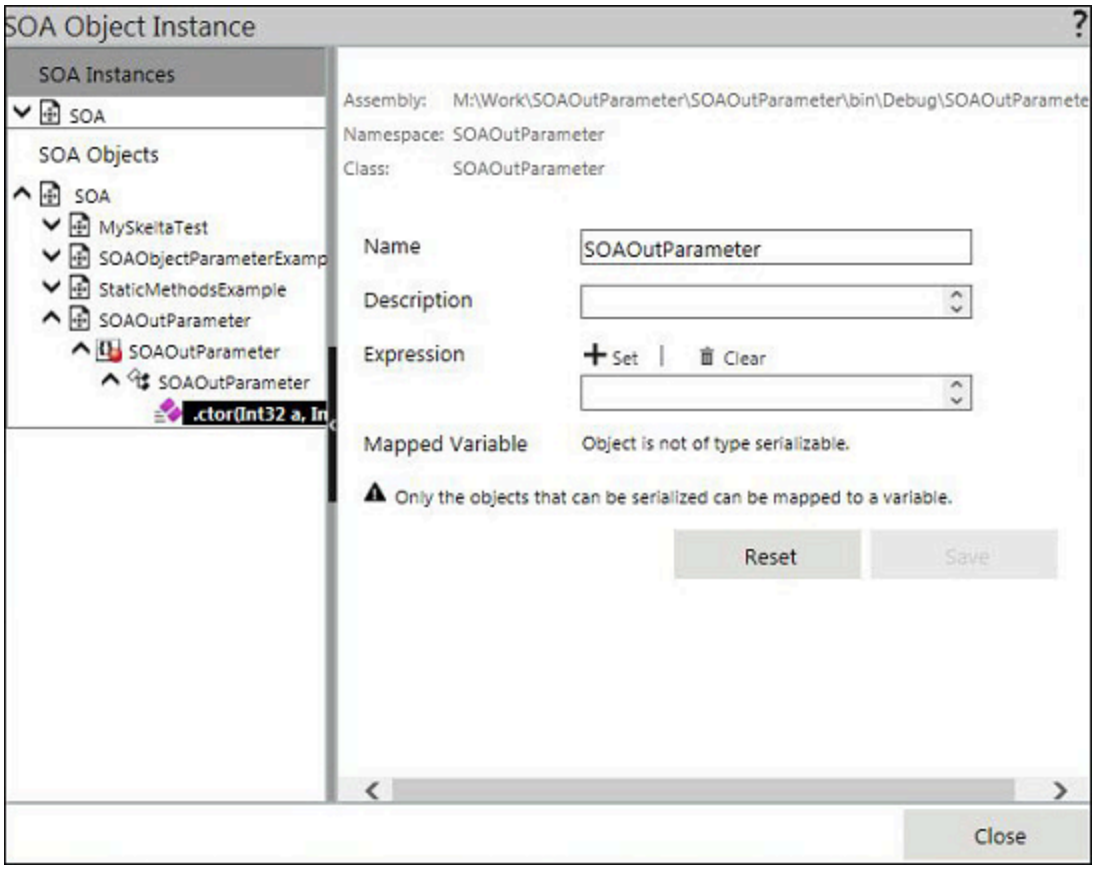
6. The **Object Setup** screen is displayed. Expand all class, methods and properties as shown below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



7. Click Finish & Publish to save the assembly. This will be listed in the Assembly List as shown below.
8. You need to create an instance for the class to use the members of the class. To make use of the static members, there is no need to create any instance. Static members can be directly used in the Expression Editor.
9. To create an instance, you first need to design a workflow.
10. Right-click the Start Activity and then select Activity Properties.
11. Select the SOA Object Instances property.
12. The SOA Object Instance screen is displayed. SOAOutParameter is the Assembly, under that SOAOutParameter as NameSpace, under that SOAOutParameter as Class and under that .ctor(Int32 a, Int32 b) is the constructor as shown in the following figure.

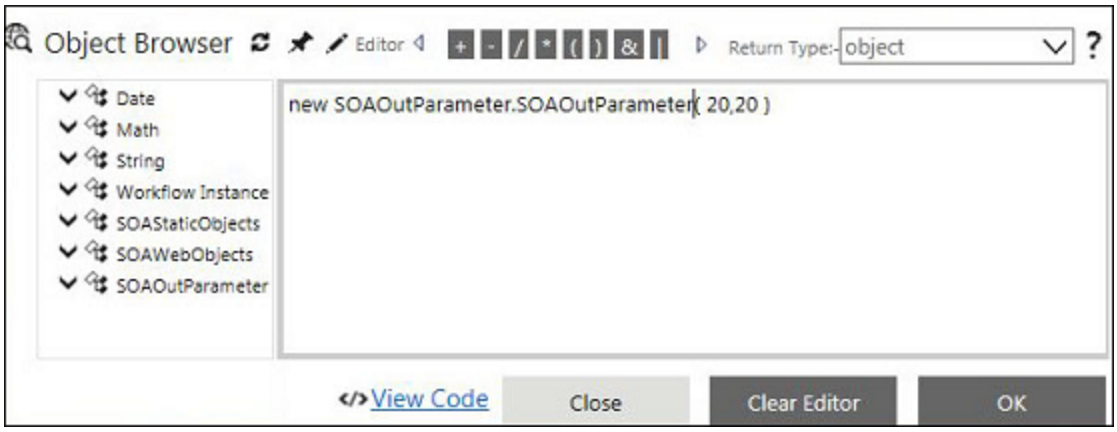
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



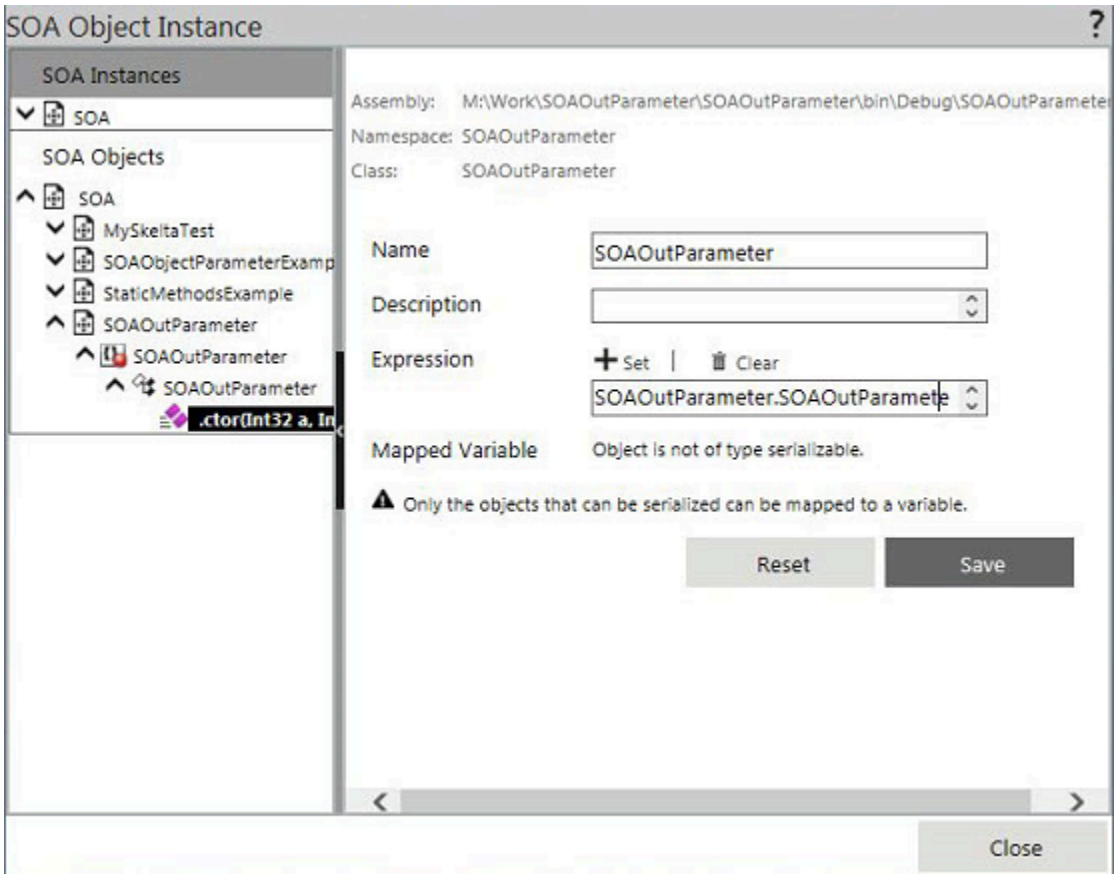
13. Click the Constructor and then click Set to set the constructor values. The Constructor will be displayed in the Expression Editor.

14. Pass the parameter value to the Constructor. For example, pass (20, 20) as the parameter for the Add Method. Click Save.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

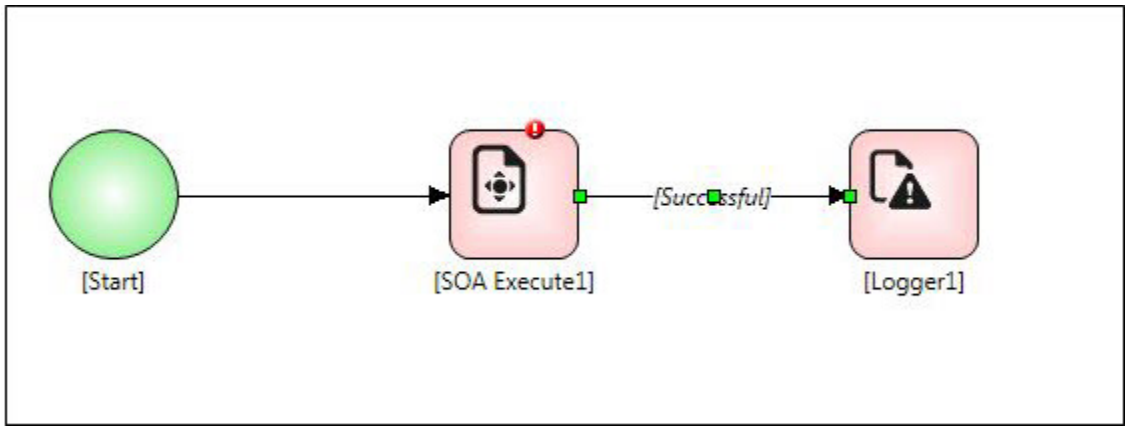


15. Click Save in the Activity Properties window.

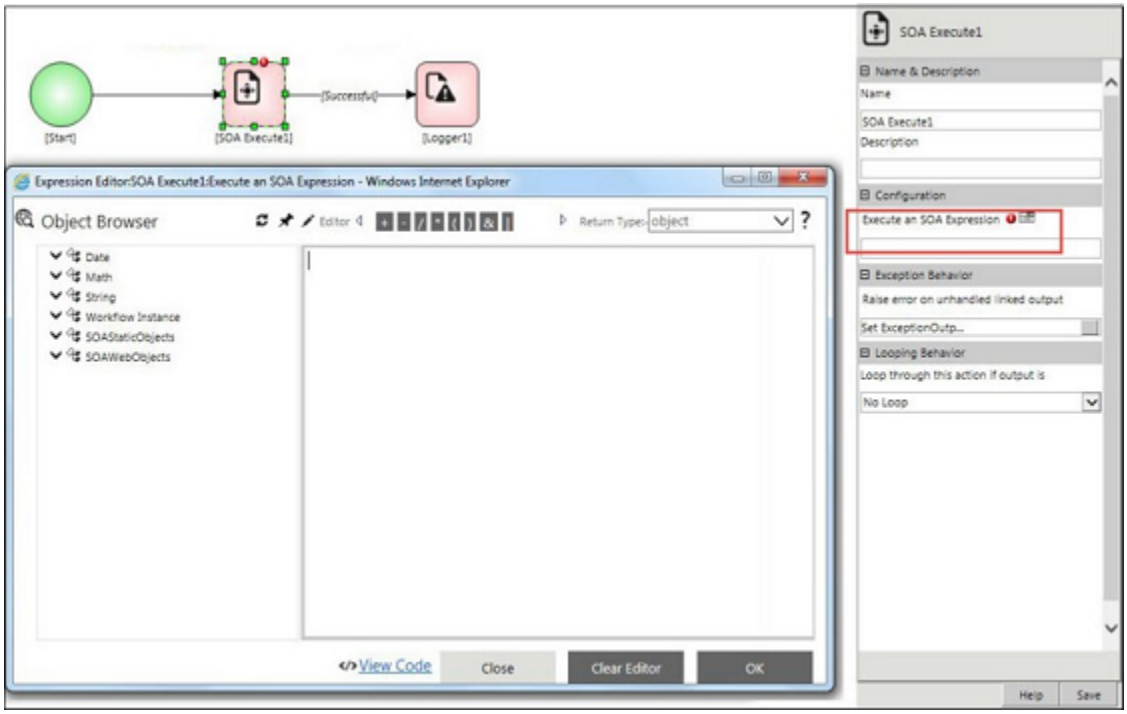


© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

16. Make use of SOAExecute activity to execute the method and update the Variable 'varResult'.



17. Right-click the SOA Execute activity and select Activity Properties.
18. Select Execute an SOA Expression property to build the expression.

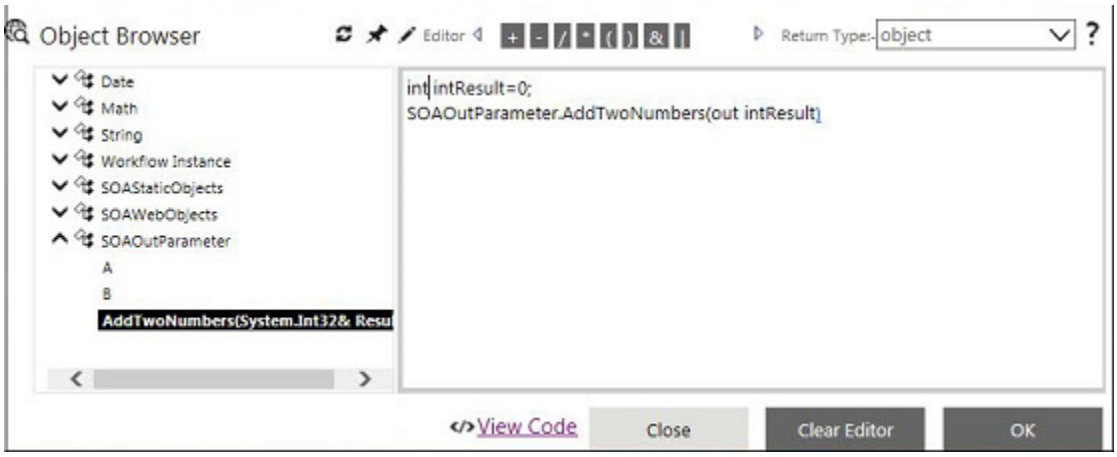


19. Click the AddTwoNumbers method to bring it into the Expression Editor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

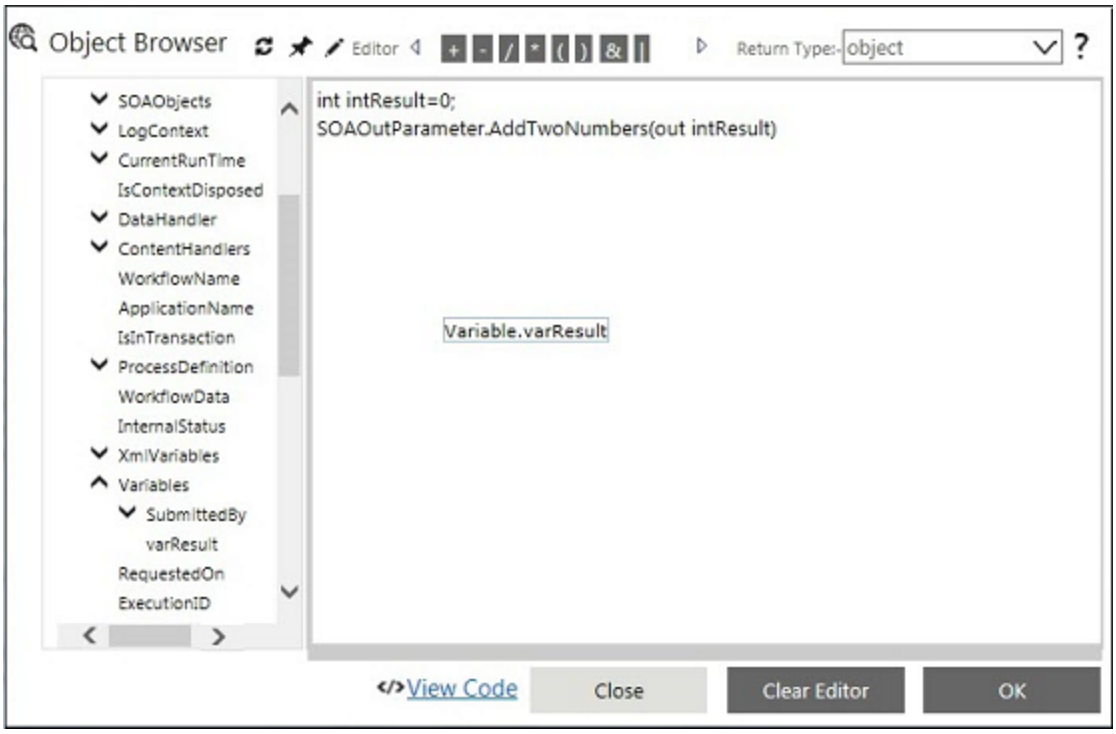
20. AddTwoNumbers is of Void as return type and expects integer as parameter with out keyword.

21. Declare a variable called intResult with value as '0'. Pass 'intResult' as parameter to AddTwoNumbers method with 'out' keyword, as shown in the following figure.

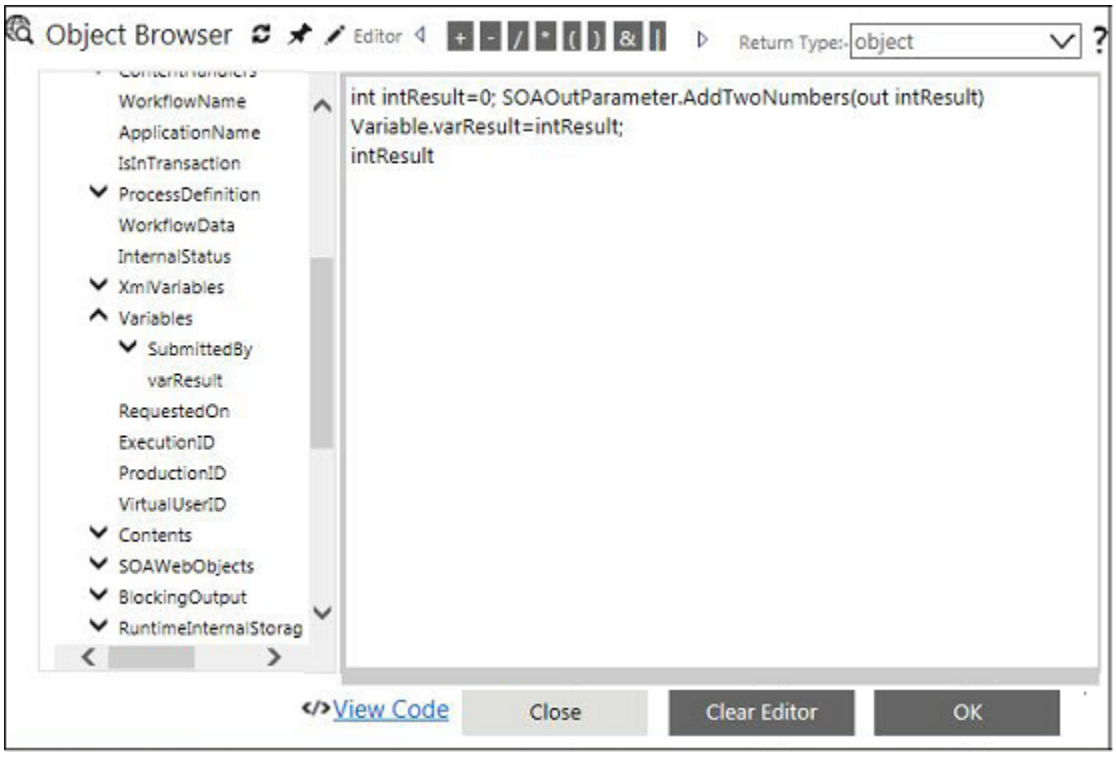


22. Drag and drop the Variable to be updated with the out return value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



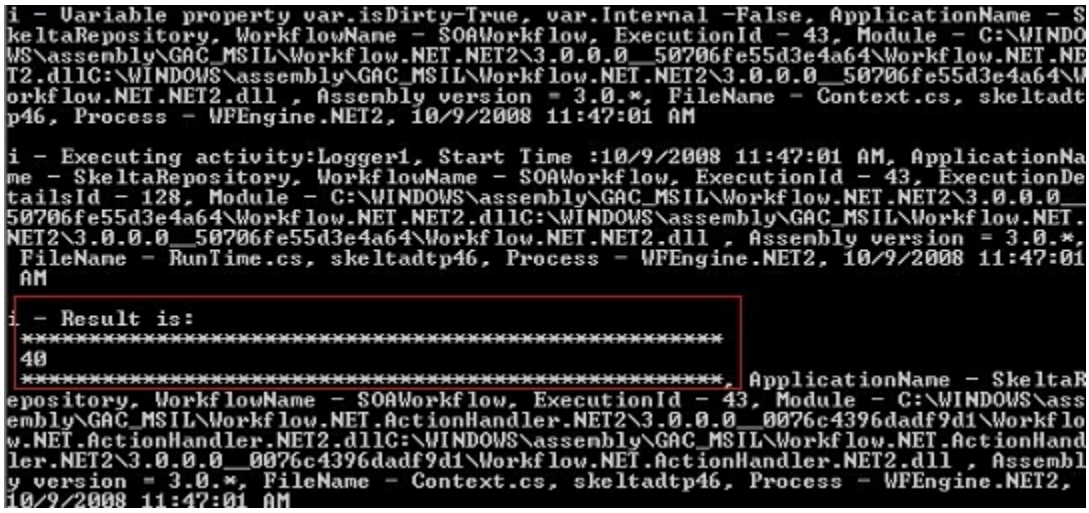
23. Here the variable to be updated is 'intResult'.



24. Click **OK** and then click **Save** in the Activity Properties window.
25. Display the value in the logger using Logger activity. Right-click the Logger activity and select Activity Properties. Select LogMessage property and then type the logger message in the Logger Message window.
26. Select LogMessage property and then type the logger message in the Logger Message window.
27. Click Update and then click Save in the Activity Properties window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

28. Execute the workflow. The method will be executed at run time and the out parameter will be updated to the Variable.



Retrieve Multiple Records from Database Using SOA

This example illustrates, how to get the multiple records from database by using SOA.

Assume that the following are the database details:

Column Name	Data Type	Allow Nulls
Id	uniqueidentifier	☐
FirstName	nvarchar(50)	☑
MiddleName	nvarchar(50)	☑
LastName	nvarchar(50)	☑
Designation	nvarchar(50)	☑
Address	nvarchar(500)	☑
Salary	decimal(18, 2)	☑
DateOfJoining	datetime	☑
		☐

The following is the SQL script:

```
USE [EmployeeDetails]
GO
/****** Object:  Table [dbo].[Employee]  Script Date: 10/09/2008 14:56:15 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Employee](
 [Id] [uniqueidentifier] NULL,
 [FirstName] [nvarchar](50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 [MiddleName] [nvarchar](50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 [LastName] [nvarchar](50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 [Designation] [nvarchar](50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 [Address] [nvarchar](500) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
 [Salary] [decimal](18, 2) NULL,
 [DateOfJoining] [datetime] NULL
) ON [PRIMARY]
```

Assume that there are three records in the Employee Table of EmployeeDetails Database.

To get the multiple records from database through SOA, follow the steps given below:

1. Prepare an Assembly with the following sample code for getting multiple records.

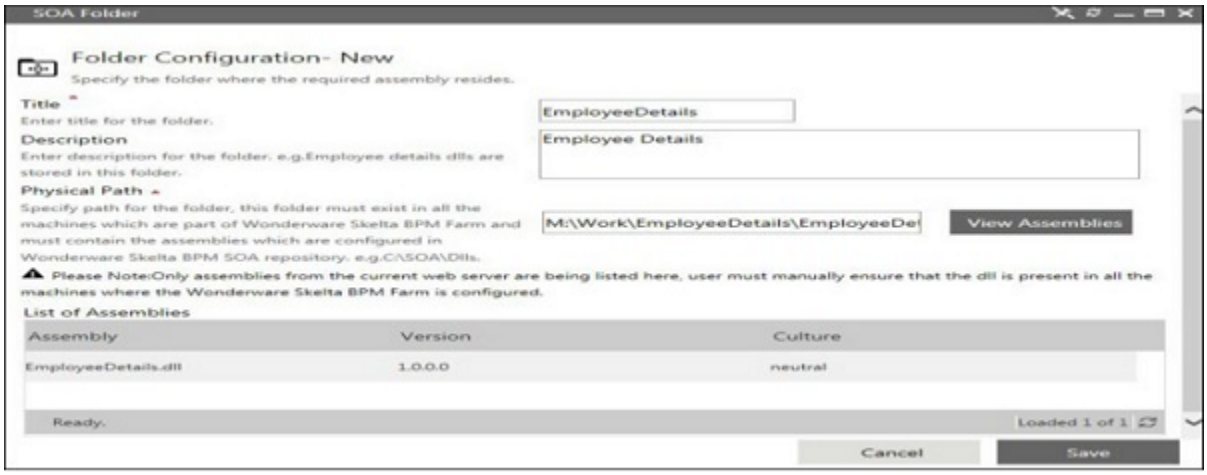
```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data;
using System.Data.SqlClient;

namespace EmployeeDetails
{
public class Employee
{
private string _connectionString = string.Empty;
public string ConnectionString
{
get
{
return _connectionString;
}
set
{
_connectionString = value;
}
}

private string _tableName = string.Empty;
public string TableName
{
get
{
return _tableName;
}
set
{
_tableName = value;
}
}

public Employee(string server, string database, string tablename)
{
this.ConnectionString = "server=" + server + ";database=" + database + ";Integrated
Security=true;pooling=false;";
this.TableName = tablename;
}

public string GetEmployeeDetails()
{
string connectionString = this.ConnectionString;
string strQry = @"SELECT * FROM " + this.TableName;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
SqlConnection connection = new SqlConnection(connectionString);
SqlDataAdapter sqlDataAdapter = new SqlDataAdapter();
sqlDataAdapter.SelectCommand = new SqlCommand(strQry, connection);
connection.Open();
DataSet dataSet = new DataSet();
sqlDataAdapter.Fill(dataSet, this.TableName);
connection.Close();
return dataSet.GetXml();
}

public string GetEmployeeSchema()
{
string connectionString = this.ConnectionString;
string strQry = @"SELECT * FROM " + this.TableName;
SqlConnection connection = new SqlConnection(connectionString);
SqlDataAdapter sqlDataAdapter = new SqlDataAdapter();
sqlDataAdapter.SelectCommand = new SqlCommand(strQry, connection);
connection.Open();
DataSet dataSet = new DataSet();
sqlDataAdapter.Fill(dataSet, this.TableName);
connection.Close();
return dataSet.GetXmlSchema();
}
}
}
```

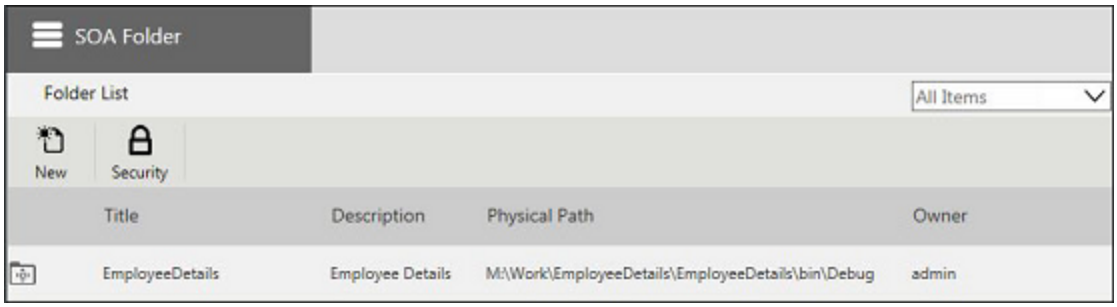
The above code having two methods. One method is to get the Database Schema and other method is to get the raw XML. By passing the Server, Database and the Table name to the constructor it will create the object by that you can access the data from the database.

2. You have to register the assembly in SOA Assembly.
3. To register a assembly, select **Settings ® SOA Folder ® New Assembly Folder** from the **Enterprise Console** pull-down menu. Register a assembly called 'EmployeeDetails'.

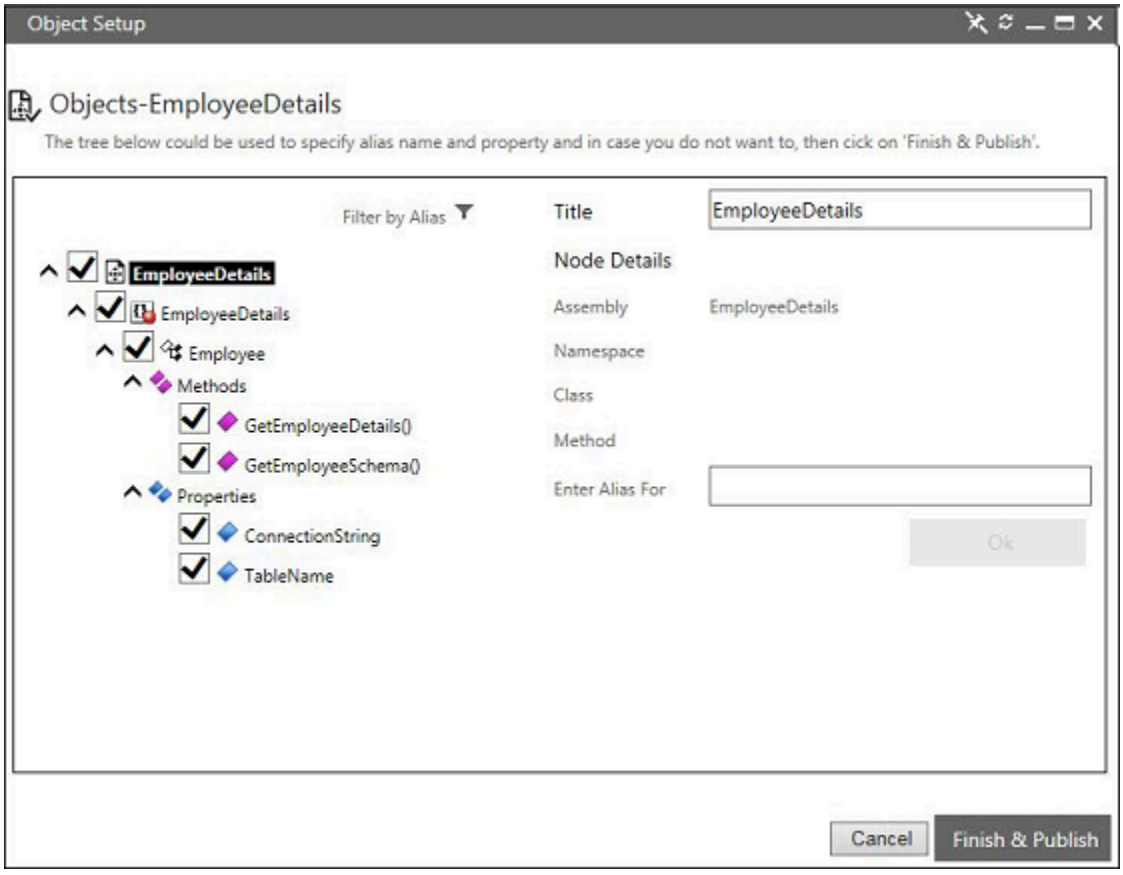


4. Click **Save**. Now the EmployeeDetails assembly has been registered as shown below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

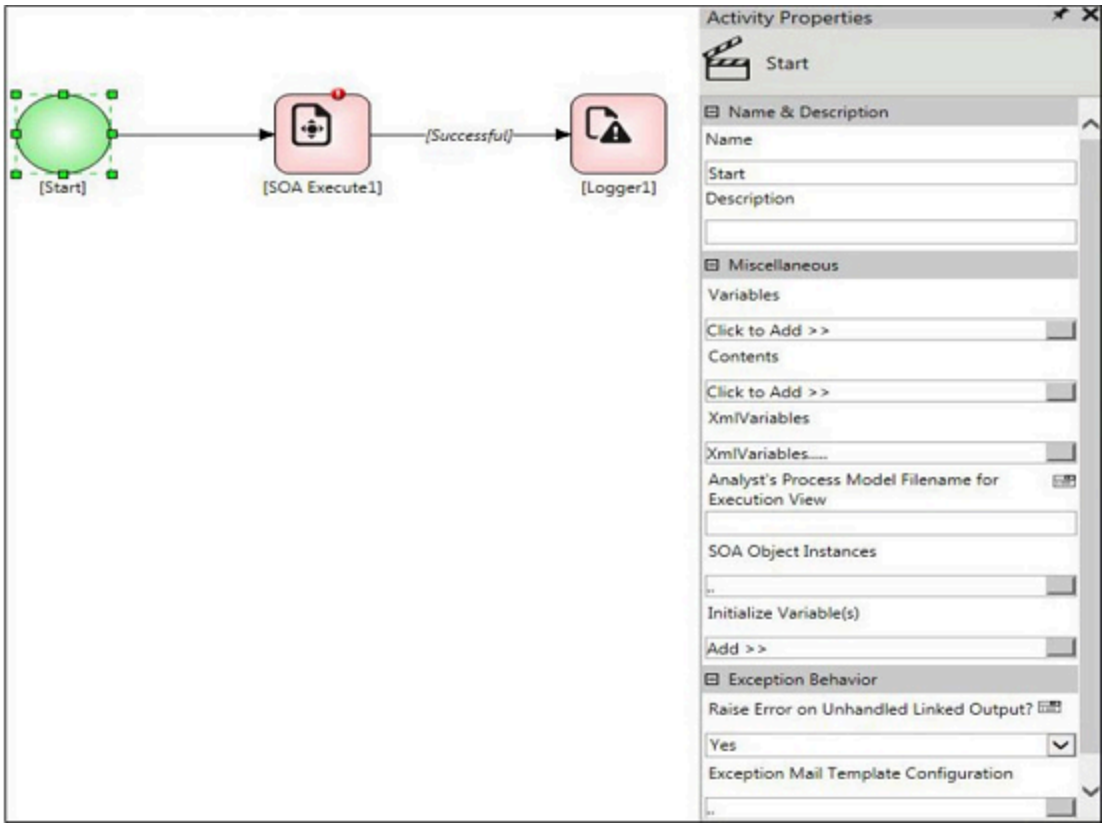


5. Select **Tools > SOA Assembly > From Assembly Folder** from the **Enterprise Console** pull-down menu. Expand EmployeeDetails and select dll and then click Next.
6. The **Object Setup** screen is displayed. Expand all class, methods and properties as shown below.

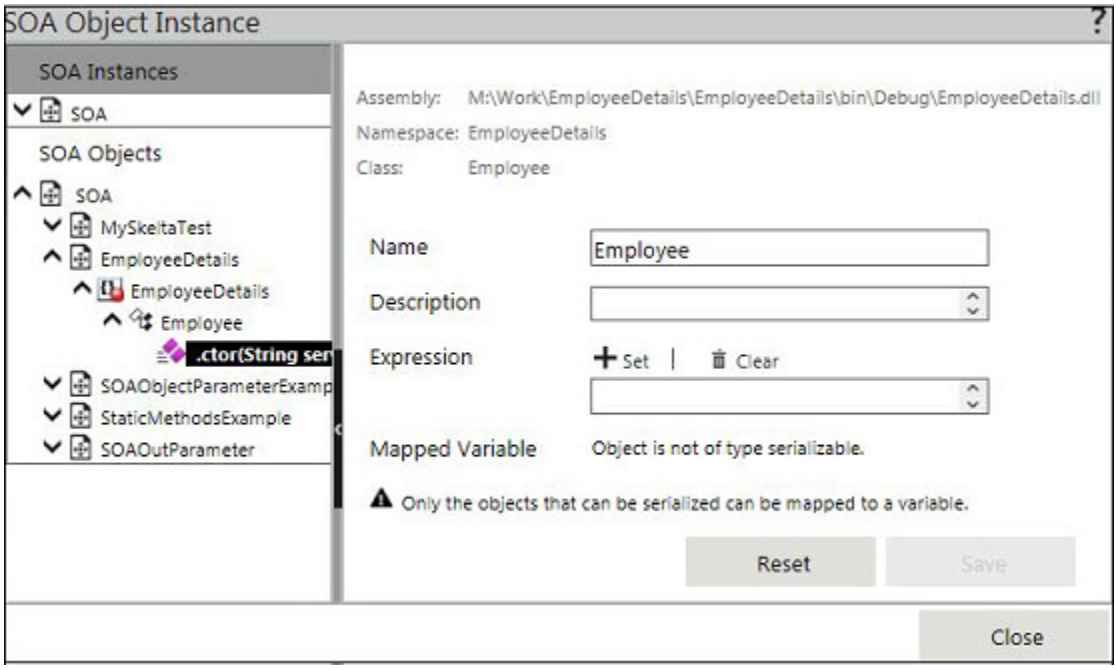


7. Click Finish & Publish to save the assembly. This will be listed in the Assembly List.
8. Design a workflow called 'Employee Details' as shown below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

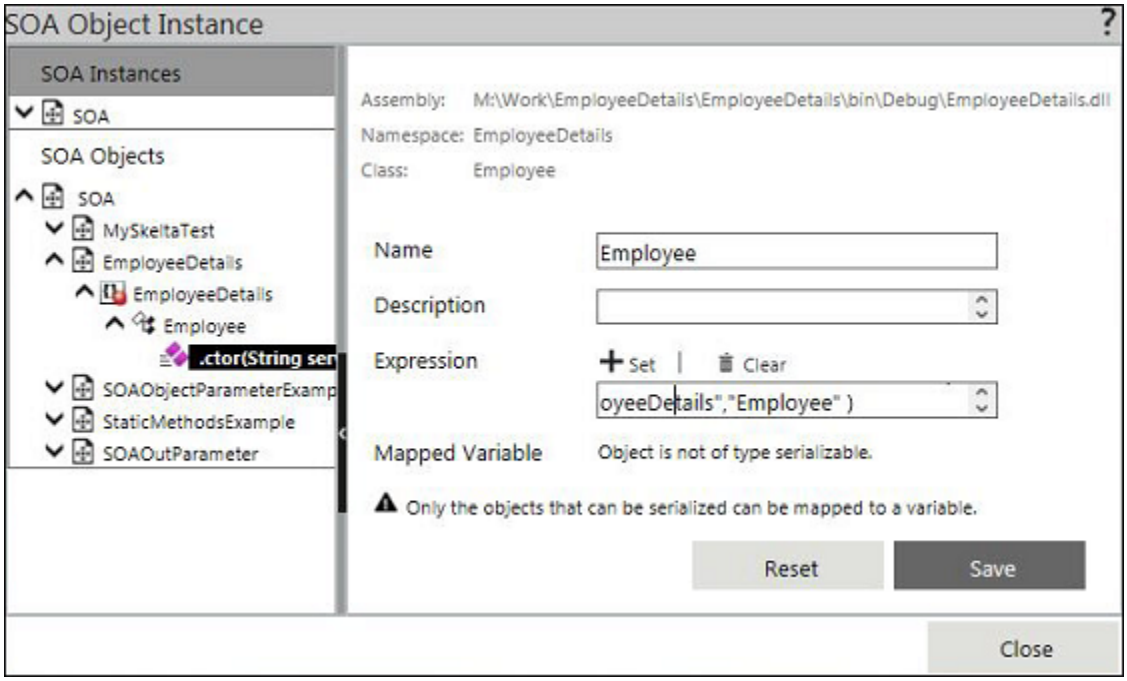


9. Right-click the Start Activity and then select Activity Properties.
10. Select the SOA Object Instances property.
11. The SOA Object Instance screen is displayed. Drag and drop the constructor of the class in SOA Property editor. Here, the constructor takes three parameters with Server name, Database name, and Table name.



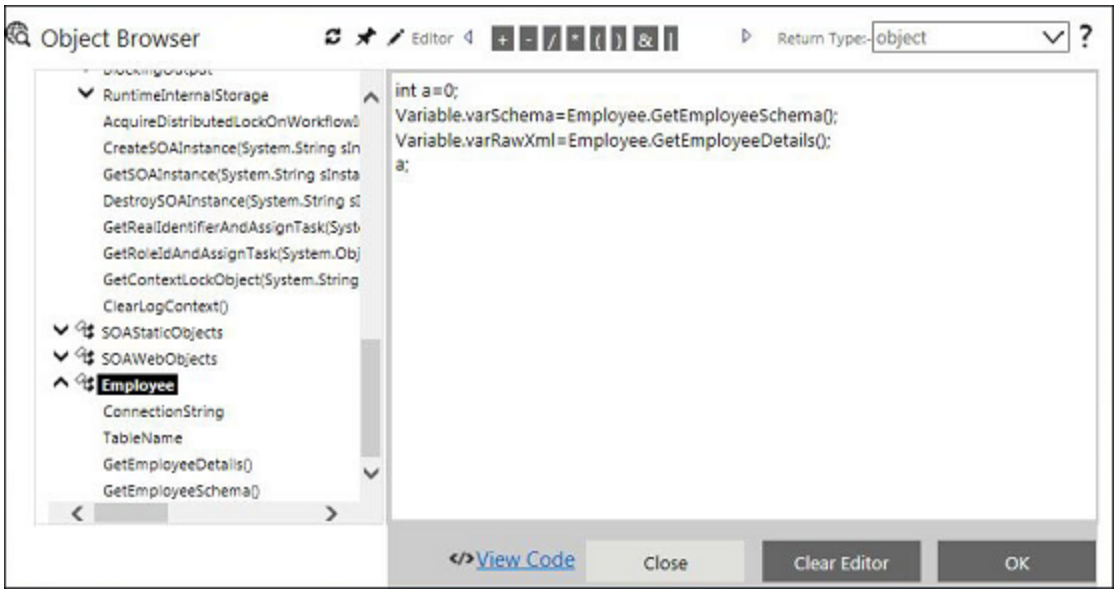
12. Pass the parameter to the constructor. Save the constructor. After saving the instance details, all the members of the class will be listed in the Expression Editor.

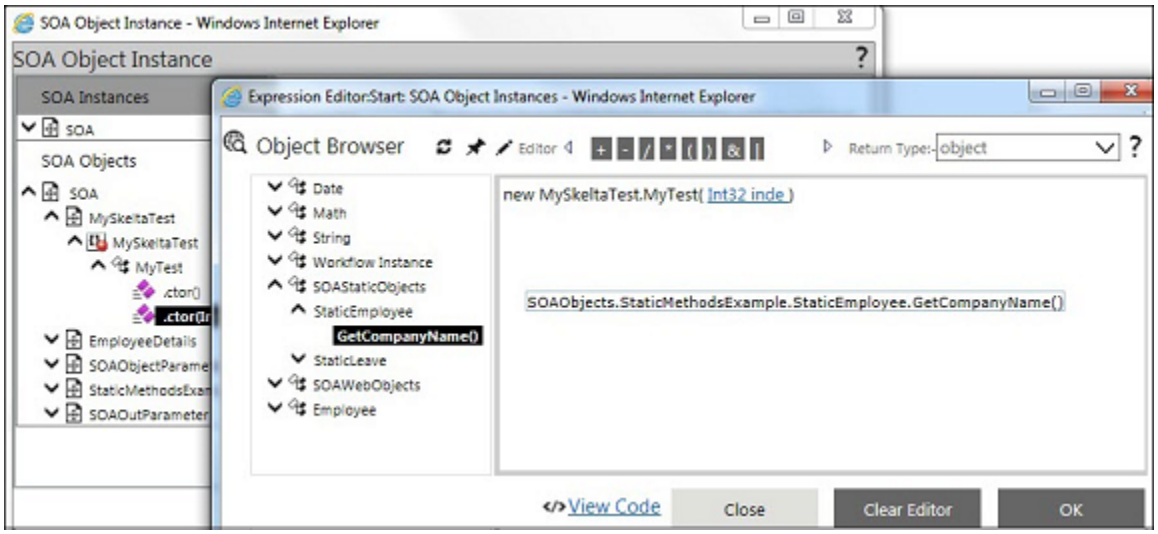
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

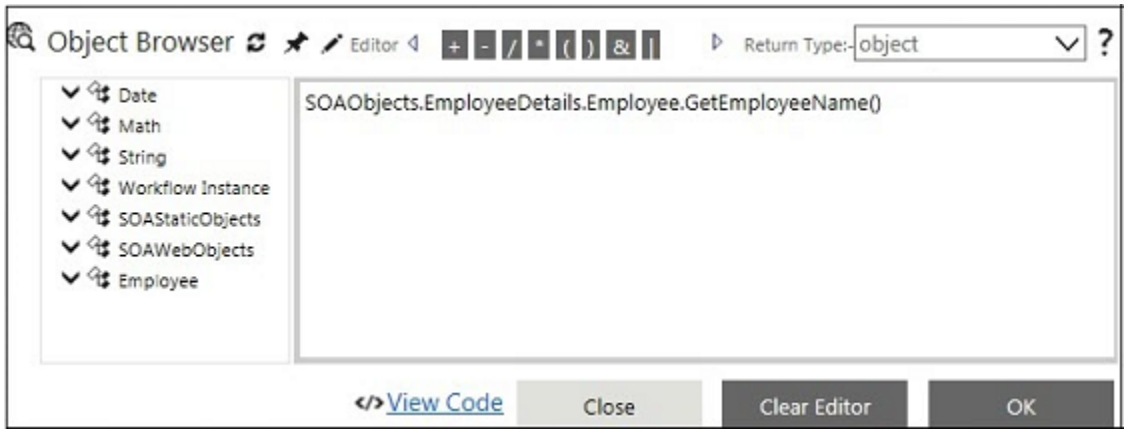


13. Click Save in the Activity Properties window.
14. Make use of the SOAExecute activity to execute the method and update the Variable with the Schema and the Raw XML details.
15. Right-click the SOA Execute activity and select Activity Properties.
16. Select Execute an SOA Expression property to build the expression.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.





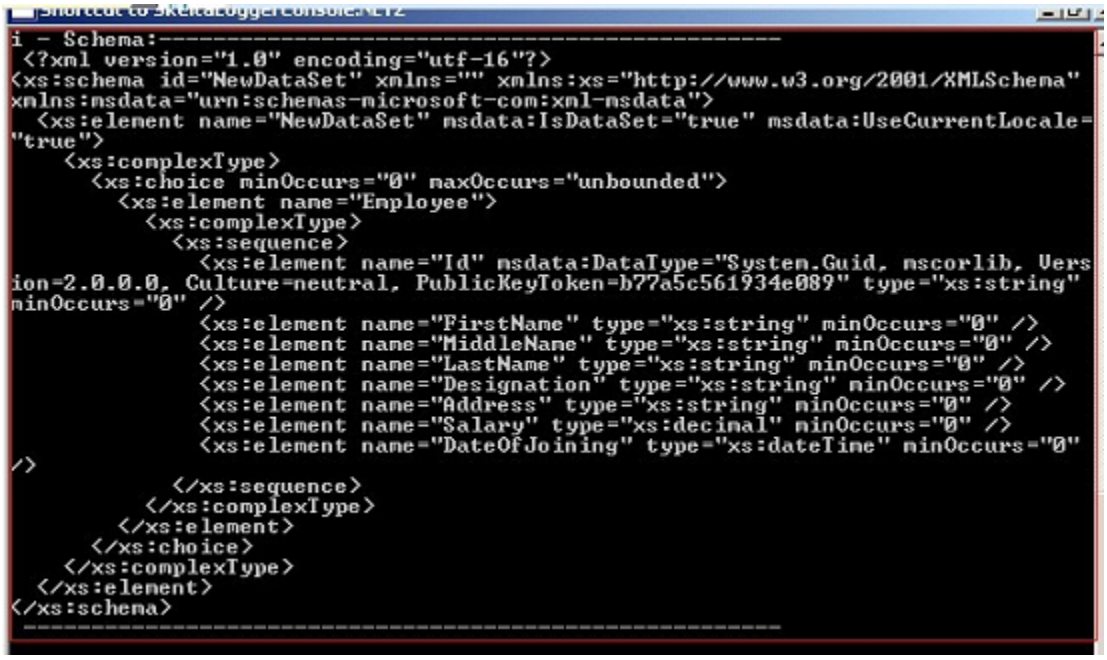


17. Click OK and then click Save in the Activity Properties window.
18. Display the value in the logger using Logger activity. Right-click the Logger activity and then select Activity

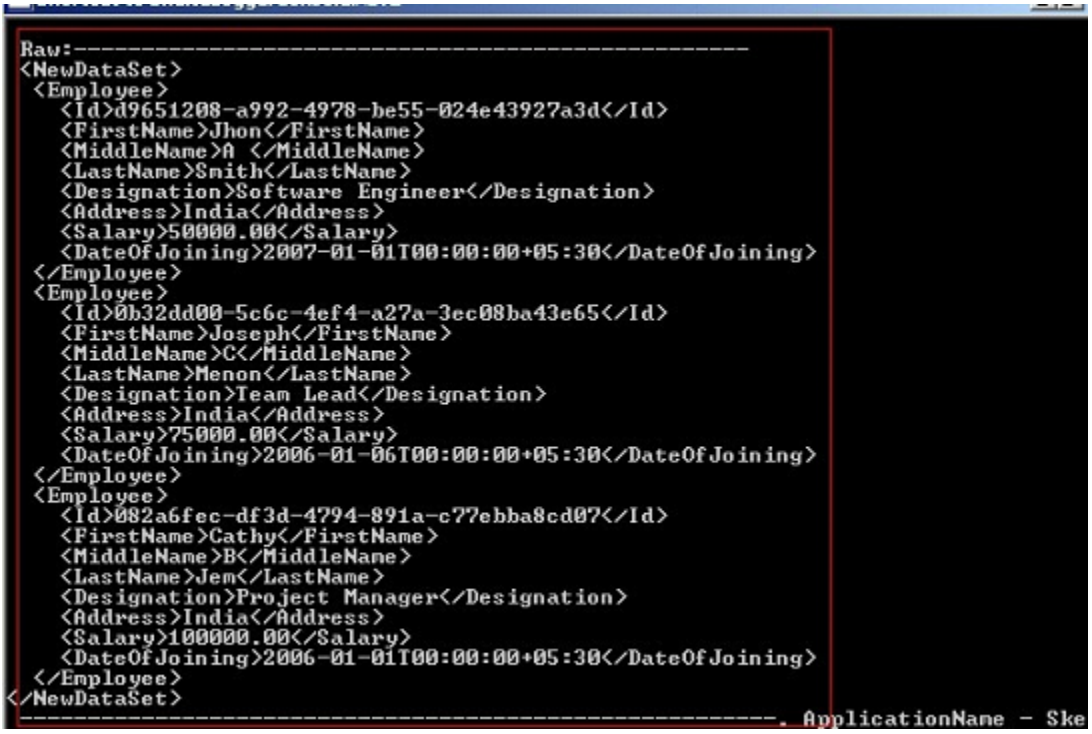
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Properties.

19. Select LogMessage property and then type the logger message in the Logger Message window.
20. Execute the workflow.
21. The following figure shows the logger with the Schema XML.



22. The following figure shows the logger with the Raw XML:



Note: You can even create an XML Variable with the Schema XML and you can bind the data with the Raw XML.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

How to Iterate Collection Object?

This example illustrates how to iterate a collection object. It is possible to persist an object's state in a long running workflow by using variables of type **Object**. In some scenarios, complex objects may be a collection of other complex objects. Here, one may be required to iterate through the collection to get the desired object.

For example, EmployeeCollection object returned from a WCF Service contains collection of an employee object. To get all the information of an employee, you are required to iterate through the employee collection.

This can be done by collective use of some AVEVA Work Tasks activities as described below. It shows how one can iterate through the collection object in a workflow. This enhances the capability of holding the collection object in variables of Object type and iterating through the collection for using the items in the collection object.

Scenario

Employee collection object, which is returned from WCF Service, is to be iterated for displaying employee information.

Steps to design the scenario

- Create a class with Employee object collection.

  Note: You can make use of an existing class to achieve the scenario.

Employee class: Holds the employee information.

EmployeeCollection class: Holds the Employee object collection i.e., collection of employee information.

Note: Only the objects that can be serialized are persisted in long running workflows. Objects that need to be stored in the variables should be marked as Serializable.

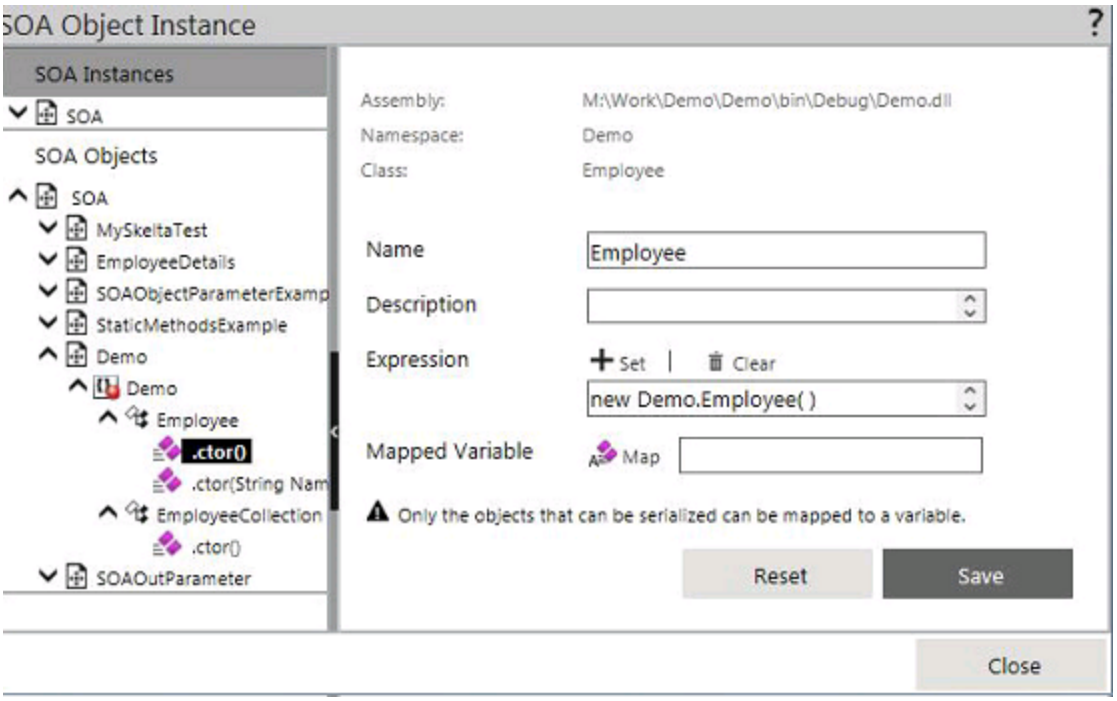
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Demo
{
[Serializable]
public class Employee
{
string _name = string.Empty;
int _age = 0;
string _employeeId = string.Empty;
public string Name
{
get {return _name ;}
set { _name = value;}
}
public int Age
{
get { return _age; }
set { _age = value; }
}
public string EmployeeId
{
get { return _employeeId; }
set { _employeeId = value; }
}
public Employee()
```

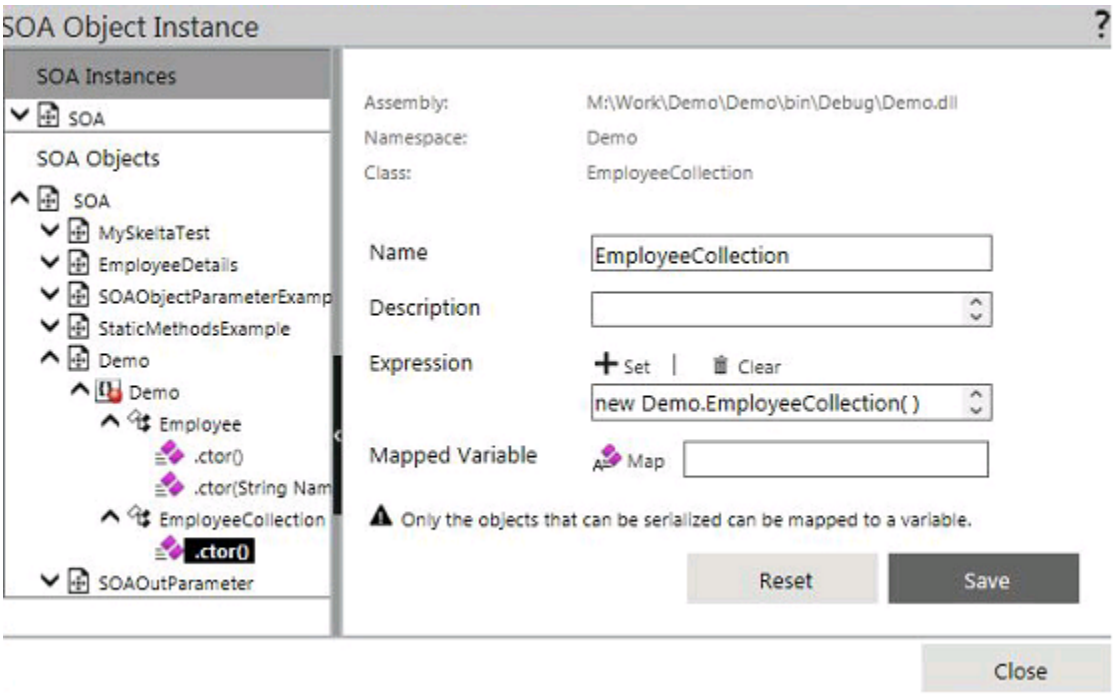
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
}
public Employee(string Name, int Age, string EmployeeId)
{
this._employeeId = EmployeeId;
this._age = Age;
this._name = Name;
}
public Employee GetEmployee()
{
return this;
}
}
[Serializable]
public class EmployeeCollection
{
List<Employee> employeeList = new List<Employee>();
public EmployeeCollection()
{
this.employeeList.Add(new Employee("Kevin",30,"SK001"));
this.employeeList.Add(new Employee("Albert",30,"SK002"));
this.employeeList.Add(new Employee("Jim",30,"SK003"));
this.employeeList.Add(new Employee("Jo",30,"SK004"));
this.employeeList.Add(new Employee("Roger",30,"SK005"));
this.employeeList.Add(new Employee("John",30,"SK006"));
}
public List<Employee> GetEmployees()
{
return this.employeeList;
}
}
}
}
```

- Use the code given above to create an assembly.
- Place the assembly in SOA folder, and register the assembly in SOA Assemblies.
- Create object instances for **Employee** and **EmployeeCollection** class in the SOA property editor.

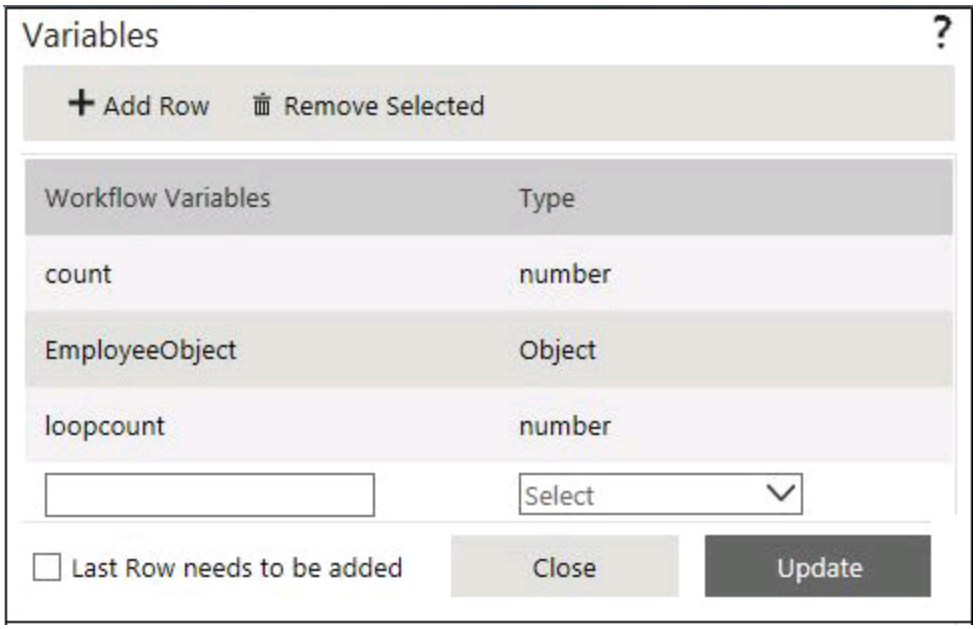
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.





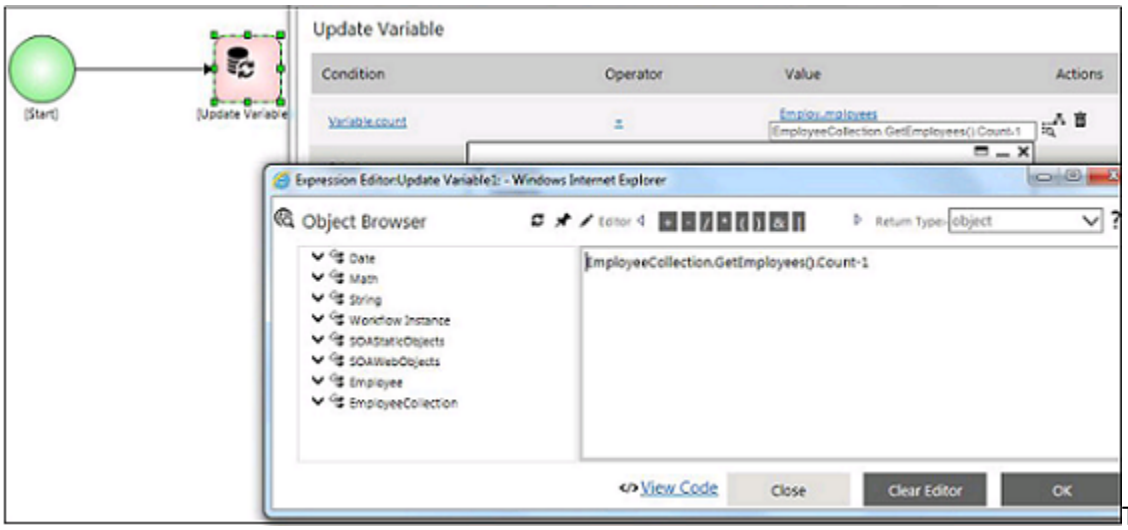
- Create variables in the Start action.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Variable description

- **count**: To keep track of the iteration
- **EmployeeObject**: To store the employee details
- **loopcount**: Total number of times the loop needs to be executed

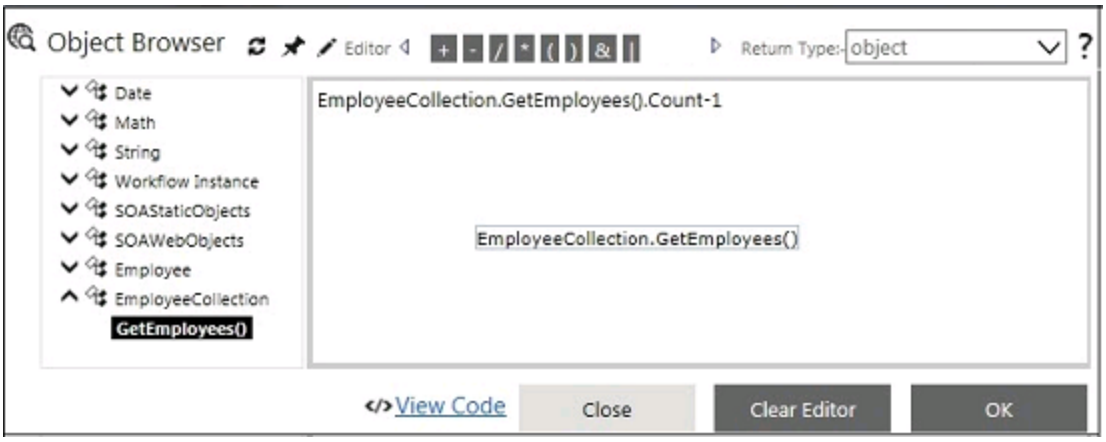
- Add an **Update Variable** activity next to the **Start** activity. This activity is used to update the **loopcount** variable for the total number of times the **EmployeeCollection** object is to be iterated.
- 

Total count can be derived from the **EmployeeCollection** object listed in the expression editor. As the GetEmployees() method returns the collection, you can get the total count of the collection from here.

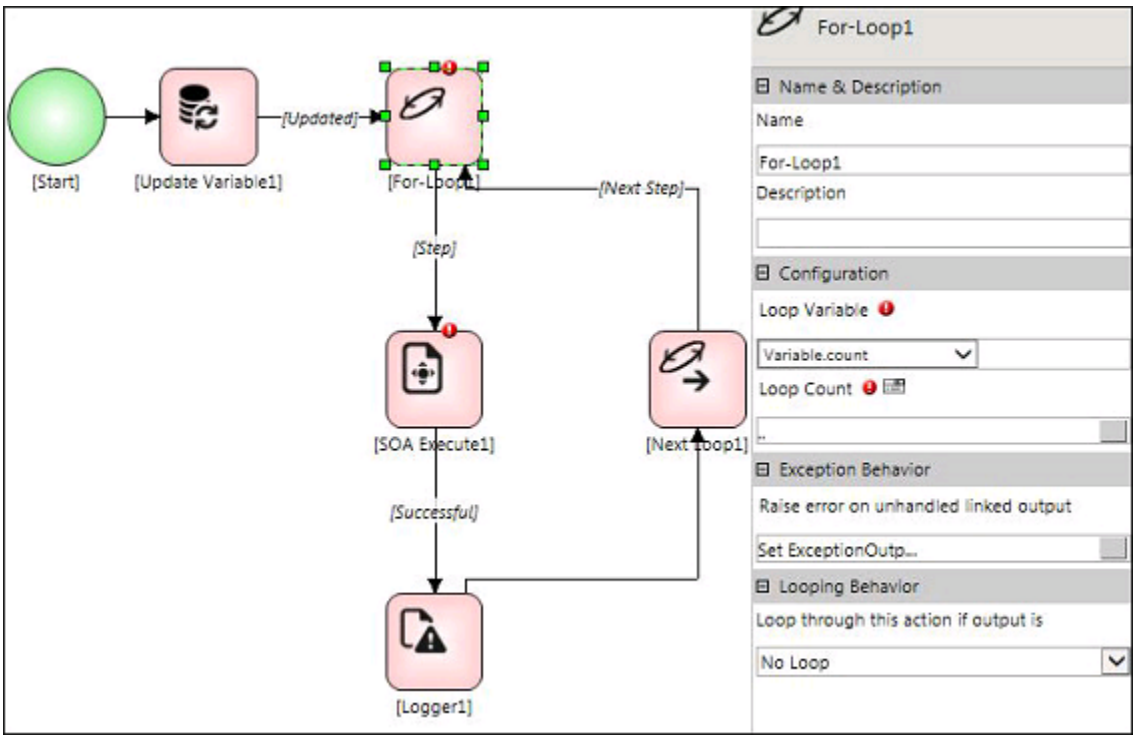
Sample Code

```
EmployeeCollection.GetEmployees().Count - 1; //Get the actual count of the items in the collection object. Subtracting one from it will give the actual loop count required.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

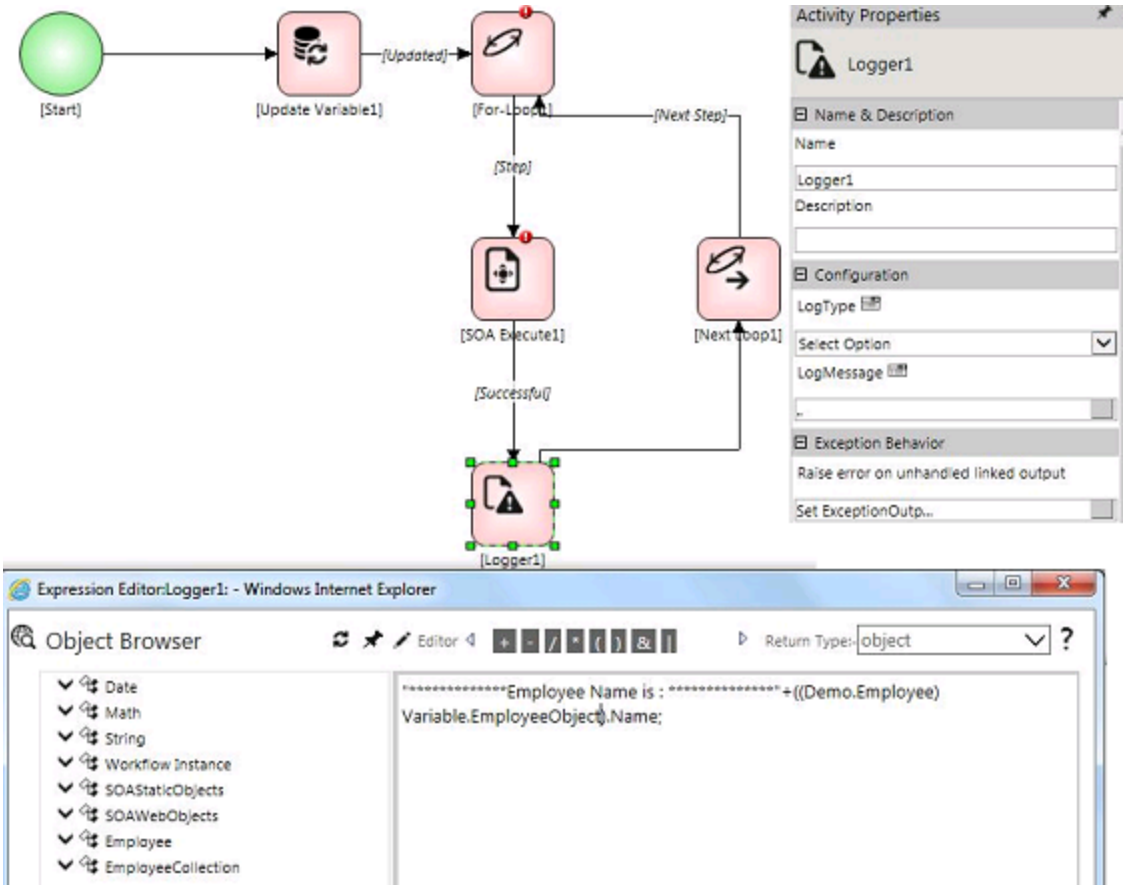


- Drag and drop the For-Loop activity from the Engine activities.



- Drag and drop the Logger activity next to the SOA Execute activity. This is done to display the employee information each time when the object is updated to the variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Sample code

```
((Demo.Employee)Variable.EmployeeObject).Name;
```

Variable EmployeeObject holds the employee information. In order to get the property of the object we need to type cast to the proper object type. i.e., the Employee object, Demo.Employee.

- Place the **Next Loop** activity to iterate through the next item in the collection.
- Execute the workflow.
- During execution, the workflow gets the EmployeeCollection object and iterates through the items, and prints the employee name in the logger.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using this, it is possible to persist complex objects in the workflow and iterate through the collection object to get an item from it.

How to Design a Workflow?

The below example shows how the user can design a AVEVA Work Tasks Workflow.

1. Drag and drop the 'SOA Execute' Activity next to the Start Activity to instantiate the 'Product' object. This will save the object in the mapped variable, so that it is possible to obtain the object from the variable the next

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

time.

Expression for SOA Execute Action: [Instantiate Product Object]

```
//Variable to hold the total quantity
Variable.TotalQuantity = 10;
//Product code
Product.Code;
```

Note: Invoking of the class is not always required. This is to show that the object state gets stored in the variable whenever you invoke the object for the first time.

2. Use 'SOA Execute' to calculate the total amount to be paid for the given quantity.

   Expression for SOA Execute Activity, GetAmount:

```
 //Total amount to be paid is calculated and stored in the 'TotalAmount' variable.
 Variable.TotalAmount =
 Convert.ToInt32(OrderLine.GetAmount((Product.Product)Variable.ProductObject,
 Convert.ToInt32(Variable.TotalQuantity)));
 // 'Variable.ProductObject' is the mapped variable for the 'Product' instance. When the
product object is instantiated for the first time, it automatically stores the object in
the variable. So, in this step we can directly get the 'Product' object from the
variable. We need to explicitly type cast the object to its appropriate
type. (Product.Product)Variable.ProductObject
```

ArchestrA Events

This section provides information on the ArchestrA Wait for Event activity.

ArchestrA Wait for Event

Activity Description

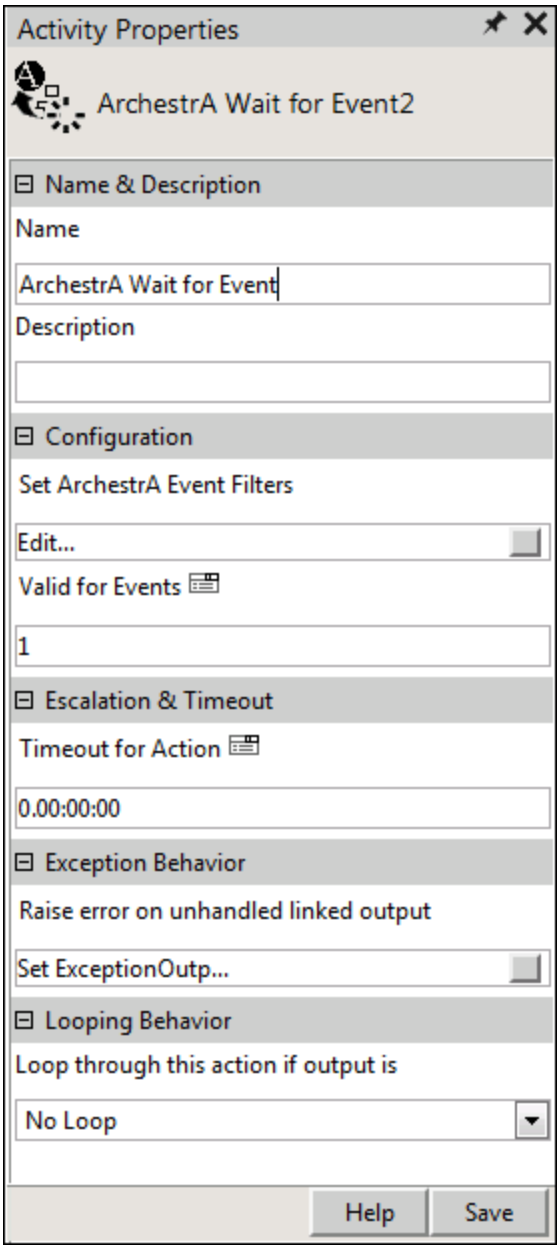
Use the ArchestrA Wait for Event activity to wait for a number of events from ArchestrA Event Services platform for a particular time period, and then consume the events in a defined workflow. You can set the number of events and the time period in the properties of that activity. It can store the event context details in the specified XML variables.

Activity Properties

Enter appropriate values in the **Activity Properties** area to configure the ArchestrA Wait for Event activity. Click the **Activity Properties** icon in the tool bar or select the appropriate option from the right-click menu for the activity.

The following image shows the **Activity Properties** window of the ArchestrA Wait for Event activity:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Name & Description

Use the name and description properties to specify the name and description for the activity.

- **Name**: Use this property to enter a name for the activity.
  Property Type: Mandatory.
- **Description**: Use this property to enter a brief description to be displayed for the activity.
  Property Type: Optional. Set this property only if required.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration

Use these properties to configure the functionality of the activity.

- **Set ArchestrA Event Filters**: Use this property to filter the ArchestrA events. Whenever an event occurs, it will check this filter, and if a match is found, the activity will be alerted.

  Property Type: Mandatory

  For more information, see Set ArchestrA Event Filters.

- **Valid for Events**: Use this property to specify the number of events for which the activity should wait. It indicates that the activity should wait till the specified number of events are received.

  Default Value: The default value is 0 which means that the Archestra Wait for Events activity waits for infinite occurrences of the events, and the workflow does not proceed to the next step in the flow. However, the workflow ends only if there is an exception in the workflow.

  Property Type: Optional. Set this property only if required. If not set, the default value is considered.

Escalation & Timeout

Use this property to specify escalation and timeout settings for the activity.

- **Timeout for Action**: Use this property to set timeouts for this activity. If specified, the activity will be timed out after the specified time span.

  Default Value: 0

  Format: d.hh:mm:ss, where "d" denotes the number of days the schedule is set for, and the time setting denotes the number of hours, minutes, and seconds.

  Property Type: Optional. Set this property only if required.

Exception Behavior

Each activity has the **Raise error on unhandled linked output** property. Click on this to see the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Looping Behavior

Use this property to specify the looping behavior of the activity.

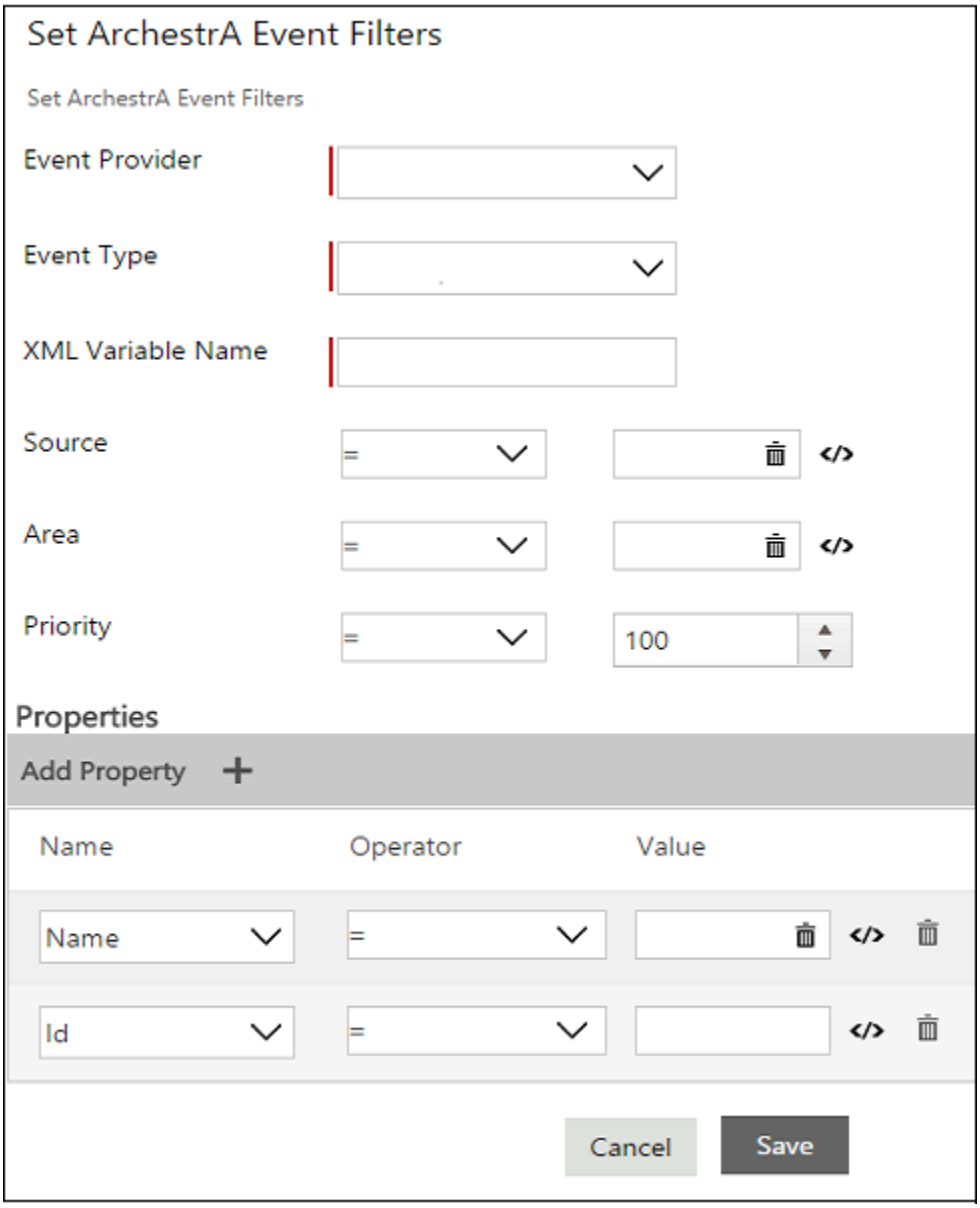
- **Loop through this action if output is**: Use this property to specify the condition for looping the activity by selecting an activity output from the drop down. Available options are as follows:
  - **No Loop**
  - **Action Completed**: This output is displayed when the activity has timed out.
  - **Event Received**: This output is displayed when an ArchestrA event matching the specified filter condition is received.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Set ArchestrA Event Filters

Use this property to filter the ArchestrA events. Whenever an event occurs, it will check this filter, and if a match is found, the activity will be alerted.

The following image shows the Set ArchestrA Event Filters properties window:



Enter the following details:

- **Event Provider**: Select the name of the event provider from which you want to poll for new events. The Event Provider field is mandatory.

  Based on the data available in the ArchestrA Event main list, the Event Provider list is populated. You can add other event providers to this main list, if required.

  Example of Event Providers are InBatch Server or Avantis.

  To go to the ArchestrA Event main list, click the **Menu** button, click **Tools**, and then click **ArchestrA Events**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The ArchestrA Events page appears which displays the available main list. For more information, see *Use the product > Enterprise Console > ArchestrA Events*.

- **Event Type**: Select the Event Type of the Event Provider.

  It is the type of event which defines the properties of the event.

  For InBatch Server, Batch is an event, and possible values for Batch events are such as, BatchId, BatchSize, CampaignId, EventCode, EventCodeName, and LotId.

  Based on the data available in the ArchestrA Event main list, the Event Type list is populated for the selected Event Provider. You can add other event types for the event provider, if required. You can view the properties of the event type in the Properties grid towards the lower section of the Set ArchestrA Event Filter window.

- **XML Variable Name**: Enter the XML variable name for storing event context. When an event occurs, this XML variable will be populated with event context data and it is used later in the flow of the process. The XML variable name must be unique for every ArchestrA Wait for Event activity. When you create an XML variable, the related schema for the XML variable is created automatically.

  If an ArchestrA Wait for Event activity is deleted from a workflow, the associated XML variable is retained. However, we recommend to not use the same XML variable for another ArchestrA Wait for Event activity as the XML variable properties gets overwritten with every workflow execution which will lead to unexpected behavior of the workflow. So, if there are multiple ArchestrA Wait for Event activity in a workflow, each activity will have their own XML variable.

- **Source**: Type the Source property name of the event provider. It is the name of the event provider sending the event. For example, for a Batch event, BatchMngr is the source.

  Select the appropriate operator from the operator drop-down list.

  Click the </> icon to define an expression in the Expression Editor.

  For more information about Expression Editor, see *User Guide > Process Designer > Setting Activity and Link Properties > Using the Expression Editor*.

  Note: If you have defined an expression, then the 🗑 icon is displayed. Click this icon if you want to remove the defined expression.

- **Area**: Type the Area of the event provider. Areas provide a key organizational role in grouping alarm information and assigning it to users who use alarm and event clients to monitor their Areas of responsibility.

  The Area object plays a key role in alarm and event distribution. All objects belong to an Area. Areas can contain sub-Areas. Alarm and event clients are configured to subscribe to a set of Areas.

  Click the </> icon to define an expression in the Expression Editor.

  For more information about Expression Editor, see *User Guide> Process Designer > Setting Activity and Link Properties > Using the Expression Editor*.

  Note: If you have defined an expression, then the 🗑 icon is displayed. Click this icon if you want to remove the defined expression.

- **Priority**: Type the priority for each designated event provider.

  Priority, which is an integer value from 1 to 999 indicating the severity of the alarm which raises an event. An alarm priority of 1 is most urgent and 999 least urgent.

- **Properties**: Based on the Event Type that you enter, select values in the Properties section. For example, for

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

a Batch event type, BatchServer is a property of the data type String. The Properties field is optional.

Based on the data available in the ArchestrA Event main list, the properties of the Event Type list is populated. You can add other properties of the event type, if required.

Click the </> icon corresponding to a property, to define an expression for that property in the Expression Editor.

For more information about Expression Editor, see *User Guide > Process Designer > Setting Activity and Link Properties > Using the Expression Editor*.

Note: If you have defined an expression, then the 🗑 icon is displayed. Click this icon if you want to remove the defined expression.

Limitations:

- The *Between* operator is not supported for **Priority** and **Properties**.
- The data type of the operators are not validated. Ensure to provide valid values to the property data type.
- While creating a filter condition, duplicate properties are not supported.
- The OR condition is not supported while creating a filter condition.

Message Queue Activities

This section provides information on the SendMQData activity. This activity is used to publish messages to Message Queue.

SendMQData

Activity Description

Use SendMQData activity to publish message to Message Queue.

Activity Properties

To use SendMQData activity, follow the given steps:

1. In the **Process Designer** window, select **Message Queue** from the **Activities** drop-down list.
2. Drag and drop the **SendMQData** activity on the **Process Designer** window.
3. Right-click the **SendMQData** activity and then select **Activity Properties**.

Name & Description

Use the name and description properties to specify the name and description for the activity.

- **Name**: Use this property to enter a name for the activity.

  Property Type: Mandatory.
- **Description**: Use this property to enter a brief description to be displayed for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional. Set this property only if required.

Configuration

Use these properties to configure the functionality of the activity.

- **Message Queue Connection**: Use this property to configure Message Queue connection. For more information, see Configuring Message Queue Connection.

  Property Type: Mandatory

Exception Behavior

Each activity has the **Raise Error on Unhandled Linked Output** property, on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down. The looping option for this activity is Completed.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The SendMQData activity returns the following values:

Successful: Displayed when the activity publishes the message to Message Queue.

Unsuccessful: Displayed when the activity fails to publish the message to Message Queue.

Configuring Message Queue Connection

Use Message Queue Connection Settings property to configure connections to Message Queue.

Based on the **Connection** type, **Message Name** and **Message Type** input fields are visible/invisible.

In case of **WM Object** connection type, these input fields are visible to the user, so that the user can pass this information. In case of other connection types, these fields are not needed, hence they are invisible.

1. Select the name of the Message Queue connection from the Connection list. Connections with Connection Type selected as Send would be listed in the list.
2. Select the Binding type for the Connection.

   The following fields appear based on the Binding Type selected for the Connection.

Binding Type for Connection	Field Name	Description
WM Object	Message Name	Enter the Message name or click

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Binding Type for Connection	Field Name	Description
		the Expression Editor Icon to define an expression in the Expression Editor.
	Message Type	Enter the Message type or click the Expression Editor Icon to define an expression in the Expression Editor.
	Message Details	Enter the message in string format.
XML	Message Details	Enter the message in XML format.
JSON	Message Details	Enter the message in JSON format.

For more information about Expression Editor, see *User Guide > Process Designer > Setting Activity and Link Properties > Using the Expression Editor*.

Note:
- It is recommended to use Expression Editor to enter or copy multi-line text in 'Message Details' of SendMQData Activity rather than directly copying into the input field.

- In Expression Editor the text has to be enclosed within double quotes " ".

WaitForMQData

Activity Description

Use the WaitForMQData activity to wait for a number of messages from Message Queue for a particular time period, and then consume the messages in a defined workflow. You can set the number of messages and the time period in the properties of that activity. It can store the message details in the specified XML variables.

Activity Properties

To use WaitForMQData Activity, follow the given steps:

1. In the **Process Designer** window, select **Message Queue** from the Activities drop-down list.
2. Drag and drop the **WaitForMQData** activity on the Process Designer window.
3. Right-click the **WaitForMQData** activity and then select Activity Properties.

Name & Description

Use the name and description properties to specify the name and description for the activity.

- **Name**: Use this property to enter a name for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Mandatory.

- **Description**: Use this property to enter a brief description to be displayed for the activity.

  Property Type: Optional. Set this property only if required.

Configuration

Use these properties to configure the functionality of the activity.

- **Set Message Queue Filters**: Use this property to filter the messages. Whenever a message is received, it will check this filter, and if a match is found, the activity will be alerted.

  Property Type: Mandatory

  For more information, see Set Message Queue Filters.
- **Valid for Events**: Use this property to specify the number of messages for which the activity should wait. It indicates that the activity should wait till the specified number of messages are received.

  Default Value: The default value is 1 which means that the WaitForMQData activity waits for one message. For infinite number of messages, enter 0 in the **Valid for Events** property.

  Property Type: Optional. Set this property only if required. If not set, the default value is considered.

Exception Behavior

Each activity has the **Raise error on unhandled linked output** property. Click on this to see the default configured mapped error outputs in red with their check boxes selected.

See Mapped Error Output for more details.

Looping Behavior

Use this property to specify the looping behavior of the activity.

- **Loop through this action if output is**: Use this property to specify the condition for looping the activity by selecting an activity output from the drop down. Available options are as follows:
  - **No Loop**
  - **Action Completed**: This output is displayed when the activity has timed out.
  - **Event Received**: This output is displayed when a Message Queue connection matching the specified filter condition is received.

Set Message Queue Filters

Use this property to filter the messages. Whenever a message occurs, it will check this filter, and if a match is found, the activity will be alerted.

1. Select the name of the Message Queue connection from the Connection list. Connections with Connection Type selected as Receive would be listed in the list.
2. Select the Binding type for the Connection.

   The following fields appear based on the Binding Type selected for the Connection.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Binding Type for Connection	Field Name	Description
WM Object	Message Type	Enter the message type that needs to be listened from Message Queue or Click the Expression Editor Icon to define an expression in the Expression Editor.
XML	XPath	Enter the XPath for the XML Message or Click the Expression Editor Icon to define an expression in the Expression Editor.
	XPath Value	Enter the XPath Value for the XML Message or Click the Expression Editor Icon to define an expression in the Expression Editor.
	XML Schema	Enter the XML Schema for the XML Message or Click the Expression Editor Icon to define an expression in the Expression Editor.
JSON	JSON Path	Enter the JSON Path for the JSON Message or Click the Expression Editor Icon to define an expression in the Expression Editor.
	JSON Path Value	Enter the JSON Path Value for the JSON Message or Click the Expression Editor Icon to define an expression in the Expression Editor.
	JSON Sample	Enter the sample for the JSON Message or Click the Expression Editor Icon to define an expression in the Expression Editor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For more information about Expression Editor, see *User Guide > Process Designer > Setting Activity and Link Properties > Using the Expression Editor*.

3. **XML Variable Name**: Enter the xml variable name.

   XML variable is created with this name and is populated with the data after the message is received

Logic App Activities

This section provides information on the Call Logic App activity. This activity is used to call Logic App Workflows.

Call Logic App Activity

Activity Description

Use Call Logic App activity to call Logic App Workflows.

Activity Properties

To use Call Logic App activity, follow the given steps:

1. In the **Process Designer** window, select **Call Logic App** activity.
2. Drag and drop the **Call Logic App** activity on the **Process Designer** window.
3. Right-click the **Call Logic App** activity and then select **Activity Properties**.

Name & Description

Use the name and description properties to specify the name and description for the activity.

- **Name**: Use this property to enter a name for the activity.

  Property Type: Mandatory.
- **Description**: Use this property to enter a brief description to be displayed for the activity.

  Property Type: Optional. Set this property only if required.

Configuration

Use these properties to configure the functionality of the activity.

- **Logic App Connection and Data**: Use this property to get connections to Logic Apps and pass data, if required.

  Property Type: Mandatory

Exception Behavior

Each activity has the **Raise Error on Unhandled Linked Output** property on click of which, it shows all the default

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

configured mapped error outputs in red with their check boxes selected

See Mapped Error Output for more details.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down. The looping option for this activity is Completed.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs

The Call Logic App activity returns the following values:

Successful: Displayed when the connection execution is successful.

Unsuccessful: Displayed when the connection execution is unsuccessful.

Configuring Logic App Connection and Data

Use Logic App Connection and Data property to configure connections to Logic Apps and pass data.

To configure Logic App connection and pass data

1. Create a Logic App Connection. For more information, see Logic App Connections.
2. In the **Logic App Connection and Data** window, select a **Logic App Connection** created earlier.
3. Enter a variable for **Variable Name**.

   This variable stores the Logic App Alert Key.
4. Get the custom JSON schema from the Logic App to pass relevant data.

   For example:

```
{
"type": "object",
"properties": {
"id": {
"type": "number"
},
"machine": {
"type": "string"
},
"status": {
"type": "boolean"
}
}
}
```

5. Click **Add Logic App Data** to add a row of data.
6. Enter **Name** and select **Data Type** obtained from the custom JSON schema, and then enter a **Value**.

   For example:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name: id

Data Type: Number

Value: 9876

Note:
- The JSON keys are case-sensitive.
- The supported data types are Boolean, Number, and String.
- Click **Expression Builder** to generate value from an expression, if required.
- Single level JSON structure is supported.

7. Repeat **Steps 4 to 5** to enter the required data.
8. Click **Save**, and then click **Close**.

Note:
- The data is not validated. Ensure name and data type are as specified in the custom JSON schema, and the value matches the data type.
-- In case of duplicate names, value passed is only for the first match.
-- In case of names from the standard JSON schema (e.g., ApplicationName), value passed will overwrite the default data.

Standard JSON Schema

Logic Apps must include the following standard JSON schema to interact with AVEVA Work Tasks.

```
{
"properties": {
"ApplicationName": {
"type": "string"
},
"ExecutionDetailsId": {
"type": "integer"
},
"ExecutionId": {
"type": "integer"
},
"LogicAppAlertKey": {
"type": "string"
},
"WorkflowName": {
"type": "string"
},
"UserName": {
"type": "string"
}
},
"type": "object"
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Wait for Logic App Alert Activity

Activity Description

Use Wait for Logic App Alert activity to wait for an alert from the Logic App based on the alert key filter.

Activity Properties

To use Wait for Logic App Alert activity, follow the given steps:

1. In the **Process Designer** window, select **Call Logic App** activity.
2. Drag and drop the **Wait for Logic App Alert** activity on the **Process Designer** window.
3. Right-click the **Wait for Logic App Alert** activity and then select **Activity Properties**.

Name & Description

Use the name and description properties to specify the name and description for the activity.

- **Name**: Use this property to enter a name for the activity.

  Property Type: Mandatory.
- **Description**: Use this property to enter a brief description to be displayed for the activity.

  Property Type: Optional. Set this property only if required.

Configuration

Use these properties to configure the functionality of the activity.

- **Logic App Alert Filter**: Use this property to set the alert key along with the Web API URL to post from the Logic App.

  Property Type: Mandatory

  In the Logic App Alter Filter window, configure these as follows:
  - **Enterprise Console**: Select an Enterprise Console site. The list shows Enterprise Console sites to which repositories are mapped, and for which Azure Service Bus (with Shared Access Signatures) are configured.
  - **URI**: The URI is automatically generated based on the selected Enterprise Console.
  - **Authorization Token**: The authorization token is automatically generated based on the selected Enterprise Console.
  - **Logic App Alert Key**: Enter a value for the alert key or click **Expression Builder** to generate value from an expression, if required.
- **Valid for Alert**: Use this property to set the number of alerts from the Logic App. If the value is set to **0**, then there is no limit for the number of alerts from the Logic App.

  Property Type: Mandatory
- **Logic App Output XML**: Use this property to set the XML variable to post from the Logic App.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional

In the Logic App Output XML window, configure these as follows:

- **Logic App JSON Data**: Enter the JSON schema to be passed from the Logic App.
- **Logic App XML Schema**: After entering the JSON schema for **Logic App JSON Data**, click **Get XML Schema** to auto generate the XML schema.
- **Output XML Variable Name** : Enter a name for the XML variable. Upon saving, this variable will hold the schema.

Exception Behavior

Each activity has the **Raise Error on Unhandled Linked Output** property on click of which, it shows all the default configured mapped error outputs in red with their check boxes selected

See Mapped Error Output for more details.

Looping Behavior

The property in the Looping Behavior group has been described below. You can use this property to specify the looping behavior of the activity.

Loop through this action if output is - This property is used to specify the condition for looping the activity by selecting an activity output from the drop down. The looping option for this activity is Completed.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs:

The Wait for Logic App Alert activity returns the following values:

- If the Valid for Alert property is set to/or is 0, then after receiving an alert from the Logic App, the output will be Alert Received, and the activity will be in sleep state.
- If the Valid for Alert property is set to/or is 1, then after receiving an alert from the Logic App, the output will both Alert Received and Action Completed, and the activity will be in completed state.
- If the Valid for Alert property is set to/or is more than 1, then after receiving an alert from the Logic App, the output will Alert Received, and the activity will be in sleep state.

Note: While posting alert from the Logic app, ensure the following are part of the HTTP request:
- **URI** as generated based on the selected Enterprise Console.
- **Authorization Token**, as generated based on the selected Enterprise Console, is in the header.
- **ApplicationName** and **LogicAppAlertKey** in JSON schema are in the body.

Design and Execute Workflows

This section shows how to design and execute a workflow in AVEVA Work Tasks.

- Create a New Workflow Process
- Execute the Workflow

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create a New Workflow Process

To create a new workflow process

1. Drag the **Process Designer** control from the toolbox onto the Web Form's Designer surface.
2. In the **Properties** area, enter the new application name in the **ApplicationName** property field.
3. Enter the new workflow name in the **WorkflowName** property field.
4. Browse to the Web Form in which the **Process Designer** control is embedded.
5. This displays the Process Designer screen.
6. In the Process Designer screen, add the content and variables to the **Start** activity of the workflow. These will be visible in the content and variables collection for the other activities.
7. Select the **Human Activities** option in the activities to open the corresponding group.
8. From the **Human Activities** group, click the activities to be added to the workflow process. The icon corresponding to the activity is displayed on the Design Mosaic. For example, you can click the **Approval** option. This will display the Approval icon in addition to the Start icon (where n is the Approval number).

   Note: By default the Process Designer screen shows the Start activity icon on the Design Mosaic.
9. Click the **Approval1** activity. Drag the connector line to the Start activity and click on it to link the two activities.
10. Select the **Approval1** activity and set the properties in the corresponding fields in the Properties area.
11. Click **Save** to save the newly entered properties.
12. For details on setting the properties for different activities, refer to the corresponding topic under Activities.
13. Repeat steps 6 to 10 to add more activities as needed.
14. Click **Publish** in the tool bar to save and deploy the newly created workflow.
15. Now click the **Start** button in the Web Form toolbar or press

    Note: By default the Process Designer screen shows the start activity icon on the Design Mosaic.

Execute the Workflow

A workflow can be executed asynchronously, or synchronously using the **Redirection Enabled Property** in Process Designer.

- Execute Workflows Asynchronously
- Use Redirection Enabled Property for Synchronous Execution

List of Internal Status Codes

This topic lists the different Internal Status codes assigned in AVEVA Work Tasks workflows. The status codes are listed in three groups corresponding to workflows, activities, and work items.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Status assigned to a Workflow

EP = Execution Pending
FE = Finished with Errors
FN = Finished successfully
FR = Failed and Retrying the workflow executed with error
EX = Executing
PA = Paused
AB = Aborted
SL = Awaiting

Runtime assigned to Activities

P = Execution Pending
E = Action Executing
S = Sleep
C = Execution Completed
Z = Next Action initiated
K = Action Killed
U = Action Paused

Status assigned to Work Items

AW = Awaiting for actor to act
OW = Ownership taken
CO = Completed
UN = Unallocated
CL = Closed
QR = Queue Removed
QT = Queue Timeout
QC = Queue Closed
AL = Allocated
RH = Release Hold
AC = Closed as min/max ownership count satisfied
CR = Created
WC = Closed as decision was taken based on properties
TO = Timeout and Work Item closed
ID = Of type Information in Deleted State
HO = Hold
IO = Of type Information

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

CS = Change Status and Save

PA = Paused

AB = Killed

Execute Workflows

The workflow can be executed from any application. The **Client** class executes the workflow. Select the **Execute ()** method of the **Client** object depending on whether the latest version of the workflow has to be executed or a specific version of the workflow has to be executed.

In this section, you will see how to execute a workflow. Workflows can be executed either Asynchronously or Synchronously. This section has descriptions of both execution methods.

You can execute a workflow using the Workflow.NET.Engine.Client API.

public Client(string ApplicationName, string WorkflowName)

Note: By default the Client.Execute method executes the latest version of the workflow. To execute a particular version, specify the corresponding Version number as shown in the following example. Example: public int Execute(int Version, object UserID, string Data)

Execute Workflows Asynchronously

1. Execute(object UserId,string Data)

   The workflow can be executed by passing the UseridString and the XML data to the Execute method.

   Example:

   ```
   string xmlcontent = "<Expense><Amount>2000</Amount></Expense>";
   Client objClient = new Client("MyRep", "MyWF");
   int ExecutionID =objClient.Execute("activedirectory::Bob ",xmlcontent);
   objClient.Close();
   ```

2. Execute(int version, object UserId,string Data)

   The version of the workflow can be specified in this method along with the UseridString and the XML data.

3. Execute(object UserId,string Data,VariablesCollection variable)

   The variables used in the workflow can be assigned values using this method. The values are passed using the variables collection object.

   Example:

   ```
   string xmlcontent = "<Expense><Amount>2000</Amount></Expense>";
   VariablesCollection varCol = new VariablesCollection();
   varCol.Add("varPrice", "Number",1);
   varCol.Add("varTotal", "Number",2);
   Client objClient = new Client("MyRep", "MyWF");int ExecutionID =
    objClient.Execute("activedirectory::Bob ", xmlcontent,varCol);
   objClient.Close();
   ```

4. Execute(string versionStamp, object UserId,string Data)

   The specified version of the workflow can be executed by passing the version of the workflow that is already existing. The version can be created by right clicking on the work flow in the workflowlist (in the enterprise Console) and select save as option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example:

```
string xmlcontent = "<Expense><Amount>2000</Amount></Expense>";
VariablesCollection varCol = new VariablesCollection();
varCol.Add("varPrice", "Number",1);
varCol.Add("varPrice", "Number",2);
Client objClient = new Client("MyRep", "MyWF");
int ExecutionID = objClient.Execute("2","activedirectory::Bob ", xmlcontent);
objClient.Close();
```

5. Execute(int version,object UserId,string Data,VariablesCollection variable)

   The variables used in the workflow can be assigned values using this method.

6. Execute(string versionStamp, int version,object UserId,string Data,VariablesCollection variable)

   The variables used in the workflow can be assigned values using this method. The values are passed using the variables collection object. VersionStamp parameter is the version of the workflow that is already created. Version of the workflow can be created by right clicking on the work flow in the workflowlist and select save as option.

   Example:

```
string xmlcontent = "<Expense><Amount>2000</Amount></Expense>";
VariablesCollection varCol = new VariablesCollection();
varCol.Add("varPrice", "Number",1);
varCol.Add("varPrice", "Number",2);
Client objClient = new Client("MyRep", "MyWF");
int ExecutionID = objClient.Execute("2","activedirectory::Bob ", xmlcontent, varCol);
objClient.Close();
```

Use Redirection Enabled Property for Synchronous Execution

Workflows can also be executed synchronously using the Redirection Enabled Property in Human Activities in Process Designer.

Example scenarios for synchronous execution

When a customer submits a support request online, the request is analyzed for product, severity and other criteria and the request number and estimated days for resolution are displayed to the customer. This requires the AVEVA Work Tasks engine to return values to the client. For a sample implementation, see the example below.

In a vendor invoice approval process, after approval by the Manager, the invoice status is changed to Awaiting entry in ERP. The Manager needs to be able to see the current status after submission of the approval without refreshing the page with the list of invoices. In this case, the Redirection Enabled property is used with the Information activity to display information on current invoice status to the manager.

In an online Customer Survey, survey questions are successively displayed based on responses to previous questions.

A company uses a different payroll application for each country in which it operates. When an employee submits a request for details of balances in the retirement funds, the system validates the employee code and based on country of employment forwards the request details to the payroll application for that country and redirects the employee to the employee account page in the payroll application for the employee's country.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Enable redirection

To enable redirection from activity 1 to activity 2:

- Set the "Redirection Enabled?" property in Activity 1 as Yes.
- In Activity 2 set the "Redirect URL" property.
- If the redirection needs to happen to the work item list of the user assigned to activity 1, enter the URL as http://<%ApplicationPath%>/Default/<%DefaultPage%>?<%workitemid%>.

Example:

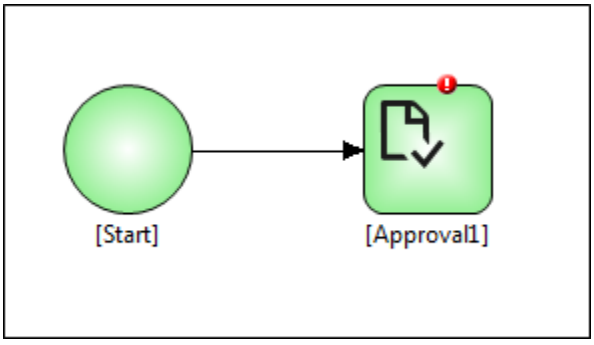
Redirecting from Client after Synchronous execution is performed (Engine sends a URL to client after the activity is executed)

Description

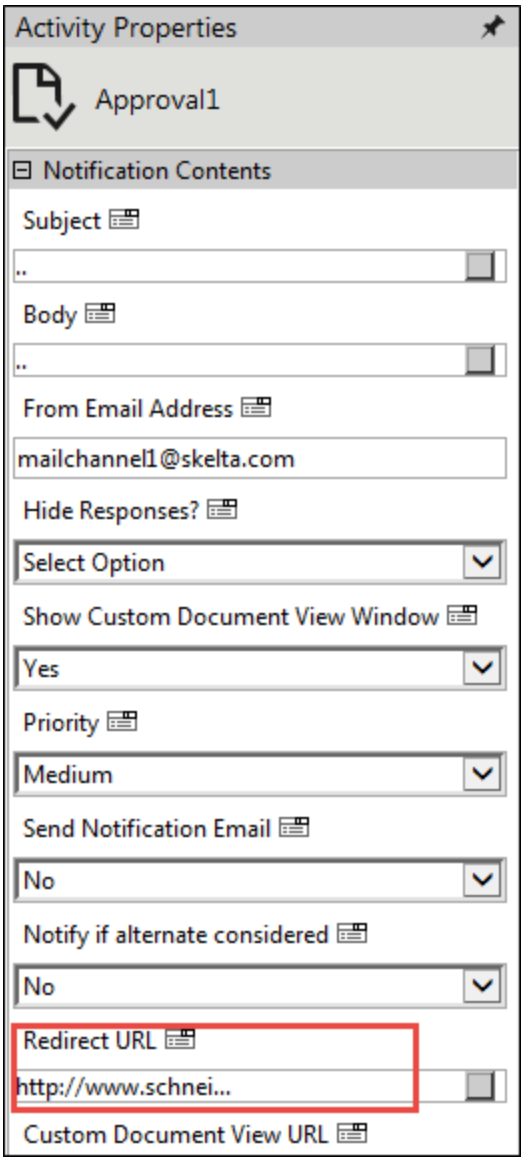
- The workflow is invoked from the code using ExecuteBlocking method.
- The workflow consists of an activity whose RedirectURL property is configured.

The client waits for the activity to execute, gets the url to be redirected from engine, and redirects to the specified link in the Redirection URL property of th activity.

Workflow Design



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Activity Properties

Redirect Url Property

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SampleCode

```
Workflow.NET.Engine.Client cl = new Workflow.NET.Engine.Client("Application", "Workflow");
Workflow.NET.Engine.SynchronousActionData syncData = new
Workflow.NET.Engine.SynchronousActionData(false, this.Context);
string url = "";
cl.ExecuteBlocking("activedirectory::bob", "<content></content>", syncData, true, out url);
cl.Close();
```

Enable Human-Workflow Interaction

This section focuses on:

- Human workflow services
- Using ASPX Form as the Detailed View Page of Work Item List Control
- Email and Microsoft Office Outlook Integration

Human Workflow Services

The **Channels** enable workflows to make available work items assigned to users through different media. Channels can be defined for specific media types, which are called Channel Types in AVEVA Work Tasks. Human Workflow Services (HWS) Activities in AVEVA Work Tasks announce their support for different Channels.

The **Forms** enable Actors to act on Work Items using one of the supported Channel Types. Forms can announce their support for multiple Channel Types.

About Channels

The **Channels** enable workflows to make available work items assigned to users through different media. Channels can be defined for specific media types, which are called **Channel Types** in AVEVA Work Tasks. Channel Types include **Web**, **Windows Forms**, **Email**, **PDA**, **Voice**, and **SMS** among others. Each Channel will support only one Channel Type.

Human Workflow Services (HWS) Activities in AVEVA Work Tasks announce their support for different Channels. For instance, some Activities will not support some of the Channels. Depending on the Channel selected, the Work Item corresponding to the Activity gets delivered to the Actor on the associated Channel Type. For example, if a Channel defined for the **Email Channel** Type is specified for an Activity, then the corresponding Work Item will get delivered to the Actor by email. The user can perform the necessary action by responding to the email appropriately. This is useful in situations where managers who have approval authority are away from their desk or are traveling. If more than one Channel is specified then the work item will get delivered on all the related Channel Types (by Email, on the PDA). This allows Actors to access and respond to Work Items from almost anywhere.

AVEVA Work Tasks provides the **Web Worklist Channel** and **Mail Channel** as the default channels. Developers can create their own channels using the **IChannelProvider** interface.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

About Forms

The **Forms** enable Actors to act on Work Items using one of the supported Channel Types. Forms can announce their support for multiple Channel Types. They are used to render the user interface for all the supported Channel Types, in which the Actor can act on the Work Item assigned to them.

The standard user interface – AVEVA Work Tasks Web Work Item List Form – for displaying Work Items supports the **Web Channel** Type. Other forms can be developed using the **IFormProvider** interface for different Channel Types.

Some of the forms that can be defined are custom ASPX forms that can be used to capture additional user inputs that the standard user interface cannot capture. These forms will correspond to the default Work Tasks Web Worklist Channel. Some of the forms that are possible for the AVEVA Work Tasks Mail Channel, which is the other default Channel, are HTML Forms, Plain Text Form, and Outlook Form.

Create Channels and Forms

New channels can be created using the Channel Provider. You can implement the methods of the IChannelProvider interface to create the custom channel. Similarly you can create new forms using the Form Provider. You need to implement the methods of the IFormProvider interface.

See Sample Implementation of Custom Work ItemForm for a sample implementation.

Use Channels and Forms

All Human Workflow Services (HWS) activities in AVEVA Work Tasks have a property named Set Channel. The user can specify the Channel and the associated form using this property. The property first lists all the Channels that are supported. Users can select the desired Channel from this list. Now the property lists all the Forms supported. You can select one or all Forms. If all Forms are selected then the activity will be delivered on all the Channel Types, which are associated with the different forms.

AVEVA Work Tasks allows Actors to personalize the Channels that are used to deliver Work Items to them by setting their Channel preferences using the Work Item List control (formerly called the Activity List). For example, an Actor who is traveling can choose to receive work items via email only. The Actor can turn off this setting on return.

Run-time Behavior of Channels and Forms

If a HWS Activity configured for Channels and Forms gets executed as part of a workflow process, the workflow engine delivers the corresponding Work Item on all the associated Forms. When the assigned Actor performs the necessary action in any one Form, the workflow engine announces this to all other Channels. These Channels will then take the appropriate compensation action(s). For example, if the Actor responds using the Skelta Web Work Item List Form, then the workflow engine announces this to the other Channels. When a Work Item has already been acted on by the Actor in one Form and does so again from another Form, the workflow engine can detect this redundant response and notify the Actor accordingly. It will display a message such as, "This is an invalid response as you have already taken the required action."

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

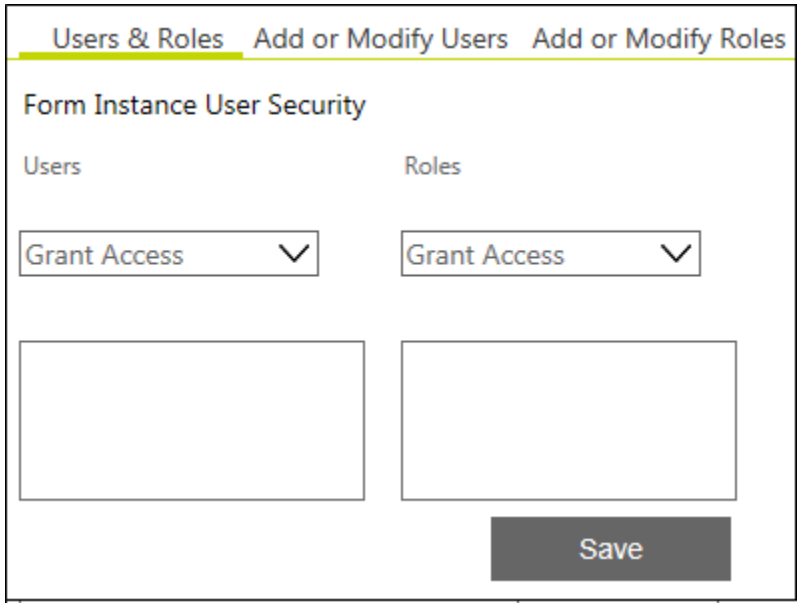
Set Access Rights for Form Instances

Access rights can be set for the individual instances of Forms. This is useful to restrict access to certain Forms to specific users or roles. For example, a leave application form can be accessed by all employees in an organization. However, a leave report form could be restricted to managers.

To set access rights

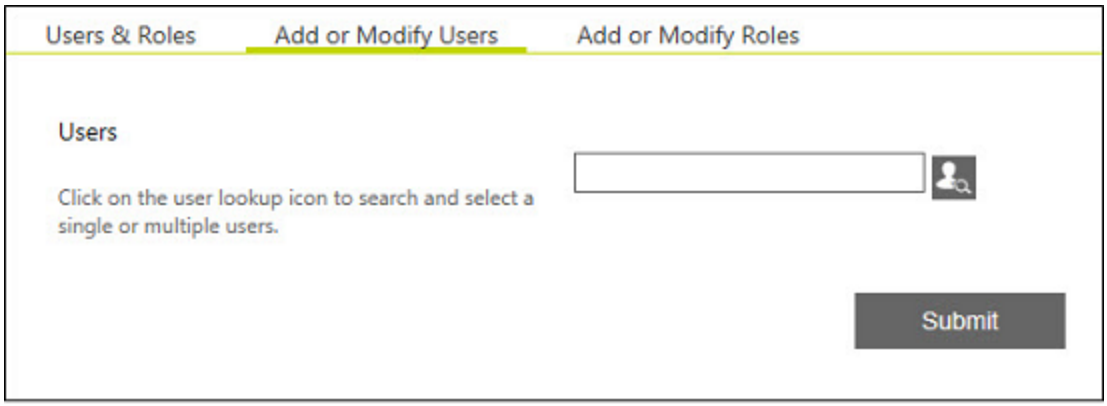
You can set access rights to Forms from the **Delivery Channel** property window. To set the property:

1. Right click the Form, as shown in the image below.
2. From the context menu, select **Set Access Rights** option.
3. This opens the blank Set Access Rights window shown below. This window has three tabs – **Users & Roles**, **Add or Modify Users**, **Add or Modify Roles** – which can be used to set access rights.
   - You can view the users and roles given access to the form from the view corresponding to the **Users & Roles** tab.
   - You can add users to the list of users permitted to access the form or modify this list from the view corresponding to the **Add or Modify Users** tab.
   - You can add users to the list of users permitted to access the form or modify this list from the view corresponding to the **Add or Modify Roles** tab.

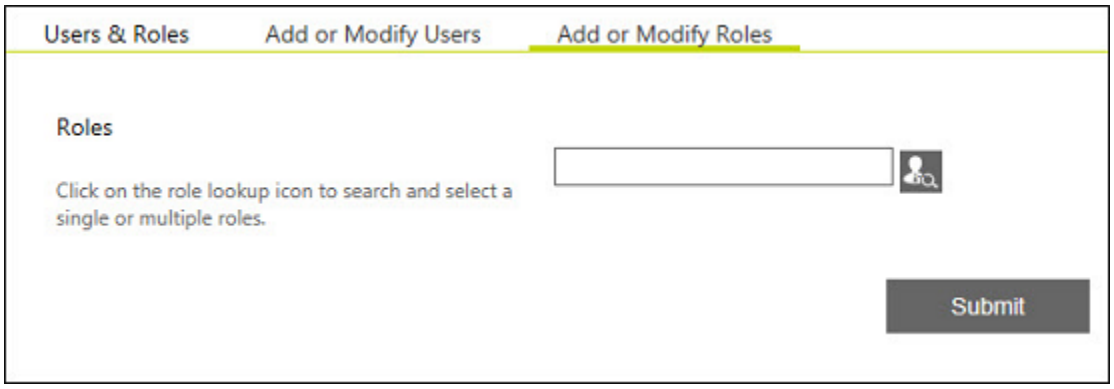


   The **Users & Roles** tab is selected by default. It has the **Users** list on the left and the **Roles** list on the right. Each list has a corresponding permissions drop down located above the list box. You can select Grant Access to give the users/roles in the list permission to access the form. Alternatively, you can select **Deny Access** to deny access to the users in the list. The **Save Users & Roles** button can be used to save the permission settings.
4. Click the **Add or Modify Users** tab to open the corresponding view, shown below before adding any resources.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

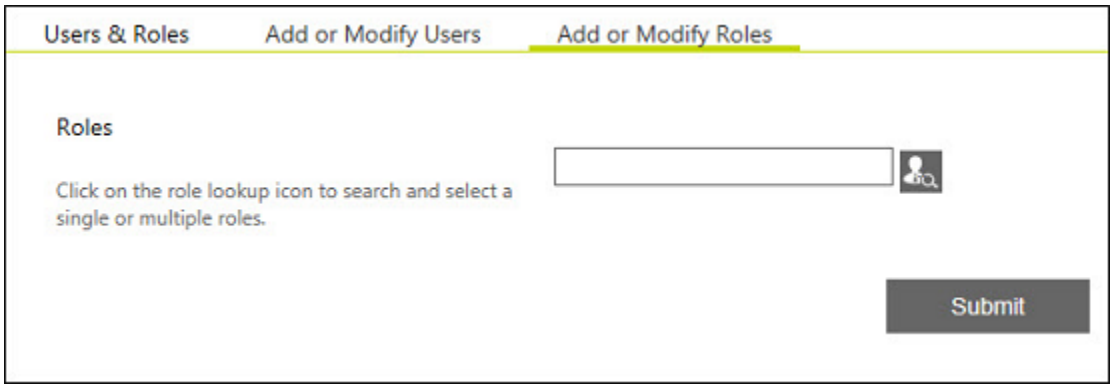


Click the **User Lookup** icon seen next to the **Select users** field. User Lookup screen appears. In this window, select the required provider from the provider drop-down. Enter a search pattern in the **Search For** field and specify the attribute in which to search for in the drop down provided. Click the **Search** button to search for the specified pattern. The resources matching the search pattern are listed in the **Available Resources** list. You can select a resource from this list and click the [>] button to move it to the **Selected Resources** list. You can move all the resources in the first list to the second by clicking the [>>] button. The window will appear as in the image shown below after resource selection. Click **Update**.



You can remove a resource from the **Selected Resources** list by selecting it and clicking on the [<] button. You can also remove all resources in this list by clicking on the [**<<**] button. After selecting the resources click on the **Submit** button to save the selection.

5. Click the **Add or Modify Roles** tab to open the corresponding view, shown below before adding any resources.

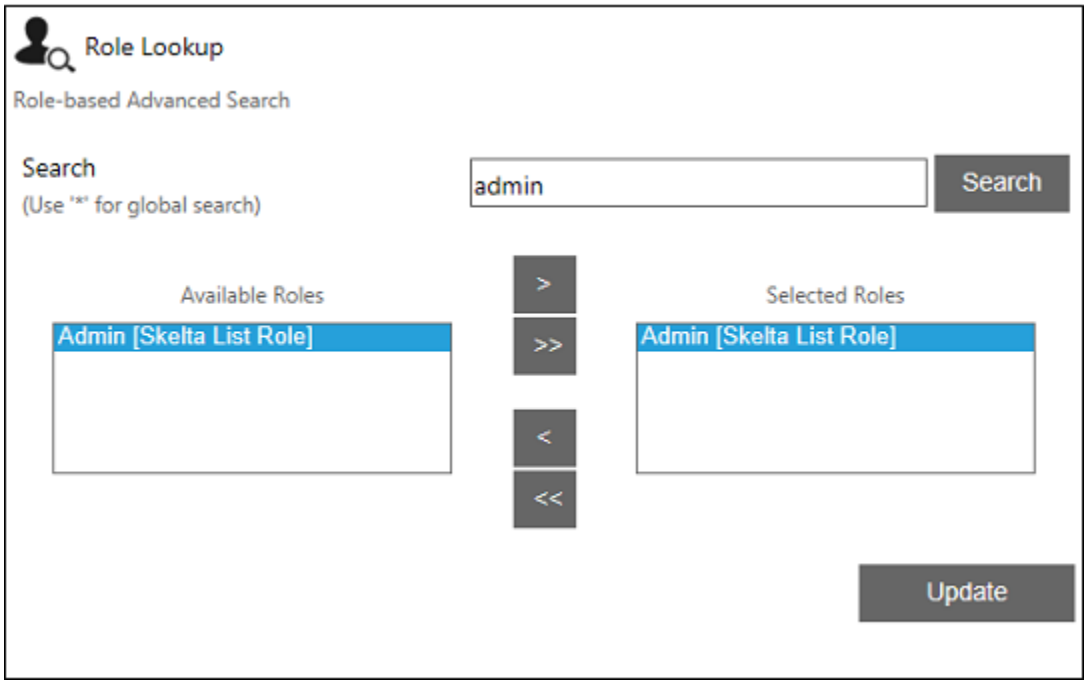


Click the **Role Lookup** icon seen next to the **Select roles** field. Role Lookup screen appears. In this window, select the required provider from the provider drop-down. Also, note that the role provider drop-down only

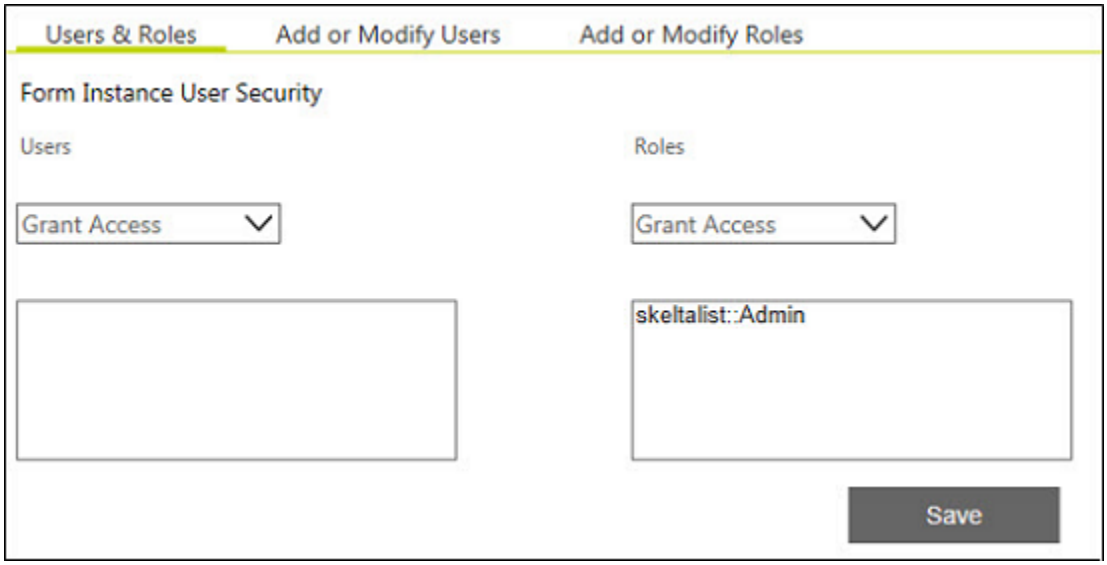
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

appears when the respository has more than one provider.

Enter a search pattern in the **Search** field. Click the **Search** button. The roles matching the search pattern are listed in the Available Roles list. You can select a role from this list and click the [>] button to move it to the **Selected Roles** list. You can move all the roles in the first list to the second by clicking the [**>>**] button. The window will appear as in the image shown below after the role selection. Click **Update** and then click **Submit**.



6. After adding the required users and roles, click the **Users & Roles** tab again to open the corresponding view, which will appear as shown below.



7. You can click **Save Users & Roles** to save the selection.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

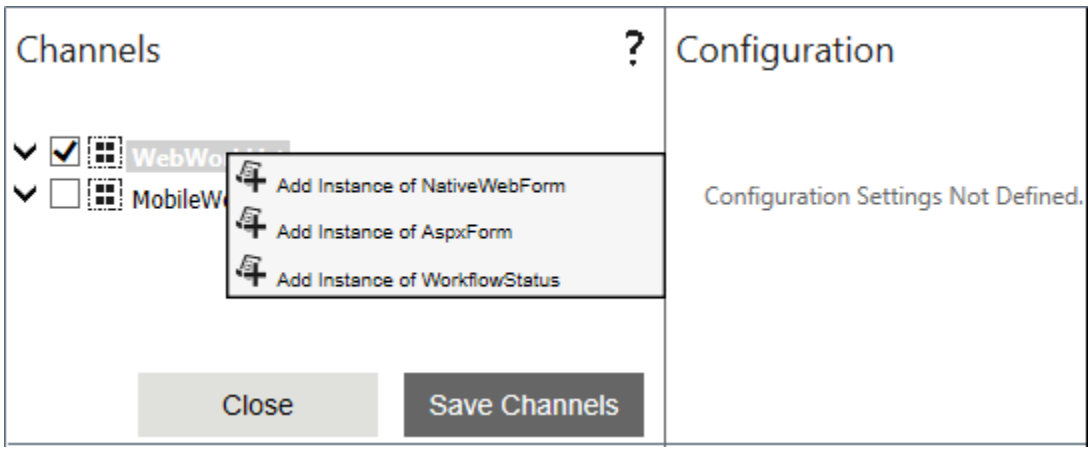
Use ASPX Form as the Detailed View Page of Work Item List Control

You can configure an ASPX form as the detailed view page of the Work Item List Control.

Complete the following steps to configure the ASPX form as the detailed view page:

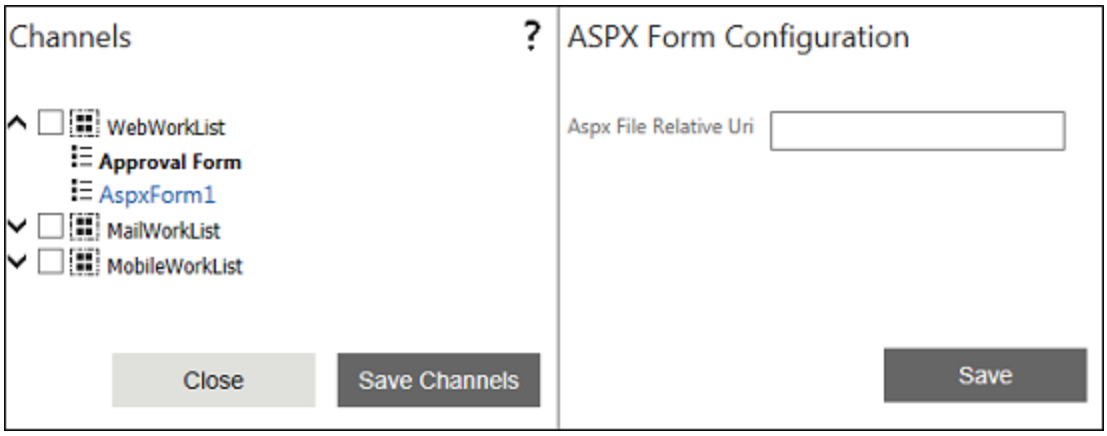
1. Select the human activity.
2. Click the **Delivery Channels** property.
3. The **Channels** dialog box appears as shown in the following image:

4. Right-click **WebWorkList** to display the context menu with the **Add Instance of AspxForm** option as shown in the following image:

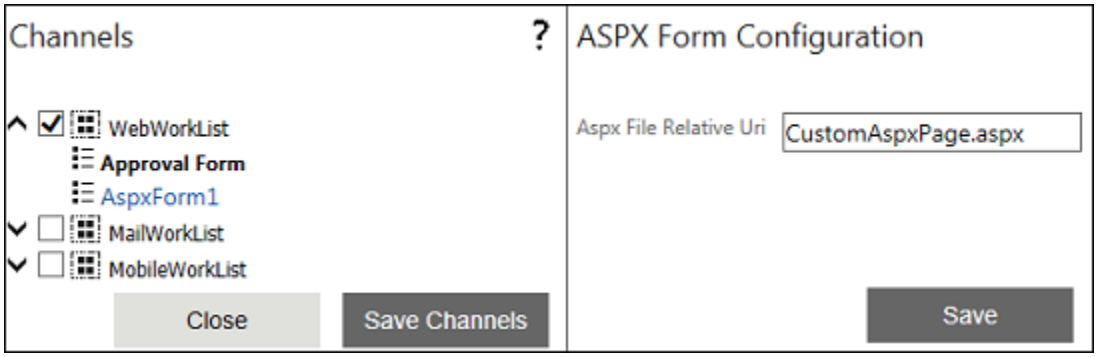


5. Click **Add Instance of AspxForm**.
   The **ASPX Form Configuration** view appears on the left side of the Channels dialog box.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



6. Specify the URL of the ASPX form as shown in the following image:

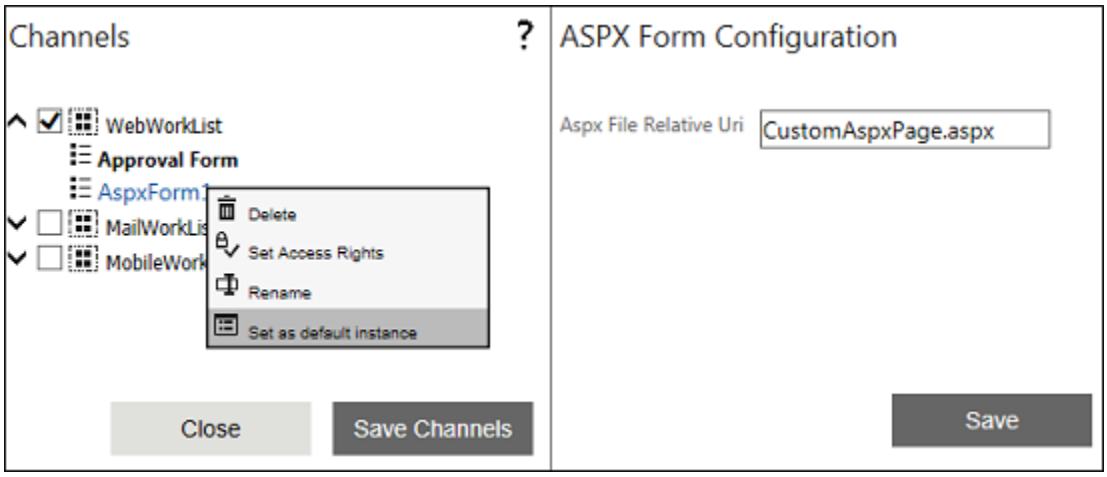


You can specify both relative and absolute URL in the text box.

If you are accessing the Work Item List Control from the Enterprise Console site, then the CustomAspxPageDefault.aspx page must be available in the BPMUITemplates\Default\Repository\Site folder.

Note: Ensure that the name of the custom form does not have the special characters, double-quotes and semicolon.

7. Right-Click *Aspxform1*. Choose **Set as default instance** to set the Aspxform1 as the default instance.



8. Click the **Save** and **Save Channels** buttons. Now, when you click any row in the Work Item List, you can see the aspx page being displayed as the detailed view.

Note: You must save the URL to rename the custom form. If you do not see the changes in the renamed

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

custom form, then close and reopen the window.

Email and Microsoft Office Outlook Integration

The **Mail Channel** allows Human Activities like Approval and Checklist to be sent by email to the intended recipient(s) who can perform the necessary action from the Inbox of email clients like Microsoft Office Outlook. The Human Activities have to be configured to enable this functionality.

Introduction to Sending Activities by Email

The Mail Channel allows Human Activities like Approval and Checklist to be sent by email to the intended recipient(s) who can perform the necessary action from the Inbox of email clients like Microsoft Office Outlook. The Human Activities have to be configured to enable this functionality. The topics in this section describe the steps required to make these configurations.

Configuring the Activities

In the Activity Properties section of the Process Designer, click on the Configuration node to expand it. This node has the **Delivery Channels** property, which you can use to set the medium through which the activity will be delivered to the recipient(s). You can select "MailWorkList" or "WebWorkList" or both from the pop up window for this property.

- If MailWorkList is selected the Activity is sent only to the recipient's email Inbox. In this case, the actor can view the work item and respond as required without leaving their email Inbox.
- If WebWorkList is selected the Activity is sent only to the recipient's Work items list Inbox. This means that the actor will have to open the Work Item List to view the activity and respond.
- If both channels are selected the activity will be sent to the Work items list as well as by email. In this case, the actor can respond to the work item email delivered to the email Inbox. They will also be able to view and respond to work items from the Work items list.

Action Response from Email clients

There are three methods in which the actors can respond to the work item assigned to them through the email channel:

1. Offline
2. Online
3. Microsoft Office Outlook

These methods are described below.

Offline: This method is useful when users are not directly connected to the Internet. For example, when you are using a laptop computer. In this method, actors can type in their comments and their choice of response within the square brackets provided against the respective field. For instance, if an Approval activity is received in the email, and you have chosen to approve, then you can reply to the email with your response. In your reply you can type your comments against the Comments field and "Approve" against the Action field. The response will

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

be sent when the email client next sends out the emails.

Online: This method can be used when users are connected to the Internet. In the body of the email there is an online section, where users can type in their comment and specify their response using the controls provided. For example, there will be two buttons - Approve and Reject - in the Approval Activity email. You can click on the preferred button. In the case of Checklist emails, after specifying the values, you can click on the Submit button that is available to submit your response. In both cases the response will be sent immediately.

Microsoft Office Outlook: This method is exclusively for users having Microsoft Office Outlook as their email client. In this method you can right click on the work item email in the Inbox and select the appropriate response from the context menu. For example, if it is an Approval activity then the context menu will show two options - Approve and Reject - from which the user can select one. This feature can be enabled in Microsoft Office Outlook by installing a utility that is provided along with AVEVA Work Tasks.

Configure Activities to be Sent by Email

To configure the Email Channel for Human Activities

In the Activity Properties section for Human Activities, carry out the following steps:

1. Click the Configuration node to expand it. This node has the delivery channel configuration property among others.
2. In this property group, click the button for the **Delivery Channels** property to open the corresponding window.
3. In this window, select the "MailWorkItem" channel by clicking the corresponding checklist, and click the Save Channels button.
4. After configuring all other Approval properties, click the **Save** button to save the settings. Now when the workflow is deployed and the activity is executed the corresponding work item will be sent to the recipient's email Inbox.

- See Approval - Delivery Channels for a description of the Delivery Channel property window for the Approval Activity
- See CheckList - Delivery Channels for a description of the Delivery Channel property window for the CheckList Activity
- See Choice - Delivery Channels for a description of the Delivery Channel property window for the Choice Activity
- See Info - Delivery Channels for a description of the Delivery Channel property window for the Information Activity
- See ResourceSet - Delivery Channels for a description of the Delivery Channel property window for the ResourceSet Activity
- See Task - Delivery Channels for a description of the Delivery Channel property window for the Task Activity

Date Format in Email Template

The date format of the **Submitted On** field in the Mail Channel email will be based on the actor's culture instead of the culture configured on the notification engine server. For example, if the culture set for the actor is English (United Kingdom) then the date format will be DDMMYYYY.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Integration with Microsoft Office Outlook

Activities configured to be sent by the mail channel allow users to act on these activities from Microsoft Office Outlook Inbox. This requires that the user machine has the **AVEVA Work Tasks Add-in for Microsoft Office Outlook** installed. See the *Administrator Guide* for information about installing this add-in.

Respond to Approval Activities from Email

Approval or Rejection from Email

When the Approval Activity in the workflow is executed it is received in the recipients' email Inbox.

You can respond to the assigned action using one of the following methods:

- Without using AVEVA Work Tasks Add-in for Microsoft Office Outlook
- Using AVEVA Work Tasks Add-in for Microsoft Office Outlook

Without using AVEVA Work Tasks Add-in for Microsoft Office Outlook:

There are two ways to act on the task:

- Acting on the task by Email
- Acting on the task through web-based

To act on the task by Email

In the body of the email carry out the following steps:

1. Click **Reply** to reply to the email.
2. Type in your comments in the square brackets provided for User Input against the Comments field.
3. Type in your choice of response within the square brackets provided against the Action field. If you choose to approve the request, type "Approve" here. If you do not wish to approve, type "Reject".
4. After specifying you response in the email, click **Send** to send it.

To act on the task through web-based

Pre-requisites:

- To access the Work items, you must be connected to the application server through LAN or WAN or Internet. If you are not connected to the application server, then the Work items list will not be accessible.
- Configure repository database to view work items from email before clicking View Work Item and act on it, or else the user will not be able to access the Work items list page.

Steps:

1. Click the **View Work Item and act on it** link. The Work items list page is displayed.
2. Click the work item and then act on that task.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To use AVEVA Work Tasks Add-in for Microsoft Office Outlook

Steps:

1. Right-click the Approval email in the Microsoft Office Outlook Inbox to display the menu. Alternatively, you can right-click the preview pane showing the email to view the menu.
2. Go to the **AVEVA Work Tasks** option in either of the menus and select "Approve" if you wish to approve. Select "Reject" if you wish to reject the request. This will submit the response immediately.

Document View:

When you click the **Document View** link, the document view with execution details is displayed.

Current Execution Status:

An actor will be able to access the Process Execution View and work items from an Email which is sent to the actor by the respective human activity.

Pre-requisites:

- To access the **Process Execution View**, the user has to be connected to the application server via LAN or WAN or Internet. In case, the user is not connected to the application server, then the **Process Execution View** will not be accessible.
- Configure the repository database for viewing the **Process Execution View** window before clicking **Current Execution Status** or else the user will not be able to access the Process Execution View window.

When you click the **Current Execution Status** link, the **Process Execution View** of the **Analysts View** is displayed.

Respond to Checklist Activities from Email

Checklist from Email

When the Checklist Action in the workflow is executed it is received in the recipients' email inbox.

You can respond to the assigned action using one of the following methods:

- Without using AVEVA Work Tasks Add-in for Microsoft Office Outlook
- Using AVEVA Work Tasks Add-in for Microsoft Office Outlook

Without using AVEVA Work Tasks Add-in for Microsoft Office Outlook

There are two ways to act on the task:

- Acting on the task by Email
- Acting on the task through web based

To act on the task by Email

In the body of the email carry out the following steps:

1. Click **Reply** to reply to the email.
2. Type in your inputs in the square brackets provided for User Input against the different checklist fields.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. After specifying you inputs in the email, click **Send** to send it.

To act on the task through web based

Pre-requisites:

- To access the **Work items**, the user has to be connected to the application server through LAN or WAN or Internet. In case the user is not connected to the application server, then the **Work items list** will not be accessible.
- Configure the repository database for viewing work items from email, before clicking **View Work Item and act on it** or else the user will not be able to access the Work items list page.

Steps:

1. Click the **View Work Item and act on it** link. The Work items list page is displayed.
2. Click the work item and then act on that task.

To use AVEVA Work Tasks Add-in for Microsoft Office Outlook:

Steps:

- Right-click the Check-List email in the Microsoft Office Outlook Inbox to display the menu. Alternatively, you can right-click the preview pane showing the email to view the menu shown below.

Document View:

When you click the **Document View** link, the document view with execution details is displayed as shown below.

Current Execution Status:

An actor will be able to access the **Process Execution View** and work items from an Email which is sent to the actor by the respective human activity.

Pre-requisites:

- To access the **Process Execution View**, the user has to be connected to the application server via LAN or WAN or Internet. In case the user is not connected to the application server, then the **Process Execution View** will not be accessible.
- Configure repository database for viewing **Process Execution View** window before clicking **Current Execution Status** or else the user will not be able to access the Process Execution View window.

When you click the **Current Execution Status** link, the **Process Execution View** of the **Analysts View** is displayed.

Respond to Choice Activities from Email

Choice from Email

When the Choice Activity in the workflow is executed it is received in the recipients' email inbox.

You can respond to the assigned action using one of the following methods:

- Without using AVEVA Work Tasks Add-in for Microsoft Office Outlook
- Using AVEVA Work Tasks Add-in for Microsoft Office Outlook

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Without using AVEVA Work Tasks Add-in for Microsoft Office Outlook

There are two ways to act on the task:

- Acting on the task by Email
- Acting on the task through web based

To act on the task by Email

In the body of the email carry out the following steps:

1. Click **Reply** to reply to the email.
2. Type in your comments in the square brackets provided for User Input against the Comments field.
3. Next, type in your choice of response within the square brackets provided against the Action field. If you choose to approve the request, type "Approve" here. If you do not wish to approve, type "Reject".
4. After specifying you response in the email click **Send** to send it.

To Act on the task through web based

Pre-requisites:

- To access the **Work items**, the user has to be connected to the application server through LAN or WAN or Internet. In case the user is not connected to the application server, then the **Work items list** will not be accessible.
- Configure repository database for viewing work items from email, before clicking View Work Item and act on it or else the user will not be able to access the Work items list page.

Steps:

1. Click the **View Work Item and act on it** link. The Work items list page is displayed.
2. Click the work item and then act on that task.

To use AVEVA Work Tasks Add-in for Microsoft Office Outlook:

Steps:

1. Right-click the choice email in the Microsoft Office Outlook Inbox to display the menu. Alternatively, you can right-click the preview pane showing the email to view the menu.
2. Go to the **AVEVA Work Tasks** option in either of the menus and select "Approve" if you wish to approve. Select "Reject" if you wish to reject the request. This will submit the response immediately.

Document View:

When you click the **Document View** link, the document view with execution details is displayed as shown below.

Current Execution Status:

An actor will be able to access the **Process Execution View** and work items from an Email which is sent to the actor by the respective human activity.

Pre-requisites:

- To access the **Process Execution View**, the user has to be connected to the application server via LAN or

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

WAN or Internet. In case the user is not connected to the application server, then the **Process Execution View** will not be accessible.

- Configure the repository database for viewing **Process Execution View** window before clicking Current Execution Status or else the user will not be able to access the Process Execution View window.

When you click the **Current Execution Status** link, the **Process Execution View** of the **Analysts View** is displayed.

Choice Response:

When you submit your response to the Choice email using any of the above methods, you will receive a confirmation email.

View Information Activities from Email

Information from Email

When the Information Activity in the workflow is executed it is received in the recipients' email inbox.

Work Item View

When you click the **View Work Item and act on it** link, the work item is displayed.

Document View

When you click the **Doc View** link the document view with execution details is displayed.

Current Execution Status

An actor will be able to access the **Process Execution View** and work items from an Email which is sent to the actor by the respective human activity.

Pre-requisites

- To access the **Process Execution View**, the user has to be connected to the application server via LAN or WAN or Internet. In case, the user is not connected to the application server, then the **Process Execution View** will not be accessible.
- Configure the repository database for viewing **Process Execution View** window before clicking Current Execution Status or else user will not be able to access the Process Execution View window.

When you click the **Current Execution Status** link, the **Process Execution View** of the **Analysts View** is displayed.

Respond to Task Activities from Email

Task from Email

When the Task Activity in the workflow is executed it is received in the recipients' email inbox.

You can respond to the assigned action using one of the following methods:

- Without using AVEVA Work Tasks Add-in for Microsoft Office Outlook
- Using AVEVA Work Tasks Add-in for Microsoft Office Outlook

Without using AVEVA Work Tasks Add-in for Microsoft Office Outlook

There are two ways to act on the task:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Acting on the task by Email
- Acting on the task through web based

To act on the task by Email

In the body of the email carry out the following steps:

1. Click **Reply** to reply to the email.
2. Type in your comments in the square brackets provided for User Input against the Comments field.
3. Next, type in your choice of response within the square brackets provided against the Action field. If you choose to approve the request, type "Approve" here. If you do not wish to approve, type "Reject".
4. After specifying you response in the email click Send to send it.

To act on the task through web based

Pre-requisites

- To access the **Work items**, the user has to be connected to the application server through LAN or WAN or Internet. In case the user is not connected to the application server, then the **Work items list** will not be accessible.
- Configure the repository database for viewing work items from email, before clicking **View Work Item and act on it** or else the user will not be able to access the Work items list page.

Steps:

1. Click the **View Work Item and act on it** link. The Work items list page is displayed.
2. Click the work item and then act on that task.

To use AVEVA Work Tasks Add-in for Microsoft Office Outlook:

Steps:

1. Right-click the choice email in the Microsoft Office Outlook Inbox to display the menu. Alternatively, you can right-click the preview pane showing the email to view the menu.
2. Go to the **AVEVA Work Tasks** option in either of the menus and select "Approve" if you wish to approve. Select "Reject" if you wish to reject the request. This will submit the response immediately.

Document View:

When you click the **Document View** link the document view with execution details is displayed.

Current Execution Status:

An actor will be able to access the **Process Execution View** and their work items from an Email which is sent to the actor by the respective human activity.

Pre-requisites:

- To access the **Process Execution View**, the user has to be connected to the application server through LAN or WAN or Internet. In case the user is not connected to the application server, then the **Process Execution View** will not be accessible.
- Configure the repository database for viewing **Process Execution View** window before clicking **Current**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Execution Status or else the user will not be able to access the Process Execution View window.

When you click the **Current Execution Status** link, the **Process Execution View** of the **Analysts View** is displayed.

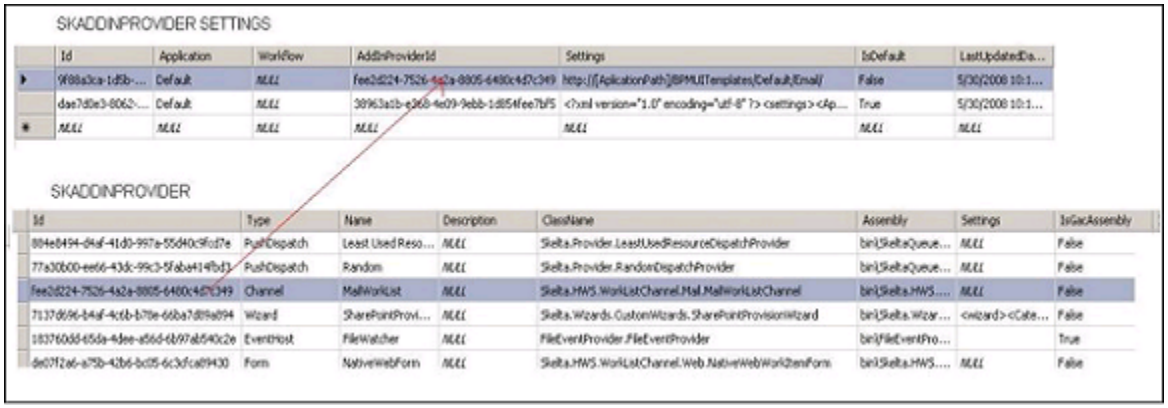
Task Response:

When you submit your response to the Task email using any of the above methods you will receive a confirmation email.

Configure Repository Database to View Process Execution View, Document View, and Work Items View from Email

Steps:

1. Go to Repository Database.
2. Open **SKAddInProviderSettings** table.
3. Set **Application** and **Workflow** columns if the **AddinProviderID** in **SKAddInProviderSettings** table should be the ID present in **SKAddInProviders** table whose **Type** column data is *Channel* and **Name** column data is *MailWorkList*.

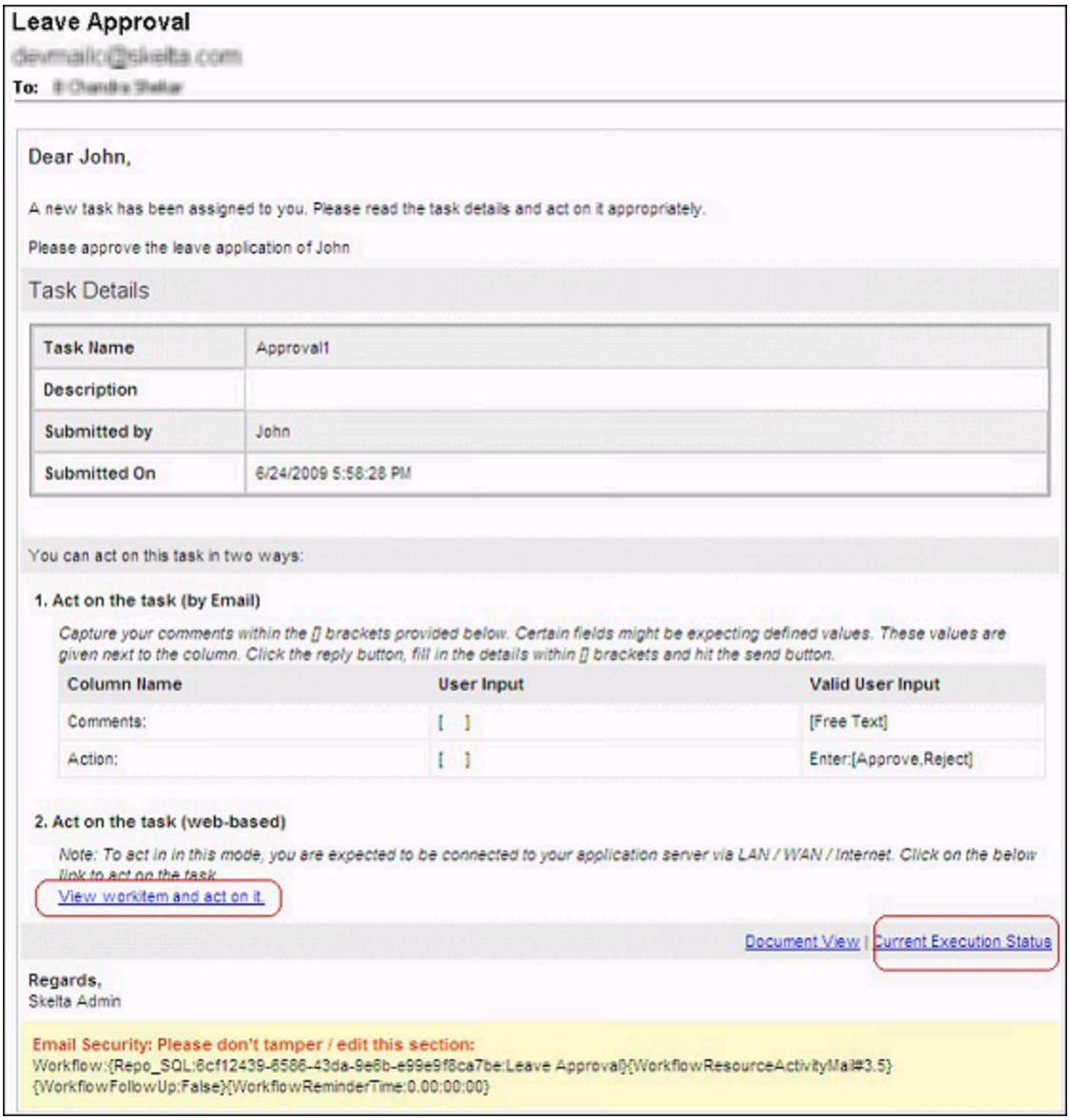


For example, if the application is *Repo*, workflow is *WF,* and the application URL is **http://localhost:8000/EnterpriseConsole/BPMUITemplates/Default/Email/,** then the entries in the **SKAddInProvidersSettings** table is as shown in the following image:

Note:
- For each workflow, a separate (Follow step 3) entry should be added in the **SKAddInProviderSettings** table.
- If a workflow is not configured for a mail channel, then the system considers the first entry of the repository, which could be an entry configured for another workflow.

The following image shows the Document View, Current Execution View Status, and Work Item view:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Note: Restart the Workflow Engine service after making changes to the Repository Database configuration.

Support Variables, Contents, and XML Variables in Email Templates

The following are the Activity Email Templates found in the Email and Microsoft Outlook integration:

- Approval Activity Email Template
- Choice Activity Email Template
- Check-list Activity Email Template
- Task Activity Email Template
- Information Activity Email Template

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Approval Activity Email Template

AVEVA Work Tasks Engine accesses the approval activity's email template from the following physical path:

```
AVEVA\AVEVA Work Tasks\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\
ApprovalEmail.html
```

This template can be modified by the user as per their UI requirement. The mails received by the actor is based on the above mentioned mail template.

The following is a default email template text for the **ApprovalEmail.html** file:

ApprovalEmail.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight: bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td><span class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>

<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

     Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>

To make use of localization, Create a Key and its value in SkeltaInstalledFolder/Locales/
en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in SkeltaInstalledFolder/Locales/en-
US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.



TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->

<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>


</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
cellspacing='1' style="width: 100%; height: 25px;">
<tr>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%"><%#locComments%>:</td>
<td class="style2" bgcolor='#FFFFFF' width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%" >[<%#locFreeText%>]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%"><%#locTakeAction%>:</td>
<td class="style2" bgcolor='#FFFFFF' width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%"
><%#locEnter%>:[<%#locApproveAction%>,<%#locRejectAction%>]</td>
</tr>
</table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActivityMa
il#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#SkeltaReminderTi
meValue%>}</td>
</tr>
</table></td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
```

Note: While modifying the Approval email template the following place holders should not be modified, as these are necessary for the engine to process the mails that are received from the actors.

1. Email URL

   ```
   <form id='Form1' action='<%#EmailURL%>' method='post' >
   ```

2. Notes

   ```
   <td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
   ```

3. Complete content

   ```
   <table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
   cellspacing='1' style="width: 100%; height: 25px;">
   <tr>
   <td class="style2" width='30%' bgcolor='#EBEBEB'>
   <strong><%#locColumnName%></strong></td>
   <td class="style2" width='30%' bgcolor='#EBEBEB'>
   <strong><%#locUserInputs%></strong></td>
   <td class="style2" bgcolor='#EBEBEB'>
   <strong><%#locValidInputs%></strong></td>
   </tr>
   <tr valign="top">
   <td class="style2" bgcolor='#FFFFFF' width="10%"><%#locComments%>:</td>
   <td class="style2" bgcolor='#FFFFFF' width="10%">[     ]</td>
   <td class="style2" bgcolor='#FFFFFF' width="20%" >[<%#locFreeText%>]</td>
   </tr>
   <tr valign="top">
   <td class="style2" bgcolor='#FFFFFF' width="10%"><%#locTakeAction%>:</td>
   <td class="style2" bgcolor='#FFFFFF' width="10%">[     ]</td>
   <td class="style2" bgcolor='#FFFFFF' width="20%"
   ><%#locEnter%>:[<%#locApproveAction%>,<%#locRejectAction%>]</td>
   </tr>
   </table>
   ```

4. Complete content

   ```
   <td class="style2" width="95%"><span
   class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
   Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActivit
   yMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#SkeltaRemi
   nderTimeValue%>}</td>
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Use Variables in the Approval Email Template

Email templates can be used to display the values of the variables that have been created in a workflow.

Consider that a variable of type string by name **'Status'** is created in **Start** activity and its value is updated as **"Variable, status is updated".** User wants to do the following:

1. Display the variable's value in Approval email.
2. Display the variable's value after **Submitted** on row in the **Task Details** Table.

Solution

To display this variable's value in Approval email and to display the variable's value after Submitted on row in the Task Details Table:

Steps:

1. Copy **ApprovalEmail.html** file.
2. Use the **<%#Variable.Variablename%>** placeholder in the html template for viewing the variable's value, where <Variablename> is a name of the variable.

   Since the variable name is "Status", the placeholder to be used is <%#Variable.Status%> where the value is to be displayed. The highlighted portion in the following code displays the position where the **<%#Variable.Status%>** placeholder is added.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>

<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>

To make use of localization, Create a Key and its value in SkeltaInstalledFolder/
Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in SkeltaInstalledFolder/
Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.



TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->

<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>User ID</b></td>
<td bgcolor="#EFEFEF">
<%#XMLVariables.Users.User.ID%></td>
</tr>
<tr>
<td>
<b>User Email</b></td>
<td>
<%#XMLVariables.Users.User.Email%></td>
</tr>
<tr>
<td  bgcolor="#EFEFEF">
<b>User Name</b></td>
<td  bgcolor="#EFEFEF">
<%#XMLVariables.Users.User.Name%></td>
</tr>
</table>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
cellspacing='1' style="width: 100%; height: 25px;">
<tr>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%"><%#locComments%>:</td>
<td class="style2" bgcolor='#FFFFFF'
width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%" >[<%#locFreeText%>]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%"><%#locTakeAction%>:</td>
<td class="style2" bgcolor='#FFFFFF'
width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%"
><%#locEnter%>:[<%#locApproveAction%>,<%#locRejectAction%>]</td>
</tr>
</table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
</tr>
</table></td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>

<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>

To make use of localization, Create a Key and its value in SkeltaInstalledFolder/
Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in SkeltaInstalledFolder/
Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.



TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->

<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
</tr>
 <tr>
<td>
<b>Status</b></td>
<td>
<%#Variable.Status%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
cellspacing='1' style="width: 100%; height: 25px;">
<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%"><%#locComments%>:</td>
<td class="style2" bgcolor='#FFFFFF'
width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%" >[<%#locFreeText%>]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%"><%#locTakeAction%>:</td>
<td class="style2" bgcolor='#FFFFFF'
width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%"
><%#locEnter%>:[<%#locApproveAction%>,<%#locRejectAction%>]</td>
</tr>
</table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
</tr>
</table></td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
```

3. The actor will get the approval email in the defined format for the above template.

Note: Out of box variables such as <%#Variable.SubmittedBy.UserId%>, <%#Variable.SubmittedBy.UserName%>, and <%#Variable.SubmittedBy.UserDisplayName%> can be used for Active Directory provider.

Use Contents in the Approval Email Template

Email templates can be used to display the values of the contents that have been created in a workflow.

Consider that a Content of type number and by name 'EmployeeAge' is created in the Start activity and at runtime its value is 21. User wants to do the following:

1. Display the content's value in Approval email.
2. Display the variable's value after Submitted on row in the Task Details Table.

Solution

To display the content's value in Approval email and to display the variable's value after **Submitted** on row in the **Task Details** Table:

Steps:

1. Copy **AprovalEmail.html** file.
2. Use the **<%#Variable.Contentname%>** placeholder in the html template for viewing the content's value where <Contentname> is a name of the content.

   Use the **<%#Content.EmployeeAge%>** placeholder in the html template, where the value is to be displayed. Since the Content name is "EmployeeAge", the placeholder to be used is <%#Content.EmployeeAge%>. The highlighted portion in the below code displays the position where the <%#Content.EmployeeAge%> placeholder is added.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>

<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
To make use of localization, Create a Key and its value
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.



TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->

<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
 </tr>
<tr>
<td>
<b>Employee Age</b></td>
<td>
<%#Content.EmployeeAge%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
cellspacing='1' style="width: 100%; height: 25px;">
<tr>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%"><%#locComments%>:</td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td class="style2" bgcolor='#FFFFFF'
width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%" >[<%#locFreeText%>]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%"><%#locTakeAction%>:</td>
<td class="style2" bgcolor='#FFFFFF'
width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%"
><%#locEnter%>:[<%#locApproveAction%>,<%#locRejectAction%>]</td>
</tr>
</table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
</tr>
</table></td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
```

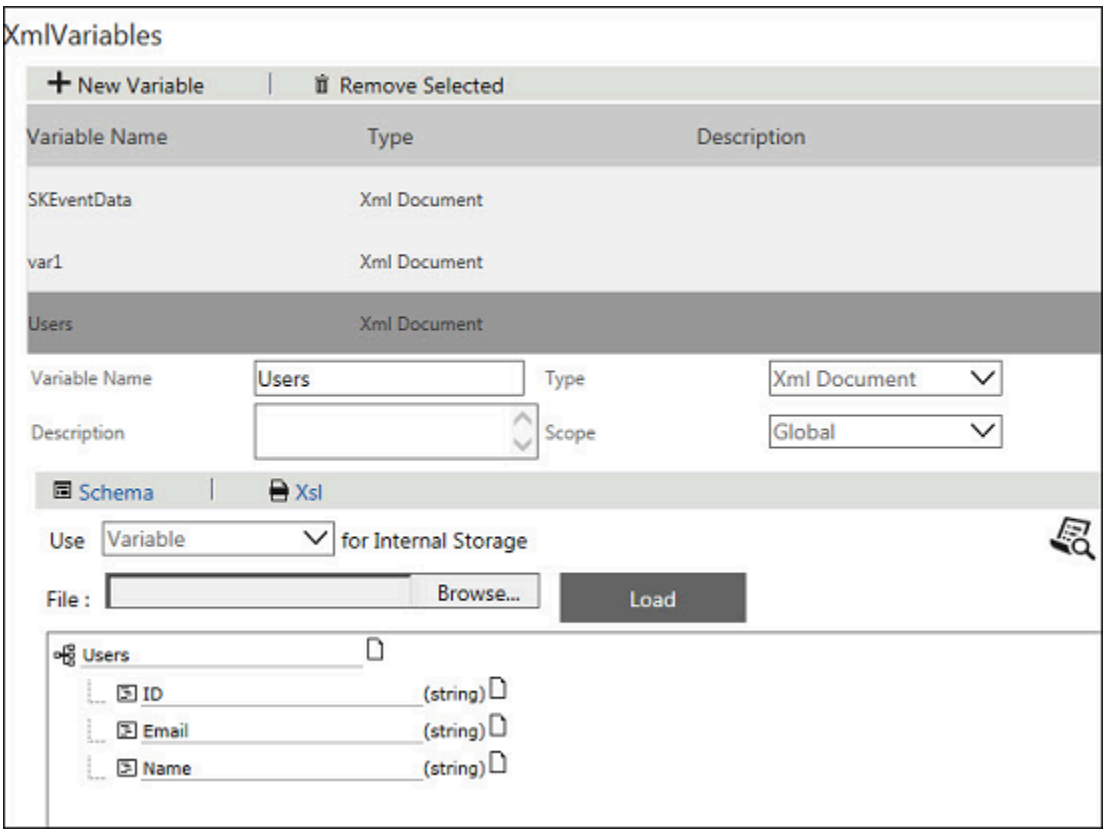
3. The actor will get the defined mail format for the above template.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Use XML Variables in the Approval Email Template

Email templates can be used to display the values of the XML variables that have been created in a workflow.

Consider that 'Users' XML Variable of type XMLDocument is created in the **Start** activity and schema is as follows:



The scenario requirement is that during the workflow execution, the user wants to display the following values during the workflow execution:

1. **ID** is updated to 1029.
2. **Email** is updated to John.Smith@domain.com
3. **Name** is updated to John.
4. Display these XML variable's value after **Submitted** on row in the Task Details Table.

Solution

To display this the XML Variable's value in Approval email and to display these variable's value after "Submitted on" row in the Task Details Table:

Steps:

1. Copy **AprovalEmail.html** file.
2. Use the **<%#XmlVariables.variablename%>** placeholder in the html template for viewing the XML Variable's value. Where <variablename> is a name of the XML variable.

   Since the 'Users' XML Variable contents the ID, Email and Name, use the following XML Variables in the AprovalEmail.html template, where the values are to be displayed.

   ```
   <%#XmlVariables.Users.User.ID%>
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<%#XmlVariables.Users.User.Email%>
<%#XmlVariables.Users.User.Name%>
```

The highlighted portion in the below code displays the position where the above placeholders to be added.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>

<!-- Appropriate sender details can be passed to the mail by using the sender detail
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
   contents as follows
   A few are listed below
   <%#Value.SendTo.UserDisplayName%>
   <%#Value.SendTo.UserEmail%>
   <%#Value.SendTo.UserId%>

   Contents declared in Start Activity can also be used in a similar way.

   XML Variables can also be used.
   Variables defined in Start Activity can also be used pass the values to the mail.
```

```
       Pre-defined variables can also be used.
   A few are listed below.
   <%#Variable.SubmittedBy.UserId%>
   <%#Variable.SubmittedBy.UserName%>
   <%#Variable.SubmittedBy.UserDisplayName%>
```

```
To make use of localization, Create a Key and its value
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.
         TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->

<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>User ID</b></td>
<td bgcolor="#EFEFEF">
<%#XMLVariables.Users.User.ID%></td>
</tr>
<tr>
<td>
<b>User Email</b></td>
<td>
<%#XMLVariables.Users.User.Email%></td>
</tr>
<tr>
<td  bgcolor="#EFEFEF">
<b>User Name</b></td>
<td  bgcolor="#EFEFEF">
<%#XMLVariables.Users.User.Name%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
cellspacing='1' style="width: 100%; height: 25px;">
<tr>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%"><%#locComments%>:</td>
<td class="style2" bgcolor='#FFFFFF' width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%" >[<%#locFreeText%>]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%"><%#locTakeAction%>:</td>
<td class="style2" bgcolor='#FFFFFF' width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%"
><%#locEnter%>:[<%#locApproveAction%>,<%#locRejectAction%>]</td>
</tr>
</table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActivit
yMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#SkeltaRemi
nderTimeValue%>}</td>
</tr>
</table></td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
```

1. The actor will get the mail in the defined format for the above template.

Choice Activity Email Template

AVEVA Work Tasks Engine accesses the Choice activity's email template from the physical path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\Choice.html

This template can be modified by the user as per their UI requirement. The mails received by the actor is based on the above mentioned mail template.

The following is default email template text for the **Choice.html** file:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight: bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>

<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>

To make use of localization, Create a Key and its value
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in WorkflowManagementInstalledFolder/
Locales/en-US/EmailTemplate.txt
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.

```
         TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->
```

```
<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td>
 <b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>


</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
cellspacing='1' style="width: 100%; height: 25px;">
<tr>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locComments%>:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[<%#locFreeText%>]</td>
</tr>

<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locTakeAction%>:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'><%#locEnter%>:[<%#action%>]</td>
</tr>
</table>
 </td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActivityMa
il#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#SkeltaReminderTi
meValue%>}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>
```

Note: While modifying the Choice email template the following placeholders should not be modified, as these are necessary for the engine to process the mails that are received from the actors.

1. Email URL

   ```
   <form id='Form1' action='<%#EmailURL%>' method='post' >
   ```

2. Notes

   ```
   <td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
   ```

3. Complete content

   ```
   <table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
   cellspacing='1' style="width: 100%; height: 25px;">
   <tr>
   <td class="style2" width='30%' bgcolor='#EBEBEB'>
   <strong><%#locColumnName%></strong></td>
   <td class="style2" width='30%' bgcolor='#EBEBEB'>
   <strong><%#locUserInputs%></strong></td>
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td class="style2" bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locComments%>:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[<%#locFreeText%>]</td>
</tr>

<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locTakeAction%>:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'><%#locEnter%>:[<%#action%>]</td>
</tr>
</table>
```

4. Complete content

```
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
```

Use Variables in Choice Email Template

Email templates can be used to display the values of the variables that have been created in a workflow.

Consider that a variable of type string by name 'Status' is created in the Start activity and its value is updated as Variable, status is updated. User wants to do the following:

1. Display the variable's value in Choice email.
2. Display the variable's value after Submitted on row in the Task Details Table.

Solution

To display this variable's value in Choice email and to display the variable's value after Submitted on row in the Task Details Table:

Steps:

1. Copy **Choice.html** file.
2. Use the **<%#Variable.Variablename%>** placeholder in the html template for viewing the variable's value where <Variablename> is a name of the variable.

   Since the variable name is "Status", the placeholder to be used is <%#Variable.Status%> where the value is to be displayed. The highlighted portion in the below code displays the position where the **<%#Variable.Status%>** placeholder is added.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>
<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
 <b><%#locSubmittedOn%></b></td>
<td>
<%#SubmitedOn%></td>
</tr> <tr>
<td>
<b>Status</b></td>
<td>
<%#Variable.Status%></td>
</tr>


</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
cellspacing='1' style="width: 100%; height: 25px;">
<tr>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong>Column Name</strong></td>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong>User Input</strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong>Valid User Input</strong></td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'>Comments:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[Free Text]</td>
</tr>

<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'>Action:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>Enter:[<%#action%>]</td>
</tr>
</table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>
```

1. The actor will get the Choice email in the defined format for the above template.

Note: Out of box variables such as <%#Variable.SubmittedBy.UserId%> , <%#Variable.SubmittedBy.UserName%>, and <%#Variable.SubmittedBy.UserDisplayName%> can be used for Active Directory provider.

Use Contents in Choice Email Template

Email templates can be used to display the values of the contents that have been created in a workflow.

Consider that a **Content** of type number and by name 'EmployeeAge' is created in the **Start** activity and at runtime its value is 21.

User wants to do the following:

1. Display the content's value in Choice email.
2. Display the variable's value after **"Submitted on"** row in the **Task Details** Table.

Solution

To display this the content's value in Choice email and to display the variable's value after "Submitted on" row in the Task Details Table:

Steps:

1. Copy **Choice.html** file.
2. Use the **<%#Variable.Contentname%>** placeholder in the html template for viewing the content's value where <Contentname> is a name of the content.

   Use the **<%#Content.EmployeeAge%>** placeholder in the html template, where the value is to be displayed. Since the Content name is "EmployeeAge", the placeholder to be used is <%#Content.EmployeeAge%>. The highlighted portion in the below code displays the position where the <%#Content.EmployeeAge%> placeholder is added.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>

<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>

To make use of localization, Create a Key and its value
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.



TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->

<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
 <b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
<tr>
<td>
<b>Status</b></td>
<td>
<%#Variable.Status%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
cellspacing='1' style="width: 100%; height: 25px;">
<tr>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locComments%>:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[<%#locFreeText%>]</td>
</tr>

<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locTakeAction%>:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'><%#locEnter%>:[<%#action%>]</td>
</tr>
</table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

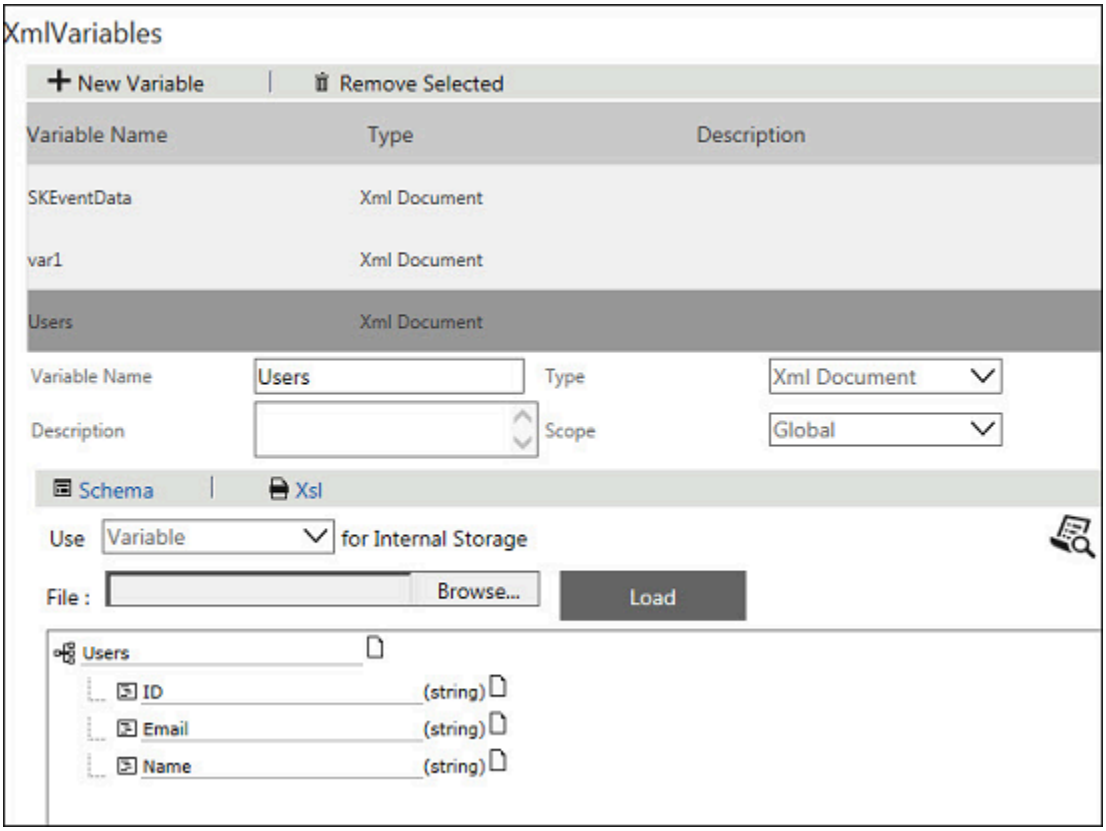
```
</form>
</body>
</html>
```

3. The actor will get the mail in the defined format for the above template.

Use XML Variables in Choice Email Template

Email templates can be used to display the values of the XML variables that have been created in a workflow.

Consider that 'Users' XML Variable of type XMLDocument is created in the **Start** activity and the schema is as follows:



The scenario requirement is that during the workflow execution, the user wants to display the following values during the workflow execution:

1. ID is updated to 1029.
2. Email is updated to John.Smith@domain.com
3. Name is updated to John.
4. Display this XML variable's value after Submitted on row in the Task Details table.

Solution

To display this XML Variable's value in Choice email and to display this variable's value after "Submitted on" row in the Task Details Table, the following steps need to be followed:

Steps:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Copy **Choice.html** file.
2. Use the <%#XmlVariables.variablename%> placeholder in the html template for viewing the XML Variable's value. Where <variablename> is a name of the XML variable.

   Since the 'Users' XML Variable contents the ID, Email and Name, use the following XML Variables in the Choice.html template, where the values are to be displayed.

```
<%#XmlVariables.Users.User.ID%>
<%#XmlVariables.Users.User.Email%>
<%#XmlVariables.Users.User.Name%>
The highlighted portion in the below code displays the position where the above
placeholders to be added.
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>

<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>

To make use of localization, Create a Key and its value
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.



TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
 <b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>User ID</b></td>
<td bgcolor="#EFEFEF">
<%#XmlVariables.Users.User.ID%></td>
</tr>
<tr>
<td>
<b>User Email</b></td>
<td>
<%#XmlVariables.Users.User.Email%></td>
</tr>
<tr>
<td  bgcolor="#EFEFEF">
<b>User Name</b></td>
<td  bgcolor="#EFEFEF">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<%#XmlVariables.Users.User.Name%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
cellspacing='1' style="width: 100%; height: 25px;">
<tr>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locComments%>:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[<%#locFreeText%>]</td>
</tr>

<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locTakeAction%>:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'><%#locEnter%>:[<%#action%>]</td>
</tr>
</table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>
```

1. The actor will get the mail in the defined format for the above template.

Check-List Activity Email Template

AVEVA Work Tasks Engine accesses the Checklist activity's email template from the physical path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\CheckListEmail.html

This template can be modified by the user as per their UI requirement. The mails received by the actor is based on the above mentioned mail template.

The following is default email template text for the CheckListEmail.html file:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight: bold; }
-->
</style>
</head>

<body>
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span class="Username"><%#locSalutation%> <strong><%#ActorName%></strong> </span>,
<p class="style2"><%#locMailDescription%></p></td></tr>

<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style4"><%#locResponseOptions%></td>
</tr>
<tr>
<td>
<table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">
<tr>
<td><a <%#workitemView%>><%#locRespondByWeb%></a></td>
</tr>
<tr>
<td><a <%#docview%>><%#locDocViewURLText%></a></td>
</tr>
<tr>
<td ><a <%#executionView%>><%#locExecutionViewURLText%></a></td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</table>
</td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style4" style="height:10%"></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

Note: While modifying the Check-List email template the following placeholders should not be modified, as these are necessary for the Skelta engine to process the mails that are received from the actors.

1. Email URL

   ```
   <form id='Form1' action='<%#EmailURL%>' method='post'>
   ```

2. Notes

   ```
   <td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
   ```

3. Complete content

   ```
   <table border='0' align='center' cellpadding='5' cellspacing='1' bgcolor='#EBEBEB'
   class="style2" style="width: 100%; height: 25px;">
   <tr bgcolor='#EBEBEB' class="style2">
   <td  bgcolor='#EBEBEB' class="style2"><strong><%#locColumnName%></strong></td>
   <td  bgcolor='#EBEBEB' class="style2"><strong><%#locUserInputs%></strong></td>
   <td  bgcolor='#EBEBEB' class="style2"><strong><%#locValidInputs%></strong></td>
   </tr> <%#InputPlaceHolderString%>
    </table>
   ```

4. Complete content

   ```
   <td width="95%" class="style2"><span
   class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
   Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
   vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
   taReminderTimeValue%>}</td>
   ```

Use Variables in Check-List Email Template

Email templates can be used to display the values of the variables that have been created in a workflow.

Consider that a variable of type string by name 'Status' is created in the Start activity and its value is updated as "Variable, status is updated".

User wants to do the following:

1. Display the variable's value in Check-List email.
2. Display the variable's value after "Submitted on" row in the Task Details Table.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Solution

To display this variable's value in Check-List email and to display the variable's value after "Submitted on" row in the Task Details Table, the following steps need to be followed:

Steps:

1. Copy CheckListEmail.html file.
2. Use the <%#Variable.Variablename%> placeholder in the html template for viewing the variable's value where <Variablename> is a name of the variable.

   Since the variable name is "Status", the placeholder to be used is <%#Variable.Status%> where the value is to be displayed. The highlighted portion in the below code displays the position where the <%#Variable.Status%> placeholder is added.

```
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>

<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>

To make use of localization, Create a Key and its value
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.


TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</td>
</tr>-->

<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
<tr>
<td>
<b>Status</b></td>
<td>
<%#Variable.Status%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table border='0' align='center' cellpadding='5' cellspacing='1' bgcolor='#EBEBEB'
class="style2" style="width: 100%; height: 25px;">
<tr bgcolor='#EBEBEB' class="style2">
<td  bgcolor='#EBEBEB' class="style2"><strong><%#locColumnName%></strong></td>
<td  bgcolor='#EBEBEB' class="style2"><strong><%#locUserInputs%></strong></td>
<td  bgcolor='#EBEBEB' class="style2"><strong><%#locValidInputs%></strong></td>
</tr> <%#InputPlaceHolderString%>
 </table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>

<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td width="95%" class="style2"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
taReminderTimeValue%>}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>
```

3. The actor will get the Check-List email in the defined format for the above template.

Note: Out of box variables such as <%#Variable.SubmittedBy.UserId%> , <%#Variable.SubmittedBy.UserName%>, and <%#Variable.SubmittedBy.UserDisplayName%> can be used for Active Directory provider.

Use Contents in Check-List Email Template

Email templates can be used to display the values of the contents that have been created in a workflow.

Consider that a **Content** of type number and by name 'EmployeeAge' is created in **Start** activity and at runtime its value is 21.

User wants to do the following:

1. Display the content's value in Check-List email.
2. Display the variable's value after "**Submitted on"** row in the **Task Details** table.

Solution

To display this the content's value in Check-List email and to display the variable's value after "Submitted on" row in the Task Details Table, the following steps need to be followed:

Steps:

1. Copy **CheckListEmail.html** file.
2. Use the **<%#Variable.**Contentname%> placeholder in the html template for viewing the content's value. Where <Contentname> is a name of the content.

   Use the **<%#Content.EmployeeAge%>** placeholder in the html template, where the value is to be displayed. Since the Content name is "EmployeeAge", the placeholder to be used is <%#Content.EmployeeAge%>. The highlighted portion in the below code displays the position where the <%#Content.EmployeeAge%> placeholder is added.

```
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
font-family: Arial, Helvetica, sans-serif;
font-size: 11px;
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
.style4 {
font-family: Arial, Helvetica, sans-serif;
font-size: 15px;
font-weight: Normal;
color: #333333;
}
.Username {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: Bold;
color: #333333;
}
.subject {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: Bold;
color: #333333;
}
.style8 {
color: #FF3300;
font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>
<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF"> d
<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>
<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>
Contents declared in Start Activity can also be used in a similar way.
XML Variables can also be used.
Variables defined in Start Activity can also be used pass the values to the mail.
Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>
To make use of localization, Create a Key and its value in
WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in
WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.
TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->
<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
<tr>
<td>
<b>Employee Age</b></td>
<td>
<%#Content.EmployeeAge%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>
<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">
<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table border='0' align='center' cellpadding='5' cellspacing='1' bgcolor='#EBEBEB'
class="style2" style="width: 100%; height: 25px;">
<tr bgcolor='#EBEBEB' class="style2">
<td bgcolor='#EBEBEB' class="style2"><strong><%#locColumnName%></strong></td>
<td bgcolor='#EBEBEB' class="style2"><strong><%#locUserInputs%></strong></td>
<td bgcolor='#EBEBEB' class="style2"><strong><%#locValidInputs%></strong></td>
</tr> <%#InputPlaceHolderString%>
</table>
</td>
</tr>
<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
</td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td width="95%" class="style2"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>
```

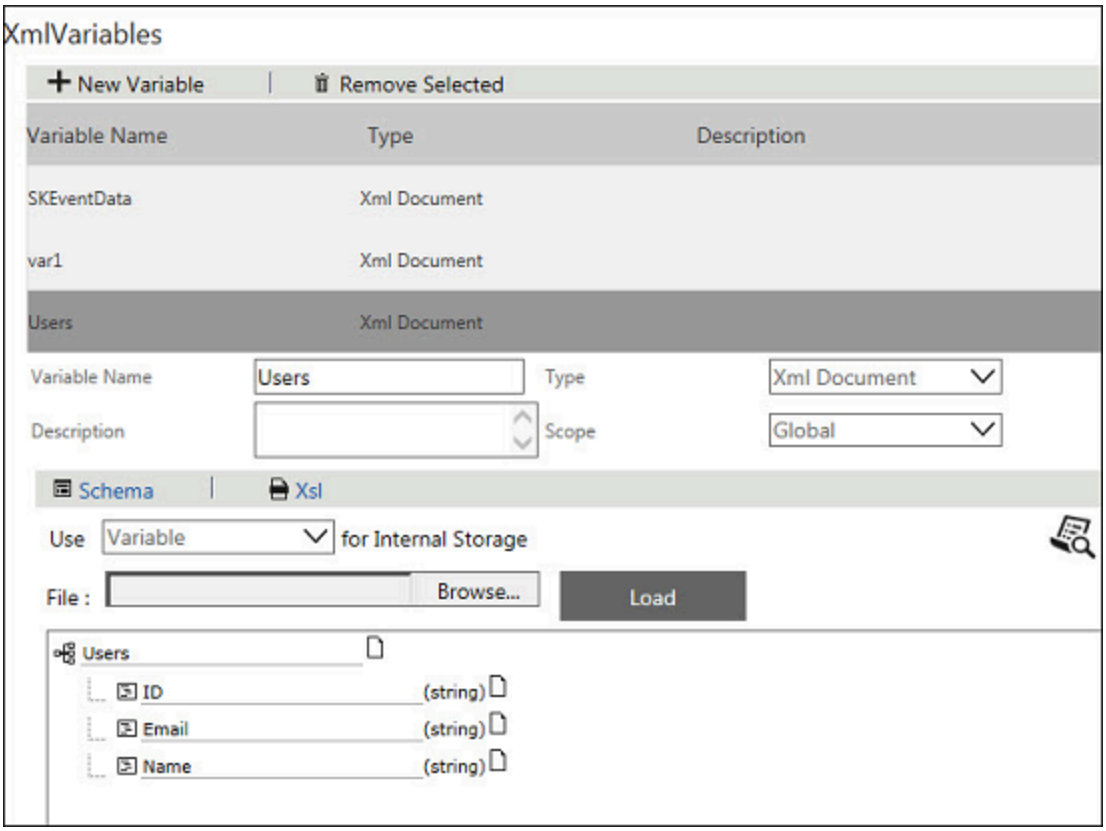
The actor will get the mail in the defined format for the above template.

Use XML Variables in Check-List Email Template

Email templates can be used to display the values of the XML variables that have been created in a workflow.

Consider that 'Users' XML Variable of type XML Document is created in the **Start** activity and schema is as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The scenario requirement is that, during the workflow execution,

User wants to display the following values during the workflow execution

1. **ID** is updated to 1029.
2. **Email** is updated to John.Smith@domain.com
3. **Name** is updated to John.
4. Display these XML variable's value after "Submitted on" row in the Task Details table.

Solution

To display this the XML Variable's value in Check-List email and to display these variable's value after "Submitted on" row in the Task Details Table:

Steps:

1. Copy **CheckListEmail.html** file.
2. Use the **<%#XmlVariables.variablename%>** placeholder in the html template for viewing the XML Variable's value. Where <variablename> is a name of the XML variable.

   Since the 'Users' XML Variable contents the ID, Email and Name, use the following XML Variables in the CheckListEmail.html template, where the values are to be displayed.

   ```
   <%#XmlVariables.Users.User.ID%>
   <%#XmlVariables.Users.User.Email%>
   <%#XmlVariables.Users.User.Name%>
   ```

   The highlighted portion in the below code displays the position where the above placeholders to be added.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>

<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>

To make use of localization, Create a Key and its value
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.
         TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF
CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->

<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>User ID</b></td>
<td bgcolor="#EFEFEF">
<%#XmlVariables.Users.User.ID%></td>
</tr>
<tr>
<td>
<b>User Email</b></td>
<td>
<%#XmlVariables.Users.User.Email%></td>
</tr>
<tr>
<td  bgcolor="#EFEFEF">
<b>User Name</b></td>
<td  bgcolor="#EFEFEF">
<%#XmlVariables.Users.User.Name%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table border='0' align='center' cellpadding='5' cellspacing='1' bgcolor='#EBEBEB'
class="style2" style="width: 100%; height: 25px;">
<tr bgcolor='#EBEBEB' class="style2">
<td  bgcolor='#EBEBEB' class="style2"><strong><%#locColumnName%></strong></td>
<td  bgcolor='#EBEBEB' class="style2"><strong><%#locUserInputs%></strong></td>
<td  bgcolor='#EBEBEB' class="style2"><strong><%#locValidInputs%></strong></td>
</tr> <%#InputPlaceHolderString%>
 </table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>

<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td width="95%" class="style2"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
</tr>
</table></td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
   </table></td>
   </tr>
   </table>
   </form>
   </body>
   </html>
```

3. The actor will get the mail in the defined format for the above template.

Task Activity Email Template

AVEVA Work Tasks Engine accesses the Task activity's email template from the physical path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\TaskEmail.html

This template can be modified by the user as per their UI requirement. The mails received by the actor is based on the above mentioned mail template.

The following is default email template text for the **TaskEmail.html** file:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight: bold; }
-->
</style>
</head>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>

<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>

    <!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>

To make use of localization, Create a Key and its value
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in WorkflowManagementInstalledFolder/
Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.
TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>
</table>
</td>
</tr>-->

<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>


</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>

 <tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' cellpadding='5' cellspacing='1'
bgcolor='#EBEBEB' id="TABLE1" style="width: 100%; height: 25px;">
<tr class="style2" bgcolor='#EBEBEB'>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
<td class="style2" width='30%'  bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2"  bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locComments%>:</td>
<td class="style2" width='30%'  bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[<%#locFreeText%>]</td>
</tr>

<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'>
<%#TaskStatus%>:</td>
<td class="style2" width='30%'  bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>
[<%#FreeTextTaskStatus%>]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor="#ffffff" width="30%" >
<%#locTakeAction%>:</td>
<td class="style2" bgcolor="#ffffff" width="30%" >
[<%#locSubmit%>]</td>
<td class="style2" bgcolor="#ffffff" >
<%#locEnter%>:[<%#locSubmit%>]</td>
</tr>
 </table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActivityMa
il#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#SkeltaReminderTi
meValue%>}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>
```

Note: While modifying the Task email template the following placeholders should not be modified, as these are necessary for the engine to process the mails that are received from the actors.

1. Email URL

   ```
   <form id='Form1' action='<%#EmailURL%>' method='post' >
   Notes
   <td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
   ```

2. Complete content

   ```
   <table width='95%' border='0' align='center' cellpadding='5' cellspacing='1'
   bgcolor='#EBEBEB' id="TABLE1" style="width: 100%; height: 25px;">
   <tr class="style2" bgcolor='#EBEBEB'>
   <td class="style2" width='30%' bgcolor='#EBEBEB'>
   <strong><%#locColumnName%></strong></td>
   <td class="style2" width='30%'  bgcolor='#EBEBEB'>
   <strong><%#locUserInputs%></strong></td>
   <td class="style2"  bgcolor='#EBEBEB'>
   <strong><%#locValidInputs%></strong></td>
   </tr>
   <tr valign="top">
   <td class="style2" width='30%' bgcolor='#FFFFFF'><%#locComments%>:</td>
   <td class="style2" width='30%'  bgcolor='#FFFFFF'>[  ]</td>
   <td class="style2" bgcolor='#FFFFFF'>[<%#locFreeText%>]</td>
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>

<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'>
<%#TaskStatus%>:</td>
<td class="style2" width='30%'  bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>
[<%#FreeTextTaskStatus%>]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor="#ffffff" width="30%" >
<%#locTakeAction%>:</td>
<td class="style2" bgcolor="#ffffff" width="30%" >
[<%#locSubmit%>]</td>
<td class="style2" bgcolor="#ffffff" >
<%#locEnter%>:[<%#locSubmit%>]</td>
</tr>
 </table>
```

3. Complete content

```
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
```

Use Variables in Task Email Template

Email templates can be used to display the values of the variables that have been created in a workflow.

Consider that a variable of type string by name **'Status'** is created in the **Start** activity and its value is updated as **"Variable, status is updated"**.

User wants to do the following:

1. Display the variable's value in Task email.
2. Display the variable's value after "Submitted on" row in the Task Details table.

Solution

To display this variable's value in Task email and to display the variable's value after "Submitted on" row in the Task Details Table:

Steps:

1. Copy TaskEmail.html file.
2. Use the <%#Variable.Variablename%> placeholder in the html template for viewing the variable's value where <Variablename> is a name of the variable.

   Since the variable name is "Status", the placeholder to be used is <%#Variable.Status%> where the value is to be displayed. The highlighted portion in the below code displays the position where the <%#Variable.Status%> placeholder is added.

```
<html>
<head>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>

<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>



<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>

To make use of localization, Create a Key and its value in Wonderware Skelta BPM
Installed Folder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in Wonderware Skelta BPM
InstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.



TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->

<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->

<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
<tr>
<td>
<b>Status</b></td>
<td>
<%#Variable.Status%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td><strong><%#locRespondbyMail%></strong></td>
</tr>

 <tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' cellpadding='5' cellspacing='1'
bgcolor='#EBEBEB' id="TABLE1" style="width: 100%; height: 25px;">
<tr class="style2" bgcolor='#EBEBEB'>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
<td class="style2" width='30%'  bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2"  bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locComments%>:</td>
<td class="style2" width='30%'  bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[<%#locFreeText%>]</td>
</tr>

<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'>
<%#TaskStatus%>:</td>
<td class="style2" width='30%'  bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>
[<%#FreeTextTaskStatus%>]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor="#ffffff" width="30%" >
<%#locTakeAction%>:</td>
<td class="style2" bgcolor="#ffffff" width="30%" >
[<%#locSubmit%>]</td>
<td class="style2" bgcolor="#ffffff" >
<%#locEnter%>:[<%#locSubmit%>]</td>
</tr>
 </table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>
```

3. The actor will get the Task email in the defined format for the above template

Note: Out of box variables such as <%#Variable.SubmittedBy.UserId%> , <%#Variable.SubmittedBy.UserName%>, and <%#Variable.SubmittedBy.UserDisplayName%> can be used for Active Directory provider.

Use Contents in Task Email Template

Email templates can be used to display the values of the contents that have been created in a workflow.

Consider that a Content of type number and by name 'EmployeeAge' is created in the Start activity and at runtime its value is 21.

User wants to do the following:

1. Display the content's value in Task email.
2. Display the variable's value after **"Submitted on"** row in the **Task Details** Table.

Scenario Solution

To display this the content's value in Task email and to display the variable's value after **"Submitted on"** row in the **Task Details** table:

Steps:

1. Copy **TaskEmail.html** file.
2. Use the **<%#Variable.Contentname%>** placeholder in the html template for viewing the content's value where <Contentname> is a name of the content.

   Use the <%#Content.EmployeeAge%> placeholder in the html template, where the value is to be displayed. Since the Content name is "EmployeeAge", the placeholder to be used is <%#Content.EmployeeAge%>. The

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

highlighted portion in the below code displays the position where the <%#Content.EmployeeAge%> placeholder is added.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>
<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>

<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>
<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>

To make use of localization, Create a Key and its value in Wonderware Skelta BPM
InstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in Wonderware Skelta BPM
InstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.
         TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF
CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->

<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
  <tr>
<td>
<b>Employee Age</b></td>
<td>
<%#Content.EmployeeAge%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 <tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' cellpadding='5' cellspacing='1'
bgcolor='#EBEBEB' id="TABLE1" style="width: 100%; height: 25px;">
<tr class="style2" bgcolor='#EBEBEB'>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
<td class="style2" width='30%'  bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2"  bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locComments%>:</td>
<td class="style2" width='30%'  bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[<%#locFreeText%>]</td>
</tr>

<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'>
<%#TaskStatus%>:</td>
<td class="style2" width='30%'  bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>
[<%#FreeTextTaskStatus%>]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor="#ffffff" width="30%" >
<%#locTakeAction%>:</td>
<td class="style2" bgcolor="#ffffff" width="30%" >
[<%#locSubmit%>]</td>
<td class="style2" bgcolor="#ffffff" >
<%#locEnter%>:[<%#locSubmit%>]</td>
</tr>
 </table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
```

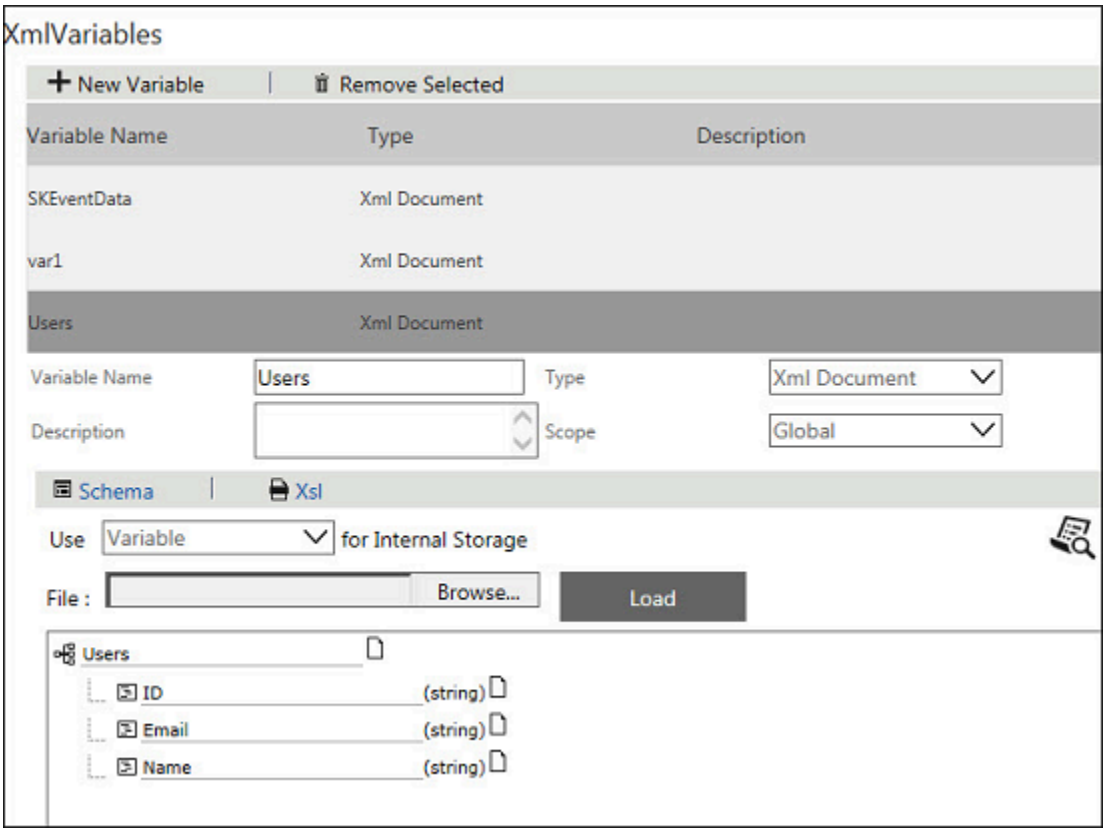
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>
```

3. The actor will get the mail in the defined format for the above template.

Use XML Variables in Task Email Template

Email templates can be used to display the values of the XML variables that have been created in a workflow. Consider that 'Users' XML Variable of type XMLDocument is created in the **Start** activity and schema is as follows:



The scenario requirement is that during the workflow execution, the user wants to display the following values

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

during the workflow execution.

1. **ID** is updated to 1029.
2. **Email** is updated to John.Smith@domain.com
3. **Name** is updated to John.
4. Display these XML variables' value after **"Submitted on"** row in the **Task Details** Table.

Solution

To display this the XML Variable's value in Task email and to display these variable's value after "Submitted on" row in the Task Details Table:

Steps:

1. Copy TaskEmail.html file.
2. Use the **<%#XmlVariables.variablename%>** placeholder in the html template for viewing the XML Variable's value where <variablename> is a name of the XML variable.

   Since the 'Users' XML Variable contents the ID, Email and Name, use the following XML Variables in the TaskEmail.html template, where the values are to be displayed.

   ```
   <%#XmlVariables.Users.User.ID%>
   <%#XmlVariables.Users.User.Email%>
   <%#XmlVariables.Users.User.Name%>
   ```

   The highlighted portion in the below code displays the position where the above placeholders to be added

   ```
   <html>
   <head>
   <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
   <title>Mailer</title>
   <style type="text/css">
   <!--
   .style2 {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 11px;
   }
   .style4 {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 15px;
    font-weight: Normal;
    color: #333333;
   }
   .Username {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 13px;
    font-weight: Bold;
    color: #333333;
   }
   .subject {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 13px;
    font-weight: Bold;
    color: #333333;
   }
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>

<body>
<form id='Form1' action='<%#EmailURL%>' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span
class="Username"><%#locSalutation%> <strong><%#ActorName%></strong></span>,
<p class="style2"><%#locMailDescription%></p></td></tr>

<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>


<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
<%#Value.SendTo.UserDisplayName%>
<%#Value.SendTo.UserEmail%>
<%#Value.SendTo.UserId%>

Contents declared in Start Activity can also be used in a similar way.

XML Variables can also be used.

Variables defined in Start Activity can also be used pass the values to the mail.

Pre-defined variables can also be used.
A few are listed below.
<%#Variable.SubmittedBy.UserId%>
<%#Variable.SubmittedBy.UserName%>
<%#Variable.SubmittedBy.UserDisplayName%>
To make use of localization, Create a Key and its value
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key
in WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.
        TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF
CODE.
-->
    <!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF"><%#Variable.SubmittedBy.UserDepartment%></td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%"><%#Variable.SubmittedBy.UserDisplayName%></td>
</tr>
</table>
</td>
</tr>-->

<tr>
<td bgcolor="#EBEBEB">
<span class="style4"><%#loctaskInfo%></span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b><%#locName%></b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#TaskName%></td>
</tr>
<tr>
<td>
<b><%#locDescription%></b></td>
<td>
<%#Description%></td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b><%#locSubmittedBy%></b></td>
<td bgcolor="#EFEFEF">
<%#Submitedby%></td>
</tr>
<tr>
<td>
<b><%#locSubmittedOn%></b></td>
<td>
<%#SubmittedDate%> <%#SubmittedTime%></td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td bgcolor="#EFEFEF">
<b>User ID</b></td>
<td bgcolor="#EFEFEF">
<%#XmlVariables.Users.User.ID%></td>
</tr>
<tr>
<td>
<b>User Email</b></td>
<td>
<%#XmlVariables.Users.User.Email%></td>
</tr>
<tr>
<td  bgcolor="#EFEFEF">
<b>User Name</b></td>
<td  bgcolor="#EFEFEF">
<%#XmlVariables.Users.User.Name%></td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>

<tr>
<td bgcolor="#EBEBEB" class="style2"><%#locRepondInTwoWays%></td>
</tr>

<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">

<tr>
<td><strong><%#locRespondbyMail%></strong></td>
</tr>

 <tr>
<td style="padding-left:20px"><i><%#locReplyToMail%></i>
<table width='95%' border='0' align='center' cellpadding='5' cellspacing='1'
bgcolor='#EBEBEB' id="TABLE1" style="width: 100%; height: 25px;">
<tr class="style2" bgcolor='#EBEBEB'>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong><%#locColumnName%></strong></td>
<td class="style2" width='30%'  bgcolor='#EBEBEB'>
<strong><%#locUserInputs%></strong></td>
<td class="style2"  bgcolor='#EBEBEB'>
<strong><%#locValidInputs%></strong></td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'><%#locComments%>:</td>
<td class="style2" width='30%'  bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[<%#locFreeText%>]</td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>

<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'>
<%#TaskStatus%>:</td>
<td class="style2" width='30%'  bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>
[<%#FreeTextTaskStatus%>]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor="#ffffff" width="30%" >
<%#locTakeAction%>:</td>
<td class="style2" bgcolor="#ffffff" width="30%" >
[<%#locSubmit%>]</td>
<td class="style2" bgcolor="#ffffff" >
<%#locEnter%>:[<%#locSubmit%>]</td>
</tr>
 </table>
 </td>
</tr>

<tr>
<td><strong><br /><%#locRespondByWeb%></strong></td>
</tr>
<tr>
 <td style="padding-left:20px"><i><%#locWorkItemView%></i>
<br />
<a <%#workitemView%>><%#locWorkItemViewURLText%></a>
 </td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a <%#docview%>><%#locDocViewURLText%></a></span>
|
<span><a <%#executionView%>><%#locExecutionViewURLText%></a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowReminderTime:<%#Skel
taReminderTimeValue%>}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</table>
</form>
</body>
</html>
```

3. The actor will get the mail in the defined format for the above template.

Information Activity Email Template

AVEVA Work Tasks Engine accesses the Information activity's email template from the physical path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\Information.html

This template can be modified by the user as per their UI requirement. The mails received by the actor is based on the above mentioned mail template.

The following is default email template text for the **Information.html** file:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight: bold; }
-->
</style>
</head>

<body>
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span class="Username"><%#locSalutation%> <strong><%#ActorName%></strong> </span>,
<p class="style2"><%#locMailDescription%></p></td></tr>

<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style4"><%#locResponseOptions%></td>
</tr>
<tr>
<td>
<table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">
<tr>
<td><a <%#workitemView%>><%#locRespondByWeb%></a></td>
</tr>
<tr>
<td><a <%#docview%>><%#locDocViewURLText%></a></td>
</tr>
<tr>
<td ><a <%#executionView%>><%#locExecutionViewURLText%></a></td>
</tr>
</table>
</td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style4" style="height:10%"></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

Note: While modifying the Task email template the following placeholders should not be modified, as these are necessary for the engine to process the mails that are received from the actors.

1. Notes

   ```
   <td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
   ```

Use Variables in Information Email Template

Email templates can be used to display the values of the variables that have been created in a workflow.

Consider that a variable of type string by name **'Status'** is created in the **Start** activity and its value is updated as

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

"Variable, status is updated".

User wants to do the following:

1. Display the variable's value in Information email.
2. Display the variable's value after **"Submitted on"** row in the **Task Details** Table.

Solution

To display this variable's value in Information email and to display the variable's value after "Submitted on" row in the Task Details Table:

Steps:

1. Copy **Information.html** file.
2. Use the **<%#Variable.Variablename%>** placeholder in the html template for viewing the variable's value, where <Variablename> is a name of the variable.

   Since the variable name is "Status", the placeholder to be used is <%#Variable.Status%> where the value is to be displayed. The highlighted portion in the below code displays the position where the <%#Variable.Status%> placeholder is added.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
-->
</style>
</head>

<body>
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span class="Username"><%#locSalutation%> <strong><%#ActorName%></strong>
</span>,
<p class="style2"><%#locMailDescription%></p></td></tr>

<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style4"><%#locResponseOptions%></td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b>Status</b></td>
<td width="79%" bgcolor="#EFEFEF">
<%#Variable.Status%>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<td>
<table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">
<tr>
<td><a <%#workitemView%>><%#locRespondByWeb%></a></td>
</tr>
<tr>
<td><a <%#docview%>><%#locDocViewURLText%></a></td>
</tr>
<tr>
<td ><a <%#executionView%>><%#locExecutionViewURLText%></a></td>
</tr>
</table>
</td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style4" style="height:10%"></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</table>
</td>
</tr>
</table>
</body>
</html>
```

1. The actor will get the Information email in the defined format for the above template.

Note: Out of box variables such as <%#Variable.SubmittedBy.UserId%>, <%#Variable.SubmittedBy.UserName%>, and <%#Variable.SubmittedBy.UserDisplayName%> can be used for Active Directory provider.

Use Contents in Information Email Template

Email templates can be used to display the values of the contents that have been created in a workflow.

Consider that a **Content** of type number and by name 'EmployeeAge' is created in the **Start** activity and at runtime its value is 21.

User wants to do the following:

1. Display the content's value in Information email.
2. Display the variable's value after "Submitted on" row in the Task Details Table.

Solution

To display this the content's value in Information email and to display the variable's value after "Submitted on" row in the Task Details Table:

Steps:

1. Copy **Information.html** file.
2. Use the **<%#Variable.Contentname%>** placeholder in the html template for viewing the content's value, where <Contentname> is a name of the content.

   Use the **<%#Content.EmployeeAge%>** placeholder in the html template, where the value is to be displayed. Since the Content name is "EmployeeAge", the placeholder to be used is <%#Content.EmployeeAge%>. The highlighted portion in the below code displays the position where the <%#Content.EmployeeAge%> placeholder is added.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 11px;
}
.style4 {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 15px;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 font-weight: Normal;
 color: #333333;
}
.Username {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.subject {
 font-family: Arial, Helvetica, sans-serif;
 font-size: 13px;
 font-weight: Bold;
 color: #333333;
}
.style8 {
 color: #FF3300;
 font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>

<body>
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span class="Username"><%#locSalutation%> <strong><%#ActorName%></strong>
</span>,
<p class="style2"><%#locMailDescription%></p></td></tr>

<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style4"><%#locResponseOptions%></td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
 <tr width="21%" bgcolor="#EFEFEF">
 <td>
 <b>Employee Age</b></td>
 <td width="79%" bgcolor="#EFEFEF">
<%#Content.EmployeeAge%></td>
</tr>
</table>
</td>
</tr>
<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td>
<table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">
<tr>
<td><a <%#workitemView%>><%#locRespondByWeb%></a></td>
</tr>
<tr>
<td><a <%#docview%>><%#locDocViewURLText%></a></td>
</tr>
<tr>
<td ><a <%#executionView%>><%#locExecutionViewURLText%></a></td>
</tr>
</table>
</td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style4" style="height:10%"></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

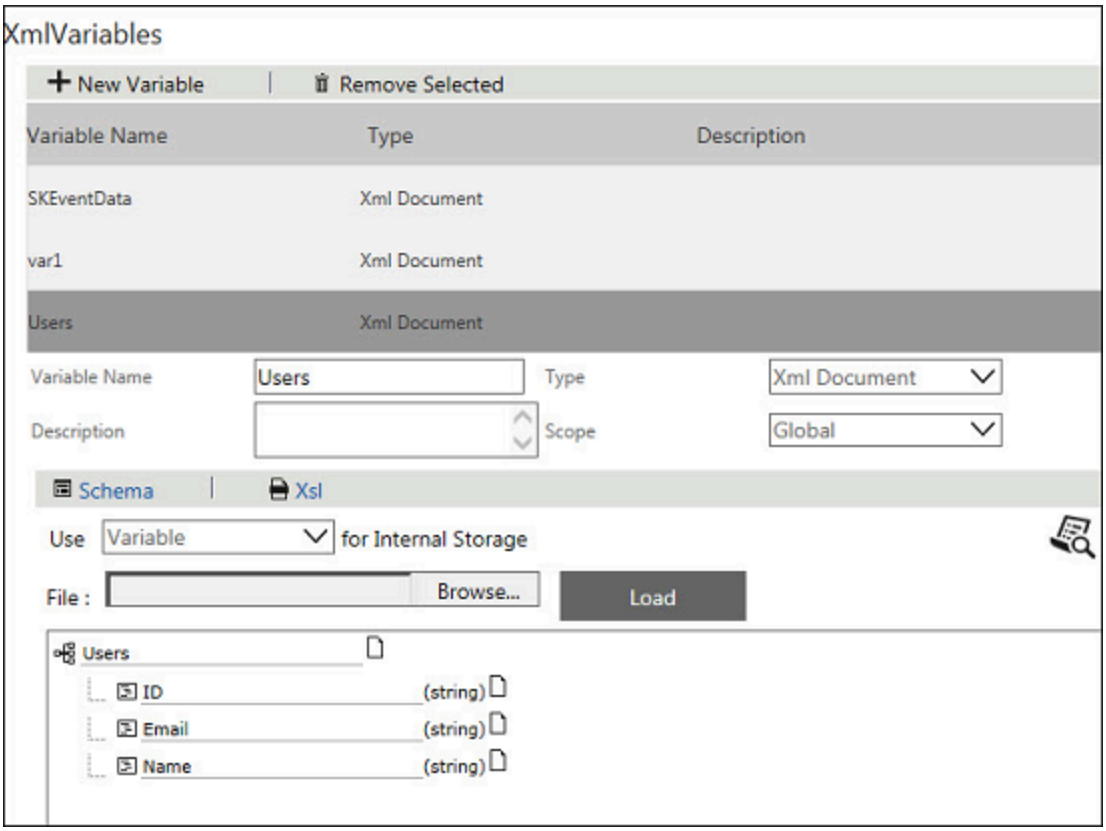
3. The actor will get the mail in the defined format for the above template.

Use XML Variables in Information Email Template

Email templates can be used to display the values of the XML variables that have been created in a workflow.

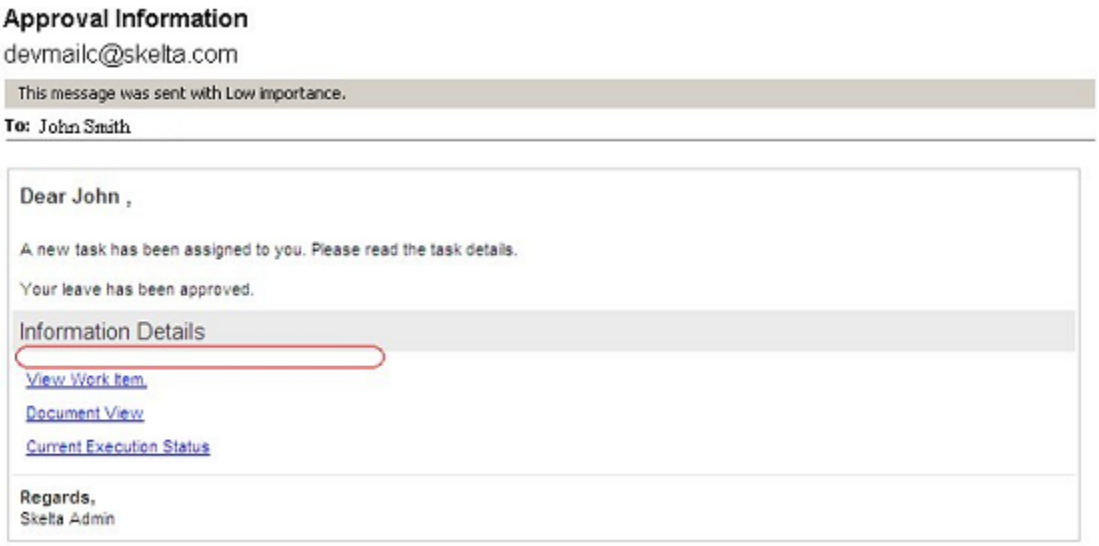
Consider that 'Users' XML Variable of type XMLDocument is created in the **Start** activity and schema is as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The scenario requirement is that during the workflow execution, the user wants to display the following values during the workflow execution:

1. **ID** is updated to 1029.
2. **Email** is updated to John.Smith@domain.com
3. **Name** is updated to John.
4. Display these XML variable's value after **"Submitted on"** row in the **Task Details** Table.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Solution

To display this the XML Variable's value in Information email and to display these variable's value after **"Submitted on"** row in the **Task Details** Table:

Steps:

1. Copy **Information.html** file.
2. Use the **<%#XmlVariables.variablename%>** placeholder in the html template for viewing the XML Variable's value, where <variablename> is a name of the XML variable.

   Since the 'Users' XML Variable contents the ID, Email and Name, use the following XML Variables in the Information.html template, where the values are to be displayed.

   ```
   <%#XmlVariables.Users.User.ID%>
   <%#XmlVariables.Users.User.Email%>
   <%#XmlVariables.Users.User.Name%>
   ```

   The highlighted portion in the below code displays the position where the above placeholders to be added.

   ```
   <html>
   <head>
   <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
   <title>Mailer</title>
   <style type="text/css">
   <!--
   .style2 {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 11px;
   }
   .style4 {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 15px;
    font-weight: Normal;
    color: #333333;
   }
   .Username {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 13px;
    font-weight: Bold;
    color: #333333;
   }
   .subject {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 13px;
    font-weight: Bold;
    color: #333333;
   }
   .style8 {
    color: #FF3300;
    font-weight: bold;
   }
   .style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
   bold; }
   -->
   </style>
   </head>
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<body>
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">

<tr>
<td><span class="Username"><%#locSalutation%> <strong><%#ActorName%></strong>
</span>,
<p class="style2"><%#locMailDescription%></p></td></tr>

<tr>
<td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style4"><%#locResponseOptions%></td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b>User ID</b>
</td>
<td width="79%" bgcolor="#EFEFEF">
<%#XmlVariables.Users.User.ID%>
</td>
</tr>
<tr>
<td>
<b>User Email</b>
</td>
<td>
<%#XmlVariables.Users.User.Email%>
</td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>User Name</b>
</td>
<td bgcolor="#EFEFEF">
<%#XmlVariables.Users.User.Name%>
</td>
</tr>
</table>
</td>
</tr>
<tr>
<tr>
<td>
<table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">
<tr>
<td><a <%#workitemView%>><%#locRespondByWeb%></a></td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td><a <%#docview%>><%#locDocViewURLText%></a></td>
</tr>
<tr>
<td ><a <%#executionView%>><%#locExecutionViewURLText%></a></td>
</tr>
</table>
</td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style4" style="height:10%"></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2"><%#locSignature%></td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

1. The actor will get the mail in the defined format for the above template.

Different Email Template for Different Activity in a Workflow

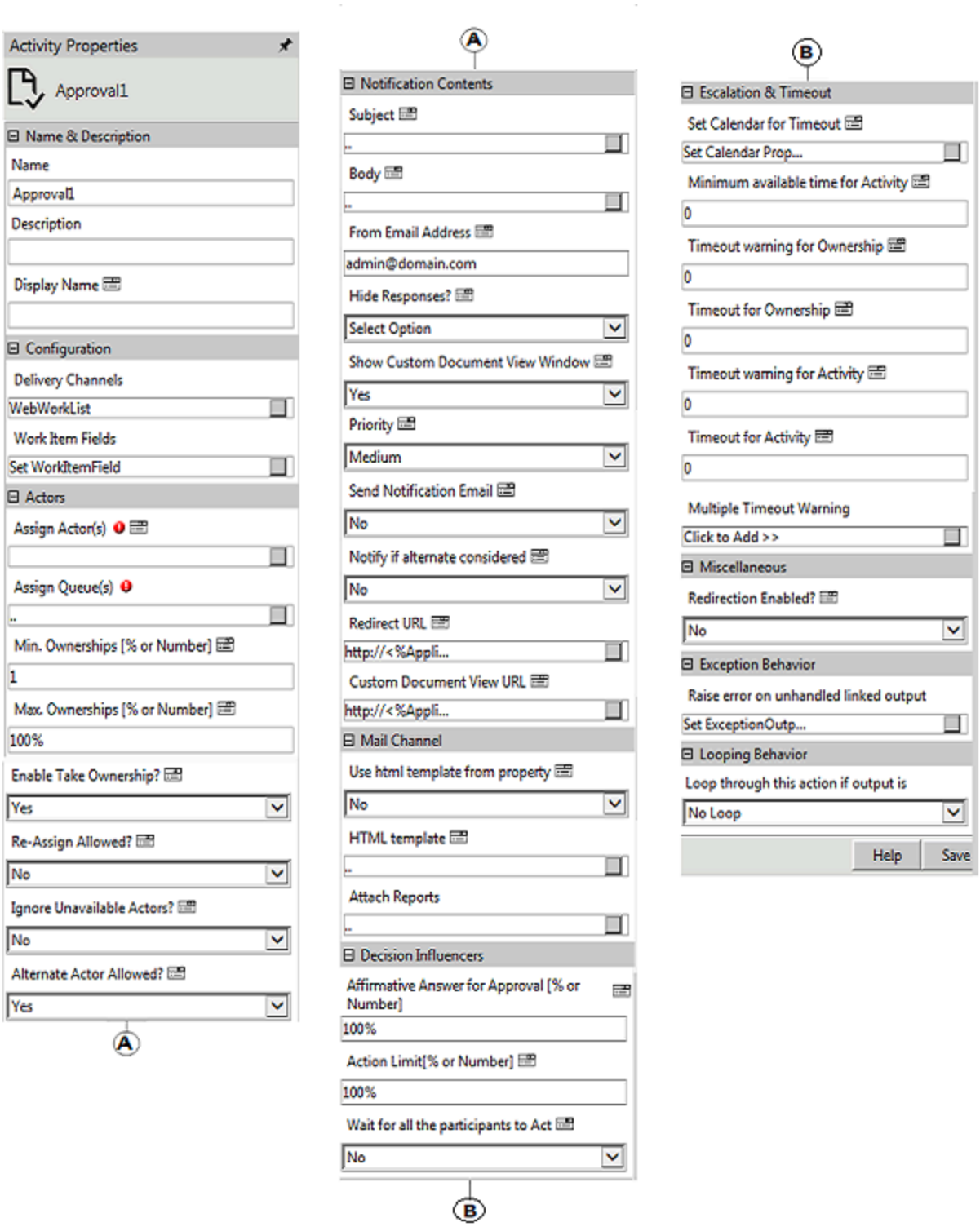
Each Human activity like Approval, Choice, Checklist and Task Activity can have its own email template that will be sent to an actor.

Example:

Let us assume that a workflow has two **Approval** activities and the mail format sent from Approval Activity1 is different from that of Approval Activity2.

To achieve the same, AVEVA Work Tasks introduced a category **Mail Channel** in the Activity Properties of the Human activity, to have different formats for different activities.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Use 'HTML Template' Property of Approval Activity

You can use the html email template that is present in the physical folder *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks*\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\ ApprovalEmail.html in the 'HTML Template' property of Approval activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Complete the following steps to customize the ApprovalEmail.html.

1. Right-click the Approval activity and then select Activity Properties.
2. Under Mail Channel group, set the Use html template from property to Yes.
3. Under Mail Channel group, click the HTML template edit icon. The HTML template window is displayed.
4. Go to *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks*\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\ApprovalEmail.html path and take a copy of the default html template file of approval activity and make the necessary changes to the copied file.

   Note: While modifying the Approval email template the following place holders should not be modified, as these are necessary for the engine to process the mails that are received from the actors.

   a. Email URL

   ```
   <form id='Form1' action='<%#EmailURL%>' method='post' >
   ```

   b. Notes

   ```
   <td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
   ```

   c. Complete content

   ```
   <table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
   cellspacing='1' style="width: 100%; height: 25px;">
   <tr>
   <td class="style2" width='30%' bgcolor='#EBEBEB'>
   <strong><%#locColumnName%></strong></td>
   <td class="style2" width='30%' bgcolor='#EBEBEB'>
   <strong><%#locUserInputs%></strong></td>
   <td class="style2" bgcolor='#EBEBEB'>
   <strong><%#locValidInputs%></strong></td>
   </tr>
   <tr valign="top">
   <td class="style2" bgcolor='#FFFFFF' width="10%"><%#locComments%>:</td>
   <td class="style2" bgcolor='#FFFFFF'
   width="10%">[     ]</td>
   <td class="style2" bgcolor='#FFFFFF' width="20%" >[<%#locFreeText%>]</td>
   </tr>
   <tr valign="top">
   <td class="style2" bgcolor='#FFFFFF' width="10%"><%#locTakeAction%>:</td>
   <td class="style2" bgcolor='#FFFFFF'
   width="10%">[     ]</td>
   <td class="style2" bgcolor='#FFFFFF' width="20%"
   ><%#locEnter%>:[<%#locApproveAction%>,<%#locRejectAction%>]</td>
   </tr>
   </table>
   ```

   d. Complete content

   ```
   <td class="style2" width="95%"><span
   class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
   Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResou
   rceActivityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowRemind
   erTime:<%#SkeltaReminderTimeValue%>}</td>
   ```

5. Replace only the following placeholders in the copied html template as per the table below. This is because the place holders which is in ' <%# %>' format will not support in customized html format.

   Note: Make sure that only the following specified place holders are replaced. Do not replace any other place holders.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Keyword	Replace with
<%#EmailURL%>	[%#EmailURL%]
<%#locSalutation%>	[%#locSalutation%]
<%#ActorName%>	[%#ActorName%]
<%#locMailDescription%>	[%#locMailDescription%]
<%#Notes%>	[%#Notes%]
<%#loctaskInfo%>	[%#loctaskInfo%]
<%#locName%>	[%#locName%]
<%#TaskName%>	[%#TaskName%]
<%#locDescription%>	[%#locDescription%]
<%#Description%>	[%#Description%]
<%#locSubmittedBy%>	[%#locSubmittedBy%]
<%#Submitedby%>	[%#Submitedby%]
<%#locSubmittedOn%>	[%#locSubmittedOn%]
<%#SubmittedDate%>	[%#SubmittedDate%]
<%#SubmittedTime %>	[%#SubmittedTime %]
<%#locRepondInTwoWays%>	[%#locRepondInTwoWays%]
<%#locRespondbyMail%>	[%#locRespondbyMail%]
<%#locReplyToMail%>	[%#locReplyToMail%]
<%#locColumnName%>	[%#locColumnName%]
<%#locUserInputs%>	[%#locUserInputs%]
<%#locValidInputs%>	[%#locValidInputs%]
<%#locComments%>	[%#locComments%]
<%#locFreeText%>	[%#locFreeText%]
<%#locTakeAction%>	[%#locTakeAction%]
<%#locEnter%>	[%#locEnter%]
<%#locApproveAction%>	[%#locApproveAction%]

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Keyword	Replace with
<%#locRejectAction%>	[%#locRejectAction%]
<%#locRespondByWeb%>	[%#locRespondByWeb%]
<%#locWorkItemView%>	[%#locWorkItemView%]
<%#workitemView%>	[%#workitemView%]
<%#locWorkItemViewURLText%>	[%#locWorkItemViewURLText%]
<%#docview%>	[%#docview%]
<%#locDocViewURLText%>	[%#locDocViewURLText%]
<%#executionView%>	[%#executionView%]
<%#locExecutionViewURLText%>	[%#locExecutionViewURLText%]
<%#locSignature%>	[%#locSignature%]
<%#locEmailSecurityWarning%>	[%#locEmailSecurityWarning%]
<%#ApplicationName%>	[%#ApplicationName%]
<%#EmailGUID%>	[%#EmailGUID%]
<%#WorkflowName%>	[%#WorkflowName%]
<%#SkeltaFollowUpValue%>	[%#SkeltaFollowUpValue%]
<%#SkeltaReminderTimeValue%>	[%#SkeltaReminderTimeValue%]

6. After replacing place holders, the code is as shown below.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
font-family: Arial, Helvetica, sans-serif;
font-size: 11px;
}
.style4 {
font-family: Arial, Helvetica, sans-serif;
font-size: 15px;
font-weight: Normal;
color: #333333;
}
.Username {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
font-weight: Bold;
color: #333333;
}
.subject {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: Bold;
color: #333333;
}
.style8 {
color: #FF3300;
font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>
<body>
<form id='Form1' action='[%#EmailURL%]' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><span
class="Username">[%#locSalutation%] <strong>[%#ActorName%]</strong></span>,
<p class="style2">[%#locMailDescription%]</p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2">[%#Notes%] </td>
</tr>
<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
[%#Value.SendTo.UserDisplayName%]
[%#Value.SendTo.UserEmail%]
[%#Value.SendTo.UserId%]
Contents declared in Start Activity can also be used in a similar way.
XML Variables can also be used.
Variables defined in Start Activity can also be used pass the values to the mail.
Pre-defined variables can also be used.
A few are listed below.
[%#Variable.SubmittedBy.UserId%]
[%#Variable.SubmittedBy.UserName%]
[%#Variable.SubmittedBy.UserDisplayName%]
To make use of localization, Create a Key and its value in
WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in
WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.
TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF">[%#Variable.SubmittedBy.UserDepartment%]</td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%">[%#Variable.SubmittedBy.UserDisplayName%]</td>
</tr>
</table>
</td>
</tr>-->
<tr>
<td bgcolor="#EBEBEB">
<span class="style4">[%#loctaskInfo%]</span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b>[%#locName%]</b></td>
<td width="79%" bgcolor="#EFEFEF">
[%#TaskName%]</td>
</tr>
<tr>
<td>
<b>[%#locDescription%]</b></td>
<td>
[%#Description%]</td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>[%#locSubmittedBy%]</b></td>
<td bgcolor="#EFEFEF">
[%#Submitedby%]</td>
</tr>
<tr>
<td>
<b>[%#locSubmittedOn%]</b></td>
<td>
[%#SubmittedDate%] [%#SubmittedTime%]</td>
</tr>
</table>
</td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

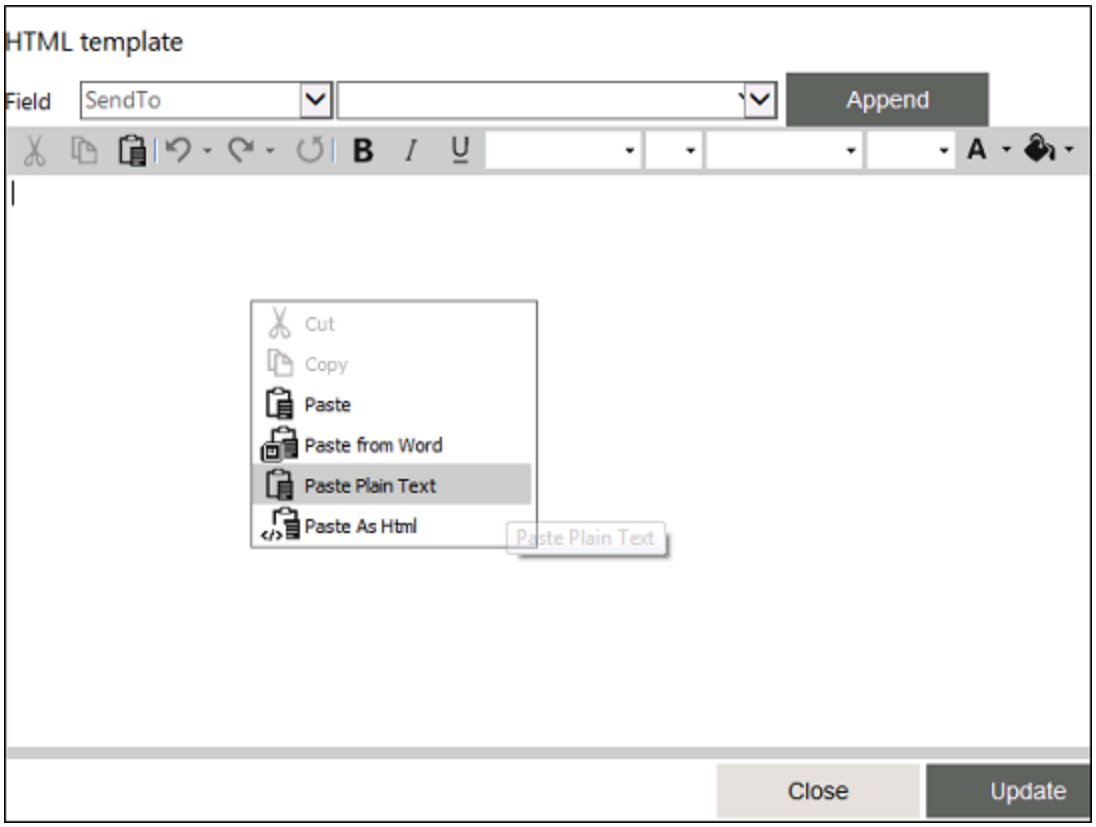
```
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style2">[%#locRepondInTwoWays%]</td>
</tr>
<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">
<tr>
<td><strong>[%#locRespondbyMail%]</strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i>[%#locReplyToMail%]</i>
<table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
cellspacing='1' style="width: 100%; height: 25px;">
<tr>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong>[%#locColumnName%]</strong></td>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong>[%#locUserInputs%]</strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong>[%#locValidInputs%]</strong></td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%">[%#locComments%]:</td>
<td class="style2" bgcolor='#FFFFFF'
width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%" >[[%#locFreeText%]]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor='#FFFFFF' width="10%">[%#locTakeAction%]:</td>
<td class="style2" bgcolor='#FFFFFF'
width="10%">[     ]</td>
<td class="style2" bgcolor='#FFFFFF' width="20%"
>[%#locEnter%]:[[%#locApproveAction%],[%#locRejectAction%]]</td>
</tr>
</table>
</td>
</tr>
<tr>
<td><strong><br />[%#locRespondByWeb%]</strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i>[%#locWorkItemView%]</i>
<br />
<a [%#workitemView%]>[%#locWorkItemViewURLText%]</a>
</td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

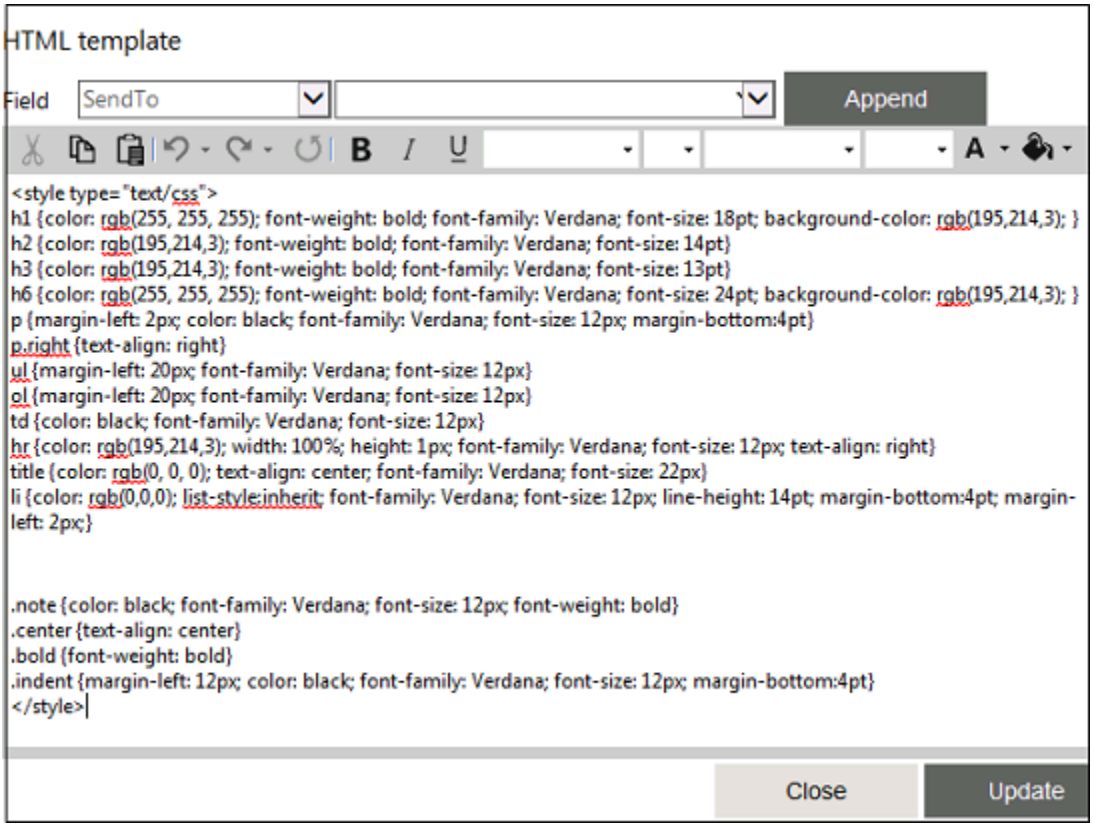
```
<td align="right" class="style2">
<span><a [%#docview%]>[%#locDocViewURLText%]</a></span>
|
<span><a [%#executionView%]>[%#locExecutionViewURLText%]</a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2">[%#locSignature%]</td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong>[%#locEmailSecurityWarning%]</strong></span><br />
Workflow:{[%#ApplicationName%]:[%#EmailGUID%]:[%#WorkflowName%]}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:[%#SkeltaFollowUpValue%]}{WorkflowReminderTime:[%#Skel
taReminderTimeValue%]}</td>
</tr>
</table></td>
</tr>
</table>
</td>
</tr>
</table>
</form>
</body>
</html>
```

7. Copy the modified code (In this case, the above code) and paste into the HTML template window as Paste Plain Text and click Update.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



8. The following code shows after pasting the code



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Variables, Contents, and XML Variables can also be inserted into the html text wherever necessary by just selecting it from the drop-down provided in the HTML template window and appending it wherever necessary.

Use 'HTML Template' Property of Choice Activity

You can use the html email template that is present in the physical folder *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks*\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\Choice.html in the 'HTML Template' property of Choice activity.

Complete the following steps to customize the Choice.html.

1. Right-click the Choice activity and select Activity Properties.
2. Under Mail Channel group, set the Use html template from property to Yes.
3. Under Mail Channel group, click the HTML template edit icon.
4. Go to *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks*\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\Choice.html path and take a copy of the default html template file of Choice activity and make the necessary changes to the copied file.

   Note: While modifying the Choice email template the following place holders should not be modified as these are necessary for the engine to process the mails that are received from the actors.

   a. Email URL

   ```
   <form id='Form1' action='<%#EmailURL%>' method='post' >
   ```

   b. Notes

   ```
   <td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
   ```

   c. Complete content

   ```
   <table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
   cellspacing='1' style="width: 100%; height: 25px;">
   <tr>
   <td class="style2" width='30%' bgcolor='#EBEBEB'>
   <strong><%#locColumnName%></strong></td>
   <td class="style2" width='30%' bgcolor='#EBEBEB'>
   <strong><%#locUserInputs%></strong></td>
   <td class="style2" bgcolor='#EBEBEB'>
   <strong><%#locValidInputs%></strong></td>
   </tr>
   <tr valign="top">
   <td class="style2" width='30%' bgcolor='#FFFFFF'><%#locComments%>:</td>
   <td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
   <td class="style2" bgcolor='#FFFFFF'>[<%#locFreeText%>]</td>
   </tr>
   <tr valign="top">
   <td class="style2" width='30%' bgcolor='#FFFFFF'><%#locTakeAction%>:</td>
   <td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
   <td class="style2" bgcolor='#FFFFFF'><%#locEnter%>:[<%#action%>]</td>
   </tr>
   </table>
   ```

   d. Complete content

   ```
   <td class="style2" width="95%"><span
   class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br />
   Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowResou
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
rceActivityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowRemind
erTime:<%#SkeltaReminderTimeValue%>}</td>
```

5. Replace only the following place holders in the copied html template as per the table below. This is because the place holders which is in ' <%# %>' format will not support in customized html format.

Note: Make sure that only the following specified place holders are replaced. Do not replace any other place holders.

Keyword	Replace with
<%#EmailURL%>	[%#EmailURL%]
<%#locSalutation%>	[%#locSalutation%]
<%#ActorName%>	[%#ActorName%]
<%#locMailDescription%>	[%#locMailDescription%]
<%#Notes%>	[%#Notes%]
<%#loctaskInfo%>	[%#loctaskInfo%]
<%#locName%>	[%#locName%]
<%#TaskName%>	[%#TaskName%]
<%#locDescription%>	[%#locDescription%]
<%#Description%>	[%#Description%]
<%#locSubmittedBy%>	[%#locSubmittedBy%]
<%#Submitedby%>	[%#Submitedby%]
<%#locSubmittedOn%>	[%#locSubmittedOn%]
<%#SubmittedDate%>	[%#SubmittedDate%]
<%#SubmittedTime %>	[%#SubmittedTime %]
<%#locRepondInTwoWays%>	[%#locRepondInTwoWays%]
<%#locRespondbyMail%>	[%#locRespondbyMail%]
<%#locReplyToMail%>	[%#locReplyToMail%]
<%#locColumnName%>	[%#locColumnName%]
<%#locUserInputs%>	[%#locUserInputs%]
<%#locValidInputs%>	[%#locValidInputs%]
<%#locComments%>	[%#locComments%]

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Keyword	Replace with
<%#locFreeText%>	[%#locFreeText%]
<%#locTakeAction%>	[%#locTakeAction%]
<%#locEnter%>	[%#locEnter%]
<%#action%>	[%#action%]
<%#locRespondByWeb%>	[%#locRespondByWeb%]
<%#locWorkItemView%>	[%#locWorkItemView%]
<%#workitemView%>	[%#workitemView%]
<%#locWorkItemViewURLText%>	[%#locWorkItemViewURLText%]
<%#docview%>	[%#docview%]
<%#locDocViewURLText%>	[%#locDocViewURLText%]
<%#executionView%>	[%#executionView%]
<%#locExecutionViewURLText%>	[%#locExecutionViewURLText%]
<%#locSignature%>	[%#locSignature%]
<%#locEmailSecurityWarning%>	[%#locEmailSecurityWarning%]
<%#ApplicationName%>	[%#ApplicationName%]
<%#EmailGUID%>	[%#EmailGUID%]
<%#WorkflowName%>	[%#WorkflowName%]
<%#SkeltaFollowUpValue%>	[%#SkeltaFollowUpValue%]
<%#SkeltaReminderTimeValue%>	[%#SkeltaReminderTimeValue%]

6. After replacing place holders, the code is as shown below.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
font-family: Arial, Helvetica, sans-serif;
font-size: 11px;
}
.style4 {
font-family: Arial, Helvetica, sans-serif;
font-size: 15px;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
font-weight: Normal;
color: #333333;
}
.Username {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: Bold;
color: #333333;
}
.subject {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: Bold;
color: #333333;
}
.style8 {
color: #FF3300;
font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>
<body>
<form id='Form1' action='[%#EmailURL%]' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><span
class="Username">[%#locSalutation%] <strong>[%#ActorName%]</strong></span>,
<p class="style2">[%#locMailDescription%]</p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2">[%#Notes%] </td>
</tr>
<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
[%#Value.SendTo.UserDisplayName%]
[%#Value.SendTo.UserEmail%]
[%#Value.SendTo.UserId%]
Contents declared in Start Activity can also be used in a similar way.
XML Variables can also be used.
Variables defined in Start Activity can also be used pass the values to the mail.
Pre-defined variables can also be used.
A few are listed below.
[%#Variable.SubmittedBy.UserId%]
[%#Variable.SubmittedBy.UserName%]
[%#Variable.SubmittedBy.UserDisplayName%]
To make use of localization, Create a Key and its value in
WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in
WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.
TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF">[%#Variable.SubmittedBy.UserDepartment%]</td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%">[%#Variable.SubmittedBy.UserDisplayName%]</td>
</tr>
</table>
</td>
</tr>-->
<tr>
<td bgcolor="#EBEBEB">
<span class="style4">[%#loctaskInfo%]</span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b>[%#locName%]</b></td>
<td width="79%" bgcolor="#EFEFEF">
[%#TaskName%]</td>
</tr>
<tr>
<td>
<b>[%#locDescription%]</b></td>
<td>
[%#Description%]</td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>[%#locSubmittedBy%]</b></td>
<td bgcolor="#EFEFEF">
[%#Submitedby%]</td>
</tr>
<tr>
<td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

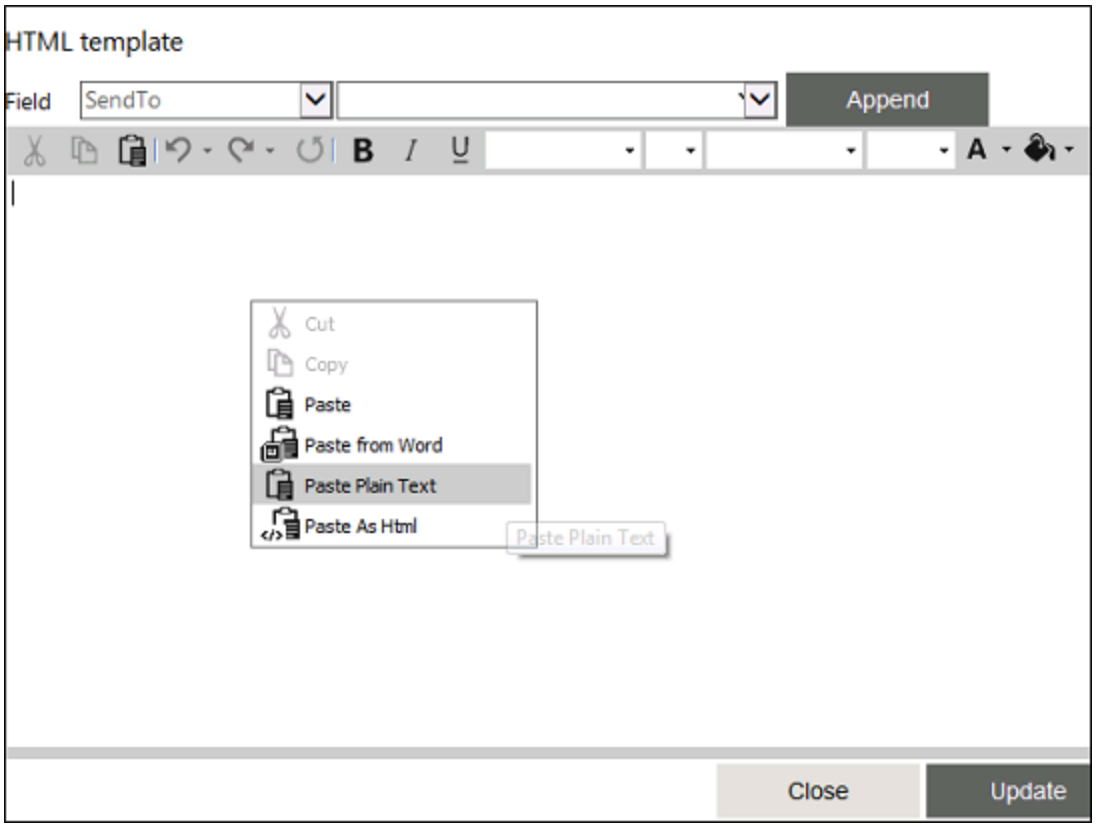
```
<b>[%#locSubmittedOn%]</b></td>
<td>
[%#SubmittedDate%] [%#SubmittedTime%]</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style2">[%#locRepondInTwoWays%]</td>
</tr>
<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">
<tr>
<td><strong>[%#locRespondbyMail%]</strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i>[%#locReplyToMail%]</i>
<table width='95%' border='0' align='center' bgcolor='#EBEBEB' cellpadding='5'
cellspacing='1' style="width: 100%; height: 25px;">
<tr>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong>[%#locColumnName%]</strong></td>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong>[%#locUserInputs%]</strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong>[%#locValidInputs%]</strong></td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'>[%#locComments%]:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[[%#locFreeText%]]</td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'>[%#locTakeAction%]:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[%#locEnter%]:[[%#action%]]</td>
</tr>
</table>
</td>
</tr>
<tr>
<td><strong><br />[%#locRespondByWeb%]</strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i>[%#locWorkItemView%]</i>
<br />
<a [%#workitemView%]>[%#locWorkItemViewURLText%]</a>
</td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

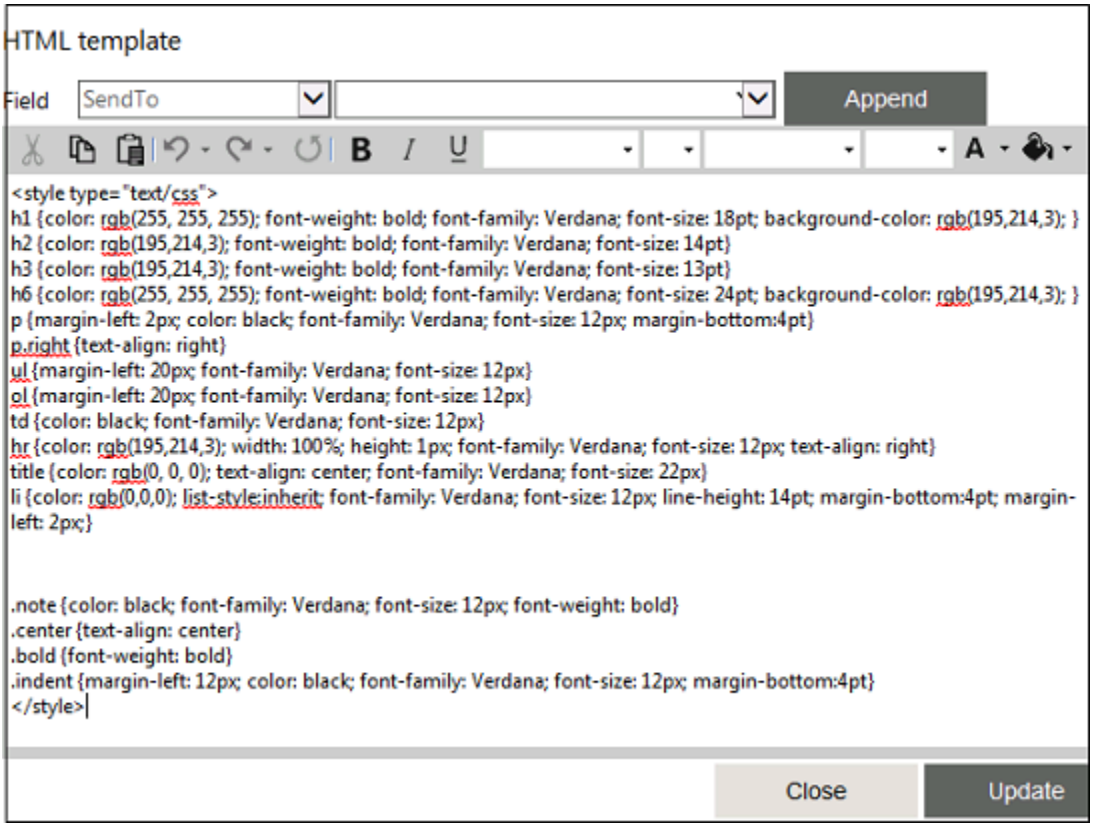
```
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a [%#docview%]>[%#locDocViewURLText%]</a></span>
|
<span><a [%#executionView%]>[%#locExecutionViewURLText%]</a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2">[%#locSignature%]</td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong>[%#locEmailSecurityWarning%]</strong></span><br />
Workflow:{[%#ApplicationName%]:[%#EmailGUID%]:[%#WorkflowName%]}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:[%#SkeltaFollowUpValue%]}{WorkflowReminderTime:[%#Skel
taReminderTimeValue%]}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>
```

7. Copy the modified code (In this case, the above code) and paste into the HTML template window as Paste Plain Text and click Update.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



8. The following code shows after pasting the code.



Variables, Contents, and XML Variables can also be inserted into the html text wherever necessary by just

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

selecting it from the drop-down provided in the HTML template window and appending it wherever necessary.

Use 'HTML Template' Property of Checklist Activity

You can use the html email template that is present in the physical folder *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks*\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\CheckListEmail.html in the 'HTML Template' property of CheckList activity.

Complete the following steps to customize the CheckListEmail.html.

1. Right-click the Check-List activity and then select Activity Properties.
2. Under Mail Channel group, set the Use html template from property to Yes.
3. Under Mail Channel group, click HTML template edit icon.
4. Go to *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks*\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\CheckListEmail.html path and take a copy of the default html template file of Check-List activity and make the necessary changes to the copied file.

   Note: While modifying the Check-List email template the following place holders should not be modified as these are necessary for the engine to process the mails that are received from the actors.

   a. Email URL

   ```
   <form id='Form1' action='<%#EmailURL%>' method='post'>
   ```

   b. Notes

   ```
   <td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
   ```

   c. Complete content

   ```
   <table border='0' align='center' cellpadding='5' cellspacing='1'
   bgcolor='#EBEBEB' class="style2" style="width: 100%; height: 25px;">
   <tr bgcolor='#EBEBEB' class="style2">
   <td bgcolor='#EBEBEB' class="style2"><strong><%#locColumnName%></strong></td>
   <td bgcolor='#EBEBEB' class="style2"><strong><%#locUserInputs%></strong></td>
   <td bgcolor='#EBEBEB' class="style2"><strong><%#locValidInputs%></strong></td>
   </tr> <%#InputPlaceHolderString%>
   </table>
   ```

   d. Complete content

   ```
   <td width="95%" class="style2"><span
   class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br
   />Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowRes
   ourceActivityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowRemi
   nderTime:<%#SkeltaReminderTimeValue%>}</td>
   ```

5. Replace only the following place holders in the copied html template as per the table below. This is because the place holders which is in ' <%# %>' format will not support in customized html format.

   Note: Make sure that only the following specified place holders are replaced. Do not replace any other place holders.

Keyword	Replace with
<%#EmailURL%>	[%#EmailURL%]
<%#locSalutation%>	[%#locSalutation%]

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Keyword	Replace with
<%#ActorName%>	[%#ActorName%]
<%#locMailDescription%>	[%#locMailDescription%]
<%#Notes%>	[%#Notes%]
<%#loctaskInfo%>	[%#loctaskInfo%]
<%#locName%>	[%#locName%]
<%#TaskName%>	[%#TaskName%]
<%#locDescription%>	[%#locDescription%]
<%#Description%>	[%#Description%]
<%#locSubmittedBy%>	[%#locSubmittedBy%]
<%#Submitedby%>	[%#Submitedby%]
<%#locSubmittedOn%>	[%#locSubmittedOn%]
<%#SubmittedDate%>	[%#SubmittedDate%]
<%#SubmittedTime%>	[%#SubmittedTime%]
<%#locRepondInTwoWays%>	[%#locRepondInTwoWays%]
<%#locRespondbyMail%>	[%#locRespondbyMail%]
<%#locReplyToMail%>	[%#locReplyToMail%]
<%#locColumnName%>	[%#locColumnName %]
<%#locUserInputs%>	[%#locUserInputs %]
<%#locValidInputs%>	[%#locValidInputs %]
<%#InputPlaceHolderString%>	[%#InputPlaceHolderString%]
<%#locRespondByWeb%>	[%#locRespondByWeb%]
<%#locWorkItemView%>	[%#locWorkItemView%]
<%#workitemView%>	[%#workitemView%]
<%#locWorkItemViewURLText%>	[%#locWorkItemViewURLText%]
<%#docview%>	[%#docview%]
<%#locDocViewURLText%>	[%#locDocViewURLText%]

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Keyword	Replace with
<%#executionView%>	[%#executionView%]
<%#locExecutionViewURLText%>	[%#locExecutionViewURLText%]
<%#locSignature%>	[%#locSignature%]
<%#locEmailSecurityWarning%>	[%#locEmailSecurityWarning%]
<%#ApplicationName%>	[%#ApplicationName%]
<%#EmailGUID%>	[%#EmailGUID%]
<%#WorkflowName%>	[%#WorkflowName%]
<%#SkeltaFollowUpValue%>	[%#SkeltaFollowUpValue%]
<%#SkeltaReminderTimeValue%>	[%#SkeltaReminderTimeValue%]

6. After replacing place holders, the code is as shown below.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
font-family: Arial, Helvetica, sans-serif;
font-size: 11px;
}
.style4 {
font-family: Arial, Helvetica, sans-serif;
font-size: 15px;
font-weight: Normal;
color: #333333;
}
.Username {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: Bold;
color: #333333;
}
.subject {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: Bold;
color: #333333;
}
.style8 {
color: #FF3300;
font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
-->
</style>
</head>
<body>
<form id='Form1' action='[%#EmailURL%]' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><span
class="Username">[%#locSalutation%] <strong>[%#ActorName%]</strong></span>,
<p class="style2">[%#locMailDescription%]</p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2">[%#Notes%] </td>
</tr>
<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
[%#Value.SendTo.UserDisplayName%]
[%#Value.SendTo.UserEmail%]
[%#Value.SendTo.UserId%]
Contents declared in Start Activity can also be used in a similar way.
XML Variables can also be used.
Variables defined in Start Activity can also be used pass the values to the mail.
Pre-defined variables can also be used.
A few are listed below.
[%#Variable.SubmittedBy.UserId%]
[%#Variable.SubmittedBy.UserName%]
[%#Variable.SubmittedBy.UserDisplayName%]
To make use of localization, Create a Key and its value in
WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in
WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.
TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF">[%#Variable.SubmittedBy.UserDepartment%]</td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%">[%#Variable.SubmittedBy.UserDisplayName%]</td>
</tr>
</table>
</td>
</tr>-->
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

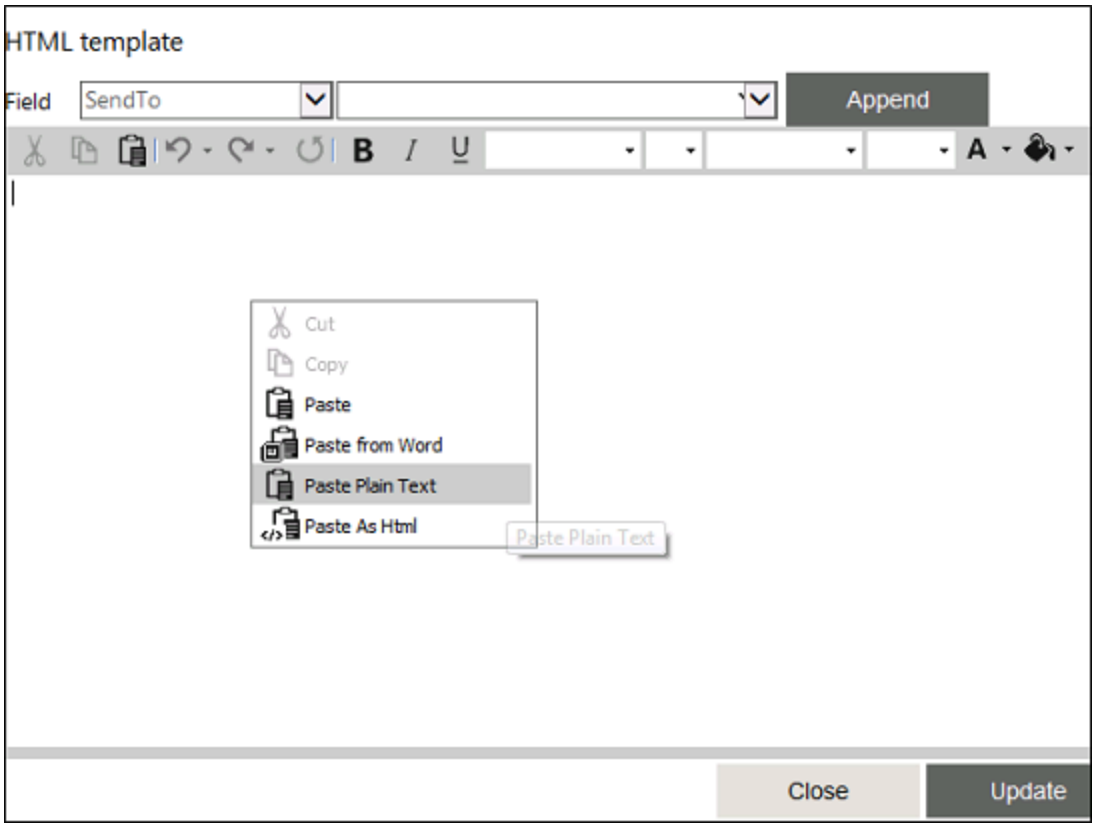
```
<tr>
<td bgcolor="#EBEBEB">
<span class="style4">[%#loctaskInfo%]</span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b>[%#locName%]</b></td>
<td width="79%" bgcolor="#EFEFEF">
[%#TaskName%]</td>
</tr>
<tr>
<td>
<b>[%#locDescription%]</b></td>
<td>
[%#Description%]</td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>[%#locSubmittedBy%]</b></td>
<td bgcolor="#EFEFEF">
[%#Submitedby%]</td>
</tr>
<tr>
<td>
<b>[%#locSubmittedOn%]</b></td>
<td>
[%#SubmittedDate%] [%#SubmittedTime%]</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style2">[%#locRepondInTwoWays%]</td>
</tr>
<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">
<tr>
<td><strong>[%#locRespondbyMail%]</strong></td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

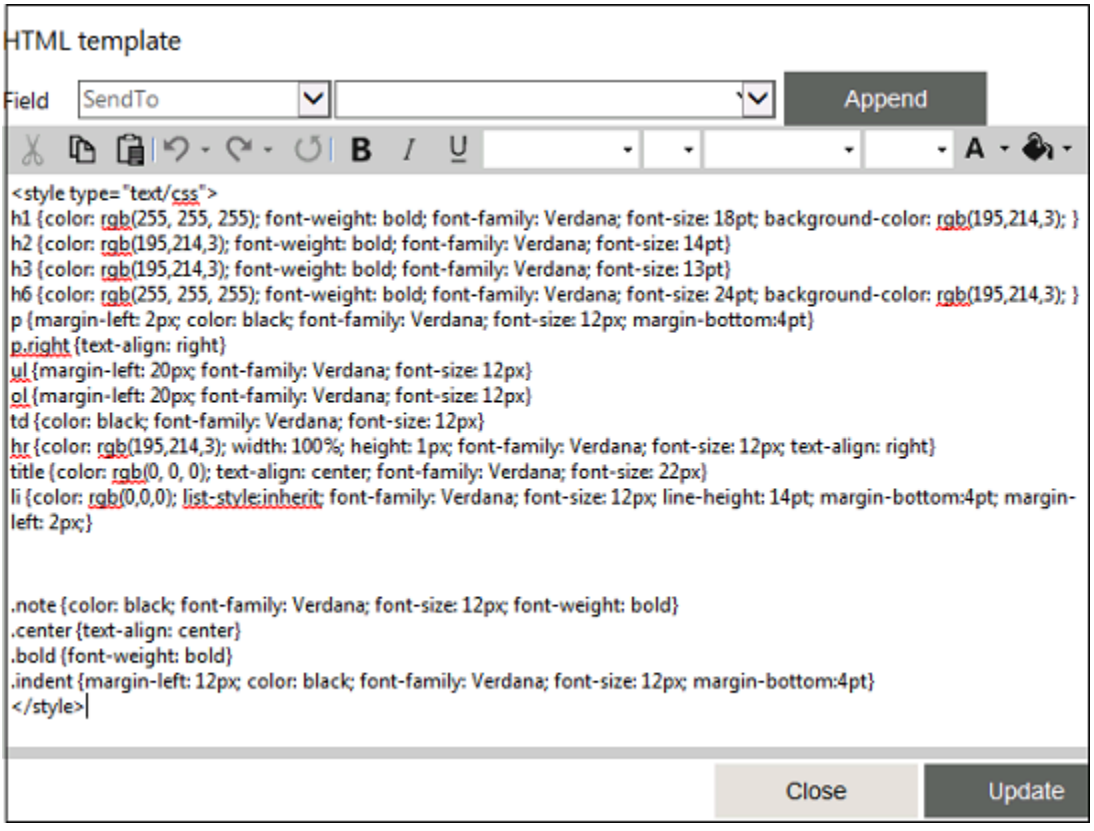
```
</tr>
<tr>
<td style="padding-left:20px"><i>[%#locReplyToMail%]</i>
<table border='0' align='center' cellpadding='5' cellspacing='1' bgcolor='#EBEBEB'
class="style2" style="width: 100%; height: 25px;">
<tr bgcolor='#EBEBEB' class="style2">
<td bgcolor='#EBEBEB' class="style2"><strong>[%#locColumnName%]</strong></td>
<td bgcolor='#EBEBEB' class="style2"><strong>[%#locUserInputs%]</strong></td>
<td bgcolor='#EBEBEB' class="style2"><strong>[%#locValidInputs%]</strong></td>
</tr> [%#InputPlaceHolderString%]
</table>
</td>
</tr>
<tr>
<td><strong><br />[%#locRespondByWeb%]</strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i>[%#locWorkItemView%]</i>
<br />
<a [%#workitemView%]>[%#locWorkItemViewURLText%]</a>
</td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a [%#docview%]>[%#locDocViewURLText%]</a></span>
|
<span><a [%#executionView%]>[%#locExecutionViewURLText%]</a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2">[%#locSignature%]</td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td width="95%" class="style2"><span
class="style8"><strong>[%#locEmailSecurityWarning%]</strong></span><br />
Workflow:{[%#ApplicationName%]:[%#EmailGUID%]:[%#WorkflowName%]}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:[%#SkeltaFollowUpValue%]}{WorkflowReminderTime:[%#Skel
taReminderTimeValue%]}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>
```

7. Copy the modified code (In this case, the above code) and paste into the HTML template window as Paste Plain Text and click Update.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



8. The following code shows after pasting the code.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Variables, Contents, and XML Variables can also be inserted into the html text wherever necessary by just selecting it from the drop-down provided in the HTML template window and appending it wherever necessary.

Use 'HTML Template' Property of Task Activity

You can use the html email template that is present in the physical folder *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks*\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\TaskEmail.html in the 'HTML Template' property of CheckList activity.

Complete the following steps to customize the the Task.html:

1. Right-click Task activity and then select Activity Properties.
2. Under Mail Channel group, set the Use html template from property to Yes.
3. Under Mail Channel group, click HTML template edit icon.
4. Go to *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks*\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\TaskEmail.html path and take a copy of the default html template file of Task activity and make the necessary changes to the copied file.

   Note: While modifying the Task email template the following place holders should not be modified as these are necessary for the engine to process the mails that are received from the actors.

   a. Email URL

   ```
   <form id='Form1' action='<%#EmailURL%>' method='post' >
   ```

   b. Notes

   ```
   <td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
   ```

   c. Complete content

   ```
   <table width='95%' border='0' align='center' cellpadding='5' cellspacing='1'
   bgcolor='#EBEBEB' id="TABLE1" style="width: 100%; height: 25px;">
   <tr class="style2" bgcolor='#EBEBEB'>
   <td class="style2" width='30%' bgcolor='#EBEBEB'>
   <strong><%#locColumnName%></strong></td>
   <td class="style2" width='30%' bgcolor='#EBEBEB'>
   <strong><%#locUserInputs%></strong></td>
   <td class="style2" bgcolor='#EBEBEB'>
   <strong><%#locValidInputs%></strong></td>
   </tr>
   <tr valign="top">
   <td class="style2" width='30%' bgcolor='#FFFFFF'><%#locComments%>:</td>
   <td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
   <td class="style2" bgcolor='#FFFFFF'>[<%#locFreeText%>]</td>
   </tr>
   <tr valign="top">
   <td class="style2" width='30%' bgcolor='#FFFFFF'>
   <%#TaskStatus%>:</td>
   <td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
   <td class="style2" bgcolor='#FFFFFF'>
   [<%#FreeTextTaskStatus%>]</td>
   </tr>
   <tr valign="top">
   <td class="style2" bgcolor="#ffffff" width="30%" >
   <%#locTakeAction%>:</td>
   <td class="style2" bgcolor="#ffffff" width="30%" >
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
[<%#locSubmit%>]</td>
<td class="style2" bgcolor="#ffffff" >
<%#locEnter%>:[<%#locSubmit%>]</td>
</tr>
</table>
```

d. Complete content

```
<td class="style2" width="95%"><span
class="style8"><strong><%#locEmailSecurityWarning%></strong></span><br
/>Workflow:{<%#ApplicationName%>:<%#EmailGUID%>:<%#WorkflowName%>}{WorkflowRes
ourceActivityMail#3.5}{WorkflowFollowUp:<%#SkeltaFollowUpValue%>}{WorkflowRemi
nderTime:<%#SkeltaReminderTimeValue%>}</td>
```

5. Replace only the following place holders in the copied html template as per the table below. This is because the place holders which is in ' <%# %>' format will not support in customized html format.

Note: Make sure that only the following specified place holders are replaced. Do not replace any other place holders.

Keyword	**Replace with**
<%#EmailURL%>	[%#EmailURL%]
<%#locSalutation%>	[%#locSalutation%]
<%#ActorName%>	[%#ActorName%]
<%#locMailDescription%>	[%#locMailDescription%]
<%#Notes%>	[%#Notes%]
<%#loctaskInfo%>	[%#loctaskInfo%]
<%#locName%>	[%#locName%]
<%#TaskName%>	[%#TaskName%]
<%#locDescription%>	[%#locDescription%]
<%#Description%>	[%#Description%]
<%#locSubmittedBy%>	[%#locSubmittedBy%]
<%#Submitedby%>	[%#Submitedby%]
<%#locSubmittedOn%>	[%#locSubmittedOn%]
<%#SubmittedDate%>	[%#SubmittedDate%]
<%#SubmittedTime%>	[%#SubmittedTime%]
<%#locRepondInTwoWays%>	[%#locRepondInTwoWays%]
<%#locRespondbyMail%>	[%#locRespondbyMail%]

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Keyword	Replace with
<%#locReplyToMail%>	[%#locReplyToMail%]
<%#locColumnName%>	[%#locColumnName%]
<%#locUserInputs%>	[%#locUserInputs%]
<%#locValidInputs%>	[%#locValidInputs%]
<%#locComments%>	[%#locComments%]
<%#locFreeText%>	[%#locFreeText%]
<%#TaskStatus%>	[%#TaskStatus%]
<%#FreeTextTaskStatus%>	[%#FreeTextTaskStatus%]
<%# locTakeAction %>	[%#locTakeAction%]
<%# locSubmit%>	[%#locSubmit %]
<%# locEnter %>	[%#locEnter %]
<%#locRespondByWeb%>	[%#locRespondByWeb%]
<%#locWorkItemView%>	[%#locWorkItemView%]
<%#workitemView%>	[%#workitemView%]
<%#locWorkItemViewURLText%>	[%#locWorkItemViewURLText%]
<%#docview%>	[%#docview%]
<%#locDocViewURLText%>	[%#locDocViewURLText%]
<%#executionView%>	[%#executionView%]
<%#locExecutionViewURLText%>	[%#locExecutionViewURLText%]
<%#locSignature%>	[%#locSignature%]
<%#locEmailSecurityWarning%>	[%#locEmailSecurityWarning%]
<%#ApplicationName%>	[%#ApplicationName%]
<%#EmailGUID%>	[%#EmailGUID%]
<%#WorkflowName%>	[%#WorkflowName%]
<%#SkeltaFollowUpValue%>	[%#SkeltaFollowUpValue%]
<%#SkeltaReminderTimeValue%>	[%#SkeltaReminderTimeValue%]

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. After replacing place holders, the code is as shown below.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
font-family: Arial, Helvetica, sans-serif;
font-size: 11px;
}
.style4 {
font-family: Arial, Helvetica, sans-serif;
font-size: 15px;
font-weight: Normal;
color: #333333;
}
.Username {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: Bold;
color: #333333;
}
.subject {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: Bold;
color: #333333;
}
.style8 {
color: #FF3300;
font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>
<body>
<form id='Form1' action='[%#EmailURL%]' method='post' >
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><span
class="Username">[%#locSalutation%] <strong>[%#ActorName%]</strong></span>,
<p class="style2">[%#locMailDescription%]</p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2">[%#Notes%] </td>
</tr>
<!-- Appropriate sender details can be passed to the mail by using the sender detail
contents as follows
A few are listed below
[%#Value.SendTo.UserDisplayName%]
[%#Value.SendTo.UserEmail%]
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
[%#Value.SendTo.UserId%]
Contents declared in Start Activity can also be used in a similar way.
XML Variables can also be used.
Variables defined in Start Activity can also be used pass the values to the mail.
Pre-defined variables can also be used.
A few are listed below.
[%#Variable.SubmittedBy.UserId%]
[%#Variable.SubmittedBy.UserName%]
[%#Variable.SubmittedBy.UserDisplayName%]
To make use of localization, Create a Key and its value in
WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
and use the key in this html.
EG: Let EmailUserDefinedName be the Localization key in
WorkflowManagementInstalledFolder/Locales/en-US/EmailTemplate.txt
Use the above key in html as @@EmailUserDefinedName@@ to get its localized value.
TO TRY OUT THE ABOVE POINTS, JUST UNCOMMENT THE NEXT COMMENTED PORTION OF CODE.
-->
<!--<tr>
<td bgcolor="#EBEBEB" class="style4">Resource Details</td>
</tr>
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">@@EmailUserDefinedDepartment@@</td>
<td width="79%" bgcolor="#EFEFEF">[%#Variable.SubmittedBy.UserDepartment%]</td>
</tr>
<tr>
<td width="21%">@@EmailUserDefinedEmail@@</td>
<td width="79%">[%#Variable.SubmittedBy.UserDisplayName%]</td>
</tr>
</table>
</td>
</tr>-->
<tr>
<td bgcolor="#EBEBEB">
<span class="style4">[%#loctaskInfo%]</span>
</td>
</tr>
<tr>
<td valign="top" bgcolor="#FFFFFF">
<table width="100%" border="0" cellspacing="0" cellpadding="1">
<tr>
<td width="50%">
<table width="100%" border="0" cellpadding="1" cellspacing="0" bgcolor="#999999">
<tr>
<td>
<table width="100%" border="1" cellpadding="6" cellspacing="0" bordercolor="#CCCCCC"
bgcolor="#FFFFFF" class="style2">
<tr>
<td width="21%" bgcolor="#EFEFEF">
<b>[%#locName%]</b></td>
<td width="79%" bgcolor="#EFEFEF">
[%#TaskName%]</td>
</tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

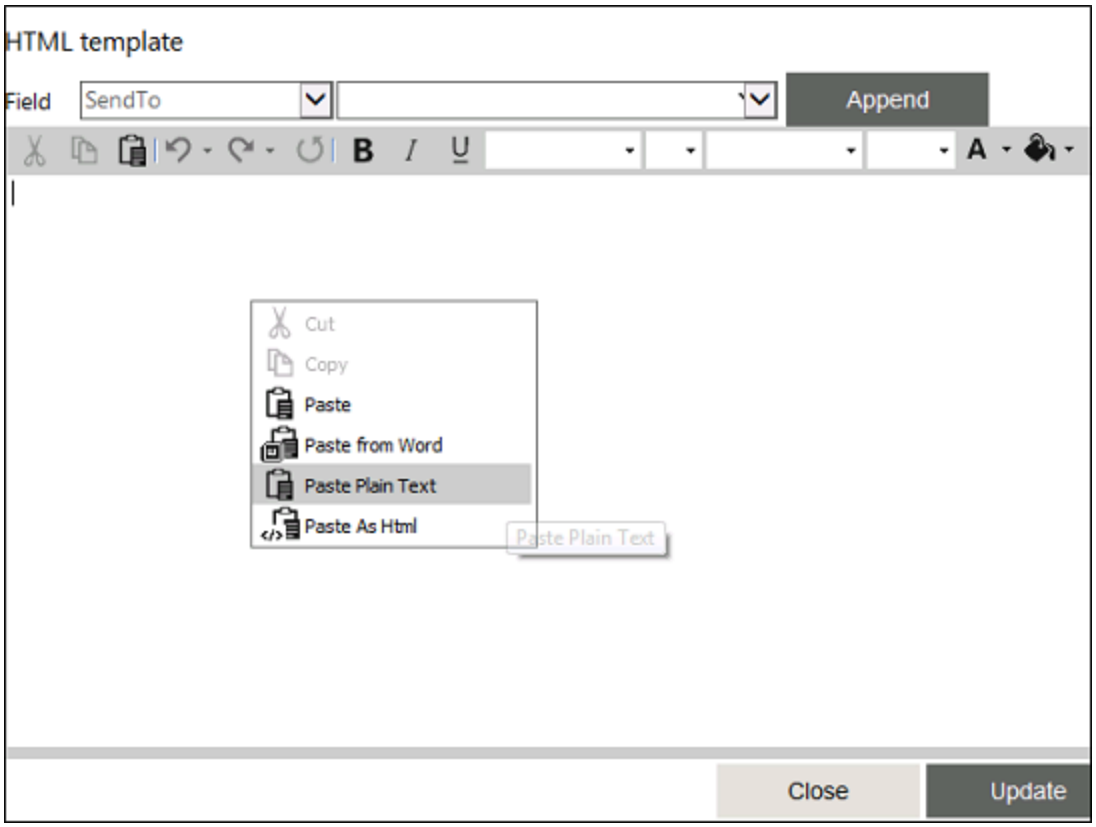
```
<tr>
<td>
<b>[%#locDescription%]</b></td>
<td>
[%#Description%]</td>
</tr>
<tr>
<td bgcolor="#EFEFEF">
<b>[%#locSubmittedBy%]</b></td>
<td bgcolor="#EFEFEF">
[%#Submitedby%]</td>
</tr>
<tr>
<td>
<b>[%#locSubmittedOn%]</b></td>
<td>
[%#SubmittedDate%] [%#SubmittedTime%]</td>
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
<br />
</td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style2">[%#locRepondInTwoWays%]</td>
</tr>
<tr>
<td><table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">
<tr>
<td><strong>[%#locRespondbyMail%]</strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i>[%#locReplyToMail%]</i>
<table width='95%' border='0' align='center' cellpadding='5' cellspacing='1'
bgcolor='#EBEBEB' id="TABLE1" style="width: 100%; height: 25px;">
<tr class="style2" bgcolor='#EBEBEB'>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong>[%#locColumnName%]</strong></td>
<td class="style2" width='30%' bgcolor='#EBEBEB'>
<strong>[%#locUserInputs%]</strong></td>
<td class="style2" bgcolor='#EBEBEB'>
<strong>[%#locValidInputs%]</strong></td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'>[%#locComments%]:</td>
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>[[%#locFreeText%]]</td>
</tr>
<tr valign="top">
<td class="style2" width='30%' bgcolor='#FFFFFF'>
[%#TaskStatus%]:</td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

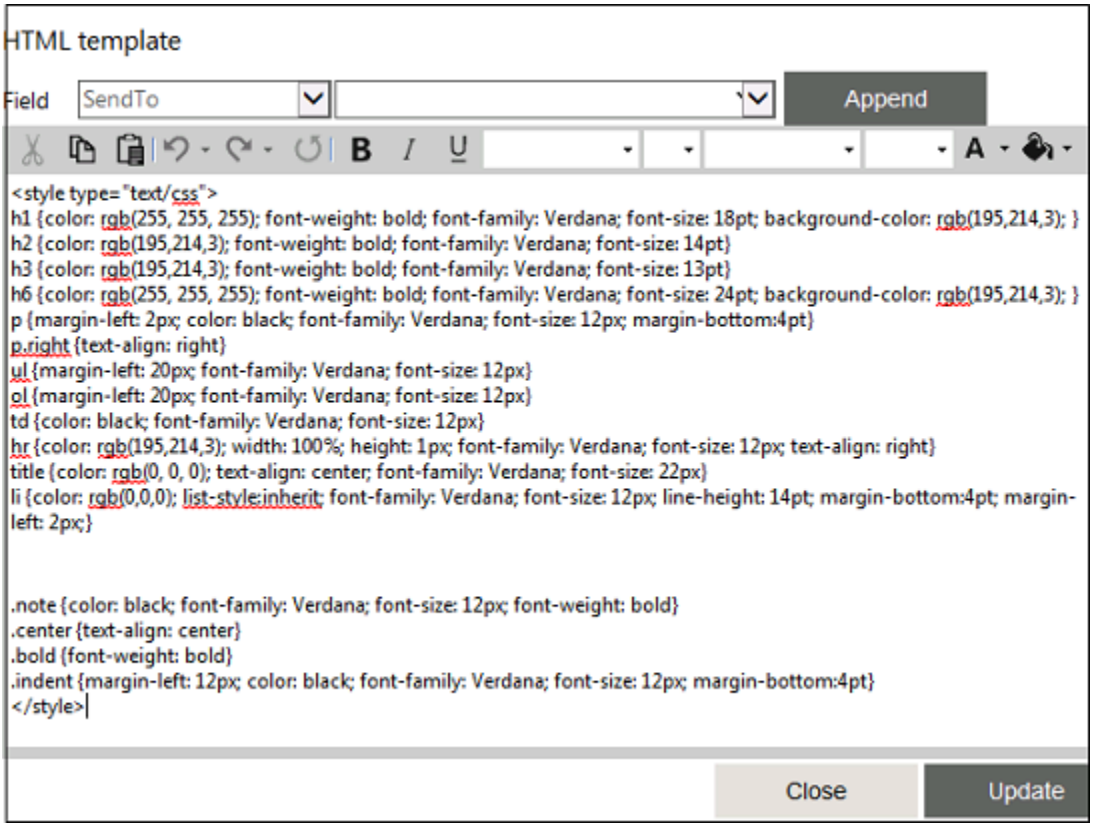
```
<td class="style2" width='30%' bgcolor='#FFFFFF'>[  ]</td>
<td class="style2" bgcolor='#FFFFFF'>
[[%#FreeTextTaskStatus%]]</td>
</tr>
<tr valign="top">
<td class="style2" bgcolor="#ffffff" width="30%" >
[%#locTakeAction%]:</td>
<td class="style2" bgcolor="#ffffff" width="30%" >
[[%#locSubmit%]]</td>
<td class="style2" bgcolor="#ffffff" >
[%#locEnter%]:[[%#locSubmit%]]</td>
</tr>
</table>
</td>
</tr>
<tr>
<td><strong><br />[%#locRespondByWeb%]</strong></td>
</tr>
<tr>
<td style="padding-left:20px"><i>[%#locWorkItemView%]</i>
<br />
<a [%#workitemView%]>[%#locWorkItemViewURLText%]</a>
</td>
</tr>
</table></td>
</tr>
<tr bgcolor="#EBEBEB">
<td align="right" class="style2">
<span><a [%#docview%]>[%#locDocViewURLText%]</a></span>
|
<span><a [%#executionView%]>[%#locExecutionViewURLText%]</a></span>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2">[%#locSignature%]</td>
</tr>
<tr>
<td bgcolor="#FFFFCC" class="style2"><table width="100%" border="0" cellspacing="0"
cellpadding="3">
<tr>
<td class="style2" width="95%"><span
class="style8"><strong>[%#locEmailSecurityWarning%]</strong></span><br />
Workflow:{[%#ApplicationName%]:[%#EmailGUID%]:[%#WorkflowName%]}{WorkflowResourceActi
vityMail#3.5}{WorkflowFollowUp:[%#SkeltaFollowUpValue%]}{WorkflowReminderTime:[%#Skel
taReminderTimeValue%]}</td>
</tr>
</table></td>
</tr>
</table></td>
</tr>
</table>
</form>
</body>
</html>
```

7. Copy the modified code (In this case, the above code) and paste into the HTML template window as Paste Plain Text and click Update.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



8. The following code shows after pasting the code.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Variables, Contents, and XML Variables can also be inserted into the html text wherever necessary by just selecting it from the drop-down provided in the HTML template window and appending it wherever necessary.

Use 'HTML Template' Property of Information Activity

You can use the html email template that is present in the physical folder *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks*\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\Information.html in the 'HTML Template' property of CheckList activity.

Complete the following steps to customize the Task.html.

1. Right-click the **Information** activity and then select **Activity Properties.**
2. Under **Mail Channel** group, set the **Use html template from property to Yes**.
3. Under **Mail Channel** group, click the **HTML template** edit icon.
4. Go to *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks*\WorkflowElements\Default\en-US\Templates\OutlookEmailTemplate\Information.html path and take a copy of the default html template file of Information activity. Make the necessary changes to the copied file.

   Note: While modifying the Information email template the following place holders should not be modified as these are necessary for the engine to process the mails that are received from the actors.

   a. Notes

   ```
   <td bgcolor="#FFFFFF" class="style2"><%#Notes%> </td>
   ```

5. Replace only the following place holders in the copied html template as per the table below. This is because the place holders which is in ' <%# %>' format will not support in customized html format.

   Note: Make sure that only the following specified place holders are replaced. Do not replace any other place holders.

Keyword	Replace with
<%#locSalutation%>	[%#locSalutation%]
<%#ActorName%>	[%#ActorName%]
<%#locMailDescription%>	[%#locMailDescription%]
<%#Notes%>	[%#Notes%]
<%#locResponseOptions%>	[%#locResponseOptions%]
<%#locRespondByWeb%>	[%#locRespondByWeb%]
<%#workitemView%>	[%#workitemView%]
<%#docview%>	[%#docview%]
<%#locDocViewURLText%>	[%#locDocViewURLText%]
<%#executionView%>	[%#executionView%]
<%#locExecutionViewURLText%>	[%#locExecutionViewURLText%]

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Keyword	Replace with
<%#locSignature%>	[%#locSignature%]

6. After replacing place holders, the code is as shown below.

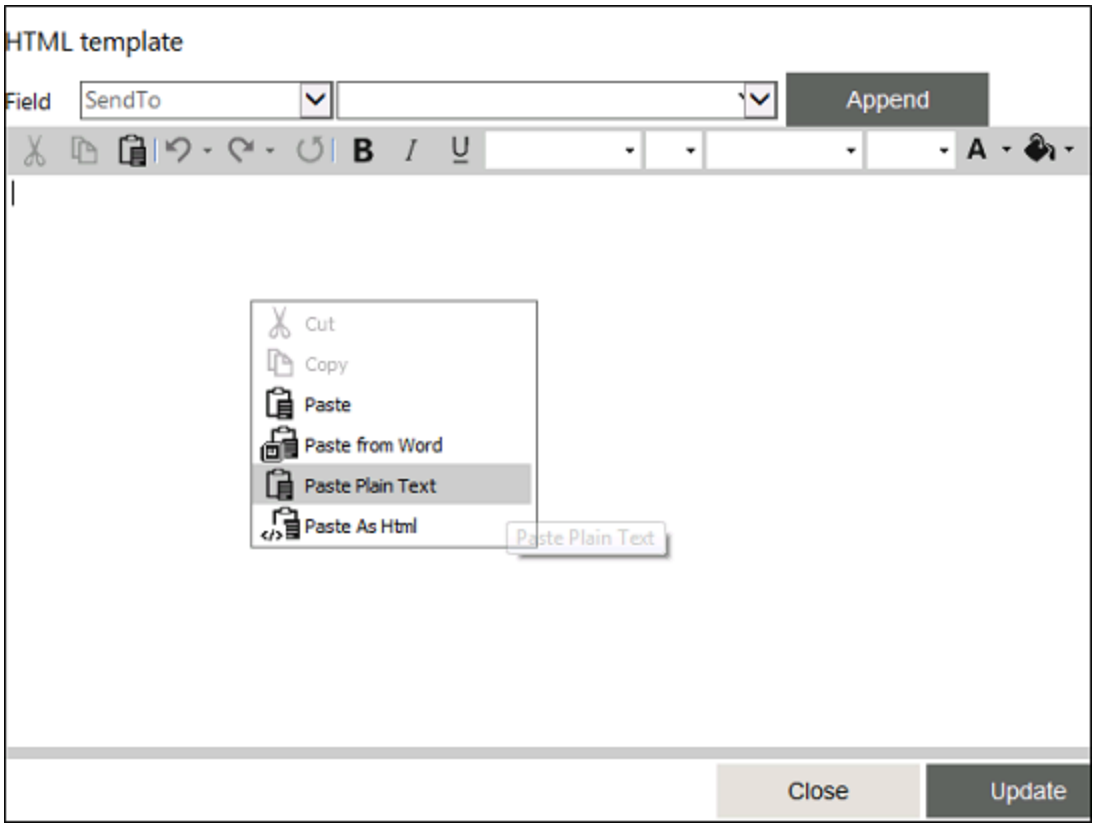
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Mailer</title>
<style type="text/css">
<!--
.style2 {
font-family: Arial, Helvetica, sans-serif;
font-size: 11px;
}
.style4 {
font-family: Arial, Helvetica, sans-serif;
font-size: 15px;
font-weight: Normal;
color: #333333;
}
.Username {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: Bold;
color: #333333;
}
.subject {
font-family: Arial, Helvetica, sans-serif;
font-size: 13px;
font-weight: Bold;
color: #333333;
}
.style8 {
color: #FF3300;
font-weight: bold;
}
.style9 {font-family: Arial, Helvetica, sans-serif; font-size: 11px; font-weight:
bold; }
-->
</style>
</head>
<body>
<table width="700" border="0" align="left" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<td valign="top"><table width="100%" border="0" cellpadding="5" bgcolor="#FFFFFF">
<tr>
<td><span class="Username">[%#locSalutation%] <strong>[%#ActorName%]</strong>
</span>,
<p class="style2">[%#locMailDescription%]</p></td></tr>
<tr>
<td bgcolor="#FFFFFF" class="style2">[%#Notes%] </td>
</tr>
<tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

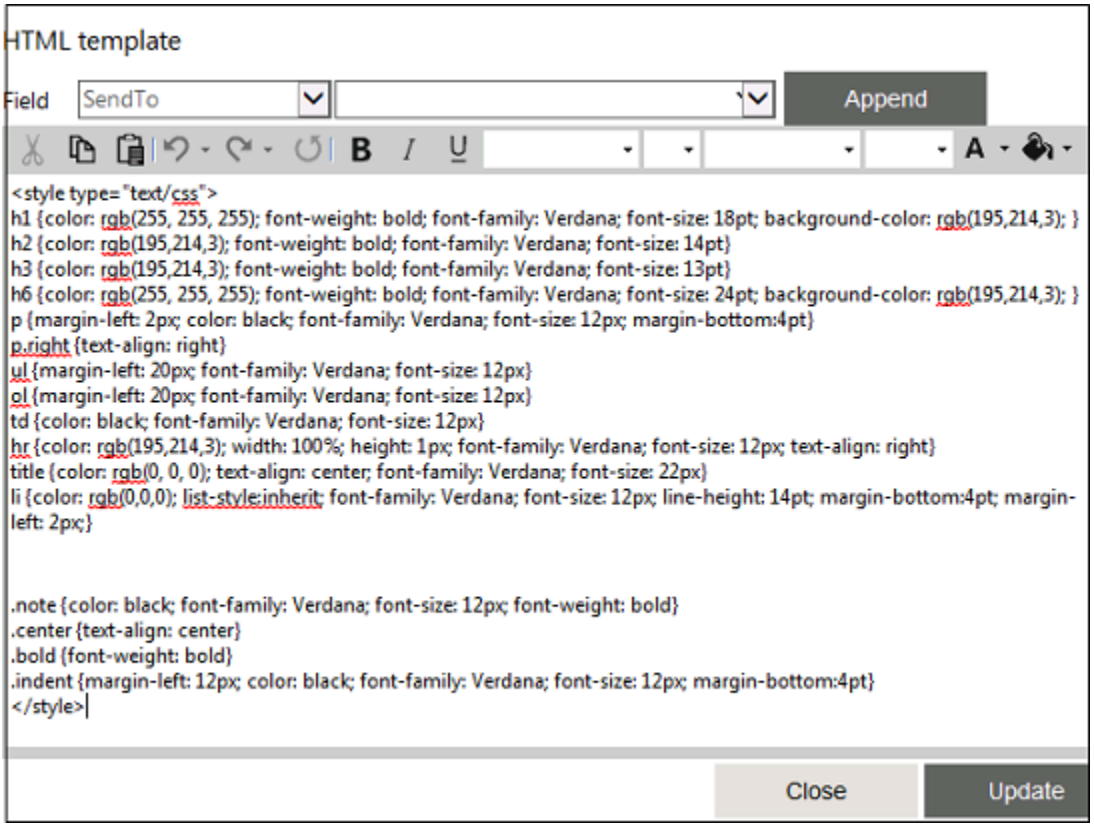
```
<td bgcolor="#EBEBEB" class="style4">[%#locResponseOptions%]</td>
</tr>
<tr>
<td>
<table width="100%" border="0" cellpadding="4" cellspacing="0" class="style2">
<tr>
<td><a [%#workitemView%]>[%#locRespondByWeb%]</a></td>
</tr>
<tr>
<td><a [%#docview%]>[%#locDocViewURLText%]</a></td>
</tr>
<tr>
<td ><a [%#executionView%]>[%#locExecutionViewURLText%]</a></td>
</tr>
</table>
</td>
</tr>
<tr>
<td bgcolor="#EBEBEB" class="style4" style="height:10%"></td>
</tr>
<tr>
<td bgcolor="#FFFFFF" class="style2">[%#locSignature%]</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

7. Copy the modified code (In this case, the above code) and paste into the HTML template window as Paste Plain Text and click Update.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The following code shows after pasting the code.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Add-in for Microsoft Office Outlook

Activities configured to be sent by the mail channel allow users to act on these activities from Microsoft Office Outlook Inbox. This requires that the user machine has the **AVEVA Work Tasks Add-in for Microsoft Office Outlook** installed.

See *WorkTasksAdministrativGuide* for information about installing **AVEVA Work Tasks Add-in for Microsoft Office Outlook**.

Context Menu Options

AVEVA Work Tasks Context Menu displays context-sensitive menu options for:

- Approval Activity
- CheckList Activity
- Choice Activity
- Task Activity

Note:
- The Context-menu is not available in Microsoft Office Outlook 2013.
- The Context menu of the Task Activity does not show the options available.

Approval Activity

1. Select the AVEVA Work Tasks mail which has the Approval activity from the Microsoft Office Outlook Inbox.
2. Right-click the mail to get the context menu.
3. If the mail is not AVEVA Work Tasks mail, then the user will not get any context menu.
4. Approval email options are **Approve** and **Reject**.
5. When a reply is made, the mail icon changes to reply icon to notify the user that the mail is already replied.
6. Once the mail is sent, the workflow engine will generate a confirmation mail.
7. If an approval mail is sent again, a duplicate confirmation mail is received.

   Note:
   - Approval mail actions can be localized.
   - In *[AVEVA Work Tasks Installed Path]*\AVEVA*Work Tasks*\Locales\en-US\EmailTemplate.txt file, change the value for Email_ApproveActivity:Approve and Email_RejectActivity:Reject. This will reflect in the approval mail option1, context menu, and ribbon bar.

Checklist Activity

1. Select the AVEVA Work Tasks mail which has the CheckList Activity from the Microsoft Office Outlook Inbox.
2. Right-click the mail to get the context menu.
3. If the mail is not AVEVA Work Tasks mail then the user will not get any context menu.
4. The option displayed in the context menu for this activity is **View CheckList**.
5. Click **View CheckList**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **CheckList Form** opens.

6. Enter checklist options.
7. Click **Submit**.
8. Once the mail is replied, the mail icon changes to reply icon to notify the user that the mail is already replied.
9. When the mail is sent, the workflow engine will generate a confirmation mail.

Choice Activity

1. Select the AVEVA Work Tasks mail which has the Choice activity from Microsoft Office Outlook Inbox.
2. Right-click the mail to get the context menu.
3. If the mail is not a AVEVA Work Tasks mail then the user will not get any context menu.
4. The Choice activity mail options are **No** and **Yes**.
5. Once the mail is replied, the mail icon changes to reply icon to notify the user that the mail is already replied.
6. When the mail is sent, the workflow engine will generate a confirmation mail,

Task Activity

Functionality of the Workflow.NET.Outlook Client for the Task activity is as follows:

1. Select the AVEVA Work Tasks mail which has the Task activity from the Microsoft Office Outlook inbox.
2. Right-click the mail to get the context menu.
3. If the mail is not AVEVA Work Tasks mail then the user will not get any context menu.
4. In this context menu, the user gets options according to the activity for which the user receives the mail.
5. Once the mail is replied, the mail icon changes to reply icon to notify the user that the mail is already replied.
6. When the mail is sent, the workflow engine will generate a confirmation mail.

Microsoft Office Outlook Follow up

The Microsoft Office Outlook Follow Up functionality helps the user to mark a mail as follow up. It is used to set reminders to perform tasks at the required time.

Details of the Workflow.NET.Outlook Follow Up are as follows:

1. Set this follow up while selecting the channel through the **Channel Selection** page.
2. Specify the time for the reminder.
3. When the user gets any mail marked for follow up, it will be flagged under task list in Microsoft Office Outlook. Until the time set for follow up is met, the mail will be in the due task list itself.
4. There is a limitation to set up reminder time.
5. Reminder should always be given in minutes. Even if the user gives the reminder in seconds, it will be converted to minutes. For Example: If the user sets 0.00:00:20 as reminder time and the mail receiving time is 10.00.00, then the reminder time will be 10.01.00
6. The reminder time specified by the user will be added to the mail receiving time to calculate the actual reminder time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. The reminder time can also be snoozed to the user's convenience.

Ribbon Bar Options

The functions of the context menu are also available by using the AVEVA Work Tasks ribbon bar. The ribbon bar has been configured for Approval, CheckList, Choice, and Task mail items.

Approval Activity

The ribbon bar for approval activity has the same options as in the context menu.

View CheckList Activity

The ribbon bar for the View CheckList activity has the same options as in the context menu.

Choice Activity

The ribbon bar for the Choice activity has the same options as in the context menu.

Task Activity

The ribbon bar for Task activity has the same options as in the context menu. Select the task status from the drop down list and submit.

Database

AVEVA Work Tasks uses the following two databases:

- Farm Database in which Farm Configuration settings are stored.
- Repository Database with tables pertaining to each repository.

Note:
The recommended collation for the databases is:
SQL_Latin1_General_CP1_CI_AS

The Farm and Repository databases support case sensitive collation:
SQL_Latin1_General_CP1_CS_AS

Use Database Scripts to create Database Tables

A database script is automatically created for the SQL Server database when AVEVA Work Tasks is installed. This is located in the corresponding folder in *C:\Program files\AVEVA\Work Tasks\DatabaseScripts* folder. You need to run the script corresponding to the database in your installation before you can use AVEVA Work Tasks.

To create the Database Tables

1. Browse to the DatabaseScripts folder corresponding to the SQLServer database. Ensure that the Database script, Workflow.NET.sql is available.
2. In SQL Server, open the Query Analyzer.
3. In the Query Analyzer, open the script and run it.

   This creates the SQL Server database tables required by AVEVA Work Tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Advanced Features

This section describes the advanced features of AVEVA Work Tasks, which include but are not limited to:

- Events Framework
- Importing BPMN Process Models from Microsoft Visio
- Using Message Logs
- Managing and Synchronizing Threads in Engine
- Entities

Events Framework

Why Event Framework?

Automate different business scenarios which have events incorporated. Some of these scenarios are:

- Waiting for a document to get updated
- On creation of a document
- On receiving an email
- On receiving a SMS

Event Framework allows users to create their own event provider which optionally can be hosted inside the Engine. It has different interfaces which will help to handle different event scenarios very easily.

Event Framework Components

The following are the main components of Event Framework.

EventProviders

This is the main component of Event Framework. These are the component which is handling a specific event. Users need to create different event providers to handle their events. All event providers need to implement an interface called IEventClientServiceProvider

There are two types of Event Providers First type is hosted inside the Engine. To create these kind of providers you need to implement an additional interface called IEventServiceProvider. If your event provider handles load balancing you need to implement another interface called IEventSupportsLoadBalancing

EventPorts

EventPorts can be considered as an endpoint for EventProviders. For example, for a wait for email event provider, need to get email from different sources like POP3 or forwarding from an Exchange server etc. Each of these sources can implement as EventPorts. For a file watcher event provider, there will be different EventPorts which are watching different Folders.

The user can use IEventPort interface to implement EventPort.

EventBinding

This client side component is responsible for consuming events and alert workflow. You need to create an

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

EventBinding for each interested events.

EventBindings are stored in SKEventBinding table.

EventParameter

This is the correlation parameter which filters events and its interested EventBindings. Every Event Provider has a set of EventParameters. Using these parameters, it will filter eventbindings interested to the specific provider events.

EventItem

This is an event message container component.

The Event Framework is designed to support any type of events such as, file system event provider, share point event provider, email event provider. The following are the interfaces and class for the implementation of the Framework. These interfaces and classes implemented in file event provider class file.

Interfaces and Classes

The following are the interfaces and classes used in the Event Framework. All the interfaces need to be implemented in the event service class files.

Interfaces:

- IEventPort:- This interface represents a port used by the event service provider

Properties:-

portname

port guid

event name

port xml string

Port generally contains the information about event listener. For e.g. Incase it is FileSystemEventProvider, the port contains name of directories which file system event provider is going to monitor.

- **IEventClientServiceProvider**:- This interface need to be implemented in the event handler class file and returns a new event port and set its values. The read only property Event name is hard coded in each event provider.

Methods	**Functionality**
GetNewEventPort()	Returns new empty port object
LoadEventPort()	Creates new port and set its values.
GetAllEventParameters()	Returns all event parameters which would be available whenever an event is raised by the event service provider. This is useful to give event notification to appropriate consumer.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Methods	Functionality
IsEventValidForTheTarget()	Changes the event message before it reaches the work flow engine. This is like additional preprocessing on the event message before it delivered.

- **IEventServiceProvider**:- This interface will start the events and need to be implemented in the event handler class file.

Methods	Functionality
StartListeningPort()	Starts listening to a specified port
StopListeningPort()	Stops listening to a specified port
RefreshListeningPort()	Refreshes the port to which it is listening.

Classes:

- **EventBinding**:- This class file is used to set the event parameters such as event name, Portguid, Application name etc. and saving to the SKEventbinding table. This file is inherited from Skelta.Core.DS.SingleTableObject class file. This base class has the method save for saving the data to the event binding table. This converts the parameters as in XML format and save the data to the event binding table. After raising the events, the specified record is deleted from the table by calling the delete method.
- **EventBindingCollection**:- This will retrieve the matching event item record from the eventbinding table. This generates the select query and set all sql parameters and retrieve the corresponding event record.
- **EventItem**:- This class file is used to create a new event object. Here the parameters are event provider name, event port, event parameters and event message.
- **EventParameter**:- This class file holds parameters used for correlation between an event and an eventbinding.
- **EventServiceManager**:- Methods of this class will be called from the engine to start the events, stop the events etc. This file called the StartListeningPort method which is implemented in the event handler class file. This contains other important methods such as GetAllEventProviders and GetAllEventPorts. First will retrieve the event providers from the table and second will retrieve the event ports from the table.
- **EventServicePort**:- This class file will set event service port details and save to the SKEventServicePorts table. This class file inherited from the Skelta.Core.DS.SingleTableObject file. This base class facilitated the saving option and represents a particular port record.
- **EventServicePortCollection**:- This will retrieve the event port details for a specific event type. This retrieves collection of ports for the event from the table SKEventServiceports table.

Event Framework Components

The following table describes the main components of AVEVA Work Tasks Event Framework

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Event Framework Components	Description
EventProviders	This is the main component of Event Framework. This component is used to handling specific events. You need to create different event providers to handle your events. All event providers need to implement an interface called IEventClientServiceProvider. There are two types of event providers. First type is hosted inside the Engine. You need to implement additional interfaces as follows: • IEventServiceProvider for creating event providers. • IEventSupportsLoadBalancing, if your event provider handles load balancing.
EventPorts	EventPorts can be considered as an endpoint for EventProviders. For example, for a wait for email event provider email need to be received from different sources such as POP3, or forwarding from Exchange Server. In each of this source EventPorts can be implemented. For a file watcher event provider, there will be different EventPorts which are watching different Folders. User can use IEventPort interface to implement EventPort.
EventBinding	This client side component is responsible for consuming events and alert workflow. You need to create an EventBinding for each interested events. EventBindings are stored in SKEventBinding table.
EventParameter	This is the correlation parameter which filters events and its interested EventBindings. Every event provider has a set of EventParameters. Using these parameters, it will filter eventbindings interested to the specific provider events.
EventItem	This is an event message container component. The Event Framework is designed to support any type of events such as file system event provider, share point event provider, email event provider, and so on.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Events Framework Interfaces and Classes

The following table 1 and table 2 describe the interfaces and classes used in the event Framework. All the interfaces need to be implemented in the event service class files.

Interfaces

Table 1 - Interfaces

Interface	Description
IEventPort	This interface represents a port used by the event service provider and contains portname, port guid, event name and port XML string. Port generally contains the information about event listener. For example, in case it is FileSystemEventProvider, the port contains name of directories which file system event provider is going to monitor.
IEventClientServiceProvider	This interface needs to be implemented in the event handler class file to return a new event port and set its values. It has the methods such as GetNewEventPort,LoadEventPort, GetAllEventParameters, and so on. The read only property event name is hard coded in each event provider. GetNewEventPort method is used to return new empty port object. LoadEventPort method creates new port and set its values. GetAllEventParameters method returns all event parameters which will be available whenever an event is raised by the event service provider. This is useful to give event notification to appropriate consumer. IsEventValidForTheTarget method is useful to change the event message before it reaches the workflow engine. This is like additional preprocessing on the event message before it delivered.
I**EventServiceProvider**	This interface will start the events and need to be implemented in the event handler class file. This has the methods such as StartListeningPort, StopListeningPort, RefreshListeningPort, and so on.

Classes

Table 2 - Classes

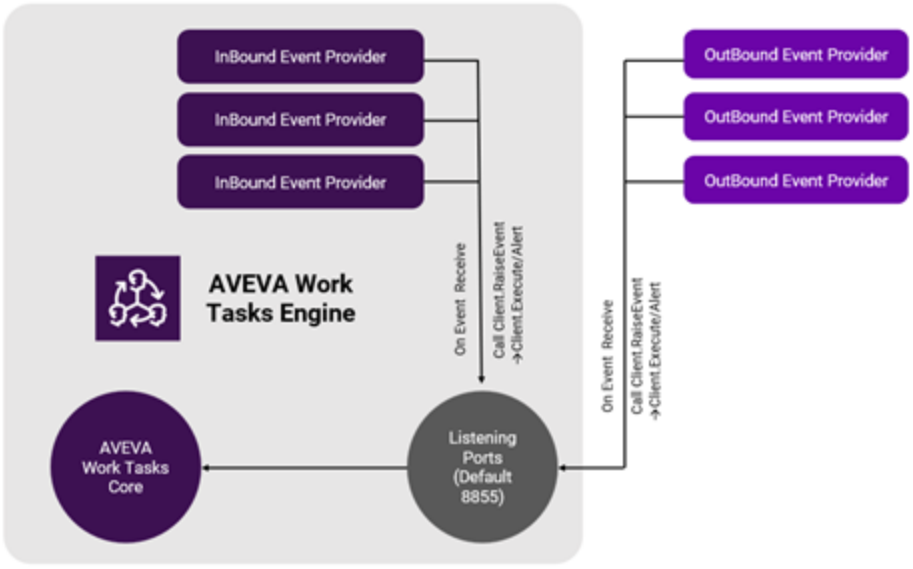
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Class	Description
EventBinding	This class file is used to set the event parameters (event name, Portguid, Application name, and so on) and saving to the SKEventbinding table. This file inherited from Skelta.Core.DS.SingleTableObject class file. This base class has the method save for saving the data to the event binding table. This converts the parameters in XML format and make the saving the data to the event binding table. After raising the events the specified record is deleted from the table by calling the delete method.
EventBindingCollection	This will retrieve the matching event item record from the eventbinding table. This generates the selected query and set all SQL parameters, and retrieves the corresponding event record.
EventItem	This class file is used to create a new event object. Here the parameters are event provider name, event port, event parameters and event message.
EventParameter	This class file holds parameters used for correlation between an event and an eventbinding.
EventServiceManager	This will be called from the Engine to start the events or stop the events. This file called the StartListeningPort method which is implemented in the event handler class file. This contains other important methods such as GetAllEventProviders and GetAllEventPorts. It will retrieve the event providers from the table, and then retrieve the event ports from the table.
EventServicePort	This class file will set event service port details and save to the SKEventServicePorts table. This class file is inherited from the Skelta.Core.DS.SingleTableObject file. This base class facilitates the saving option and represents a particular port record.
EventServicePortCollection	This will retrieve the event port details for a specific event type. This retrieves collection of ports for the event from the table SKEventServiceports table.

Event Provider Architecture

The following figure shows the event provider architecture:

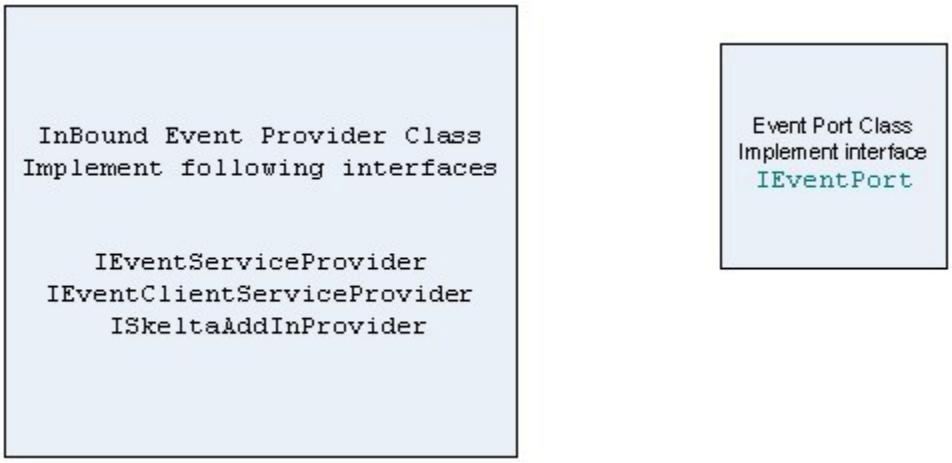
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Inbound Event Provider

Inbound event providers are loaded inside the Engine itself and will wait for events. All event providers configured in SKAddInProviders table with type 'EventHost' are loaded in the Engine and all Event Ports of each provider will be started listening for events. When an event receives, each event provider should call Client.RaiseEvent method.

The following figure shows the Inbound event provider Class and Interfaces:



Outbound Event Provider

Outbound event providers are not loaded inside the Engine. These providers are handling events using their own environment. These kinds of providers are configured in SKAddInProviders table with type 'Event'. Receiving events are responsibility of provider not the Engine. On receiving events, each event provider should call Client.RaiseEvent method.

The following figure shows the Outbound event provider Class and Interfaces:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Link Event Binding and Provider

- Each Provider will have specific set of parameter.
- Each parameter should have a specific mapping field in SKEventBinding table.
- Once an event occurs, the provider parameter values are set by the Provider. This is provider's responsibility.
- These parameters are matched with SKEventBinding collection, matching EventBindings are filtered. This filter can be done inside Client.RaiseEvent method.
- Each matching event binding will process individually.
  - If BindingType property set for workflow, it will execute a new workflow by getting other EventBinding properties.
  - If BindingType property is set for activity, it will alert a waiting activity by getting other properties.

Use Default Event Providers

AVEVA Work Tasks by default provides the following Event Providers:

- Enterprise Console
- BAM
- FileWatcher
- Scheduler

This means that events from the following providers can be used for triggering a Workflow or an Activity by associating these events to the required task. Ensure that FileEventProvider.dll file is available in the bin folder for the following example.

Step 1: Set Folder for File Watcher Event Provider.

To associate to a File Watcher event, first we have to create/set a folder to which file watcher should listen to. For this, go to **Enterprise Console** > **Settings** > **File Watcher**. In the newly opened page, click **New**. The **Add Folder** page appears.

Enter a name for the new File Watcher and specify the physical path of the folder which has to be watched for

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

events. Change the other default settings if necessary and click **Save**. This will successfully create a new File Watcher.

Step 2: Associate workflow with a specific File Watcher Event.

For this example, associate a workflow called **"Test"** to the File Watcher event raised from the specific folder.

Go to **Enterprise Console** > **Workflows**. Right-click the workflow which should be alerted on the File Watcher Event and click **Create Associations**. The **Create Associations** page appears.

Choose one of the file events and the physical folder which has already been configured. Click **Attach Event** which will successfully attach the workflow to the specific event.

A new entry will be made automatically in the **"SKEventBindings"** table in Repository Database which hold the Event Information.

Another entry is made automatically in the **SKEEventAssociations** table which hold information about associated events.

When **"View all Association"** option is clicked, data is retried from the **SKEEventAssociations** table.

The workflow **"Test"** will start execution automatically whenever any new file is created in the specified folder.

Add a Custom Event Provider

AVEVA Work Tasks allows creation of custom event providers using its Event Framework. Using this framework any custom event raised can be used to trigger a workflow or an activity.

Steps to Create Custom Event Provider

AVEVA Work Tasks allows creation of custom event providers using its Event Framework. Using this framework any custom event raised can be used to trigger a workflow or an activity.

Step 1: Creating the Class Library Project

Open a new Class Library type project and name it as you want to name the custom event provider. For this example we will be using **"SkeltaSampleEventProvider"** as the project name, and this should create a namespace of the same name. Please refer to Code Sample to see code implementation.

1. Add reference to Workflow.NET.NET2.dll
2. Add the following to the project:

```
using Skelta.Events.Interfaces;
using Skelta.Core;
using Workflow.NET;
using Skelta.Events;
using Workflow.NET.Engine;
using System.Collections;
```

3. Create a new class for EventPort which implements the interface IEventPort. The IEventPort interface has the following properties.

Properties	Description
string EventName	Get Event Provider name specific to provider.
Guid PortGuid	Id used Internally to identify the port

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Properties	Description
string PortName	Name used to represent the port internally
string PortXmlString	Port level custom logic can be persisted inside this XML string.

4. Create another class in the same file for EventProvider which implements the interfaces
    - **IEventClientServiceProvider**

The **IEventClientServiceProvider** interface has the following properties and methods.

Properties	Description
string EventName	The name of the event can be hard coded in class implementation.
string ListBigIcon	Path of the image file to be displayed in event association page for the custom activity.
string ListSmallIcon	Path of the image file to be displayed in event association page for the custom activity.
Methods	Description
bool EventBindingAdding(EventBinding eventBinding)	This method will get called while adding an eventbinding of the particular provider type. The code to be executed while adding an event binding should be written here.
bool EventBindingDeleting(EventBinding eventBinding)	This method will get called while deleting an eventbinding of this provider type. The code to be executed while deleting an event binding should be written here.
EventBindingUpdating(EventBinding eventBinding)	This method will get called while Updating an eventbinding of this provider type. The code to be executed while Updating an event binding should be written here.
EventParameter[] GetAllEventParameters(IEventPort port)	This method gives all correlation parameters used for filtering event bindings when an event occurs.
string GetEventBindingNameForDisplay(EventBinding eventBinding)	This method will give name for the Event Binding for displaying purpose.
IEventPort GetNewEventPort()	Will return new port specific to the provider.
List<EventProvisionFormDetails> GetProvisionForms(EventProvisionFormType provisionType)	Used to render forms for association. Returns list of EventProvisionFormDetails which can be used for rendering Event association page. (This is the page

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Properties	Description
	which should appear when we are trying to associate a workflow with the Custom Created Event Provider. On the page we should specify the details of the event on which workflow has to be triggered). See the code Samples for more details.
bool IsEventValidForTheTarget(EventItem eventItem, EventBinding eventBinding, VariablesCollection varCollection)	This method will be called from Engine after filtering the eventbindings for specific events. If it returns true then this eventbinding should be considered for executing/alerting workflow. Users business logic can be implemented here. For Activity type eventbindings, variables can be updated here.
IEventPort LoadEventPort(Guid id, string portName, string portXml)	Used internally to get a specific event port.

- **IEventServiceProvider**

Note: This interface should be implemented only if EventProvider is going to be hosted inside the Engine. If the EventProvider is going to be hosted outside, the region "OnEventReceived" in the sample code file should be shifted to the custom implemented Event receiving class.

The **IEventServiceProvider** interface has the following methods:

Methods	Description
Dictionary<string, IEventPort> GetListeningEventPorts()	Returns all the listening Event Ports.
void Initialize()	This method is called to initialize the provider before starting it.
bool IsGlobal()	returns true if the provider is global else returns false. This method is used internally.
object OnClientMessage(string methodName, object message)	-
void OnStop()	-
void RefreshListeningPort(IEventPort eventPort)	This method is from Engine for re-starting an eventport.
void StartListeningPort(IEventPort eventPort)	Method is called for starting an EventPort
void StopListeningPort(IEventPort eventPort)	Method is called for Stopping an EventPort

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **ISkeltaAddInProvider**

Properties	Description
Guid Id	Unique is for the Add in provider
string Settings	Contains the settings for the add in provider
Methods	Description
void InitializeProvider(string settings, Guid id)	Initializes the ad

Note: To implement load Balancing, the interface IEventSupportsLoadBalancing also has to be implemented. Implementing this interface is completely optional and depends on the requirement. The interface has the following definition.

IEventSupportsLoadBalancing	
Methods	**Description**
void OnMachineAdded(string machineName)	Implement this method to specify what has to done when a new machine is added. it takes the new machine name added as its parameter.
void OnMachineDown(string machineName)	Implement this method to specify how load should be distributed when any of the machine is down.

1. Strong name the dll and build the project.
2. Register the dll in the GAC and one copy in the AVEVA Work Tasks bin folder.

Step 2: Add the new Event provider to the Repository

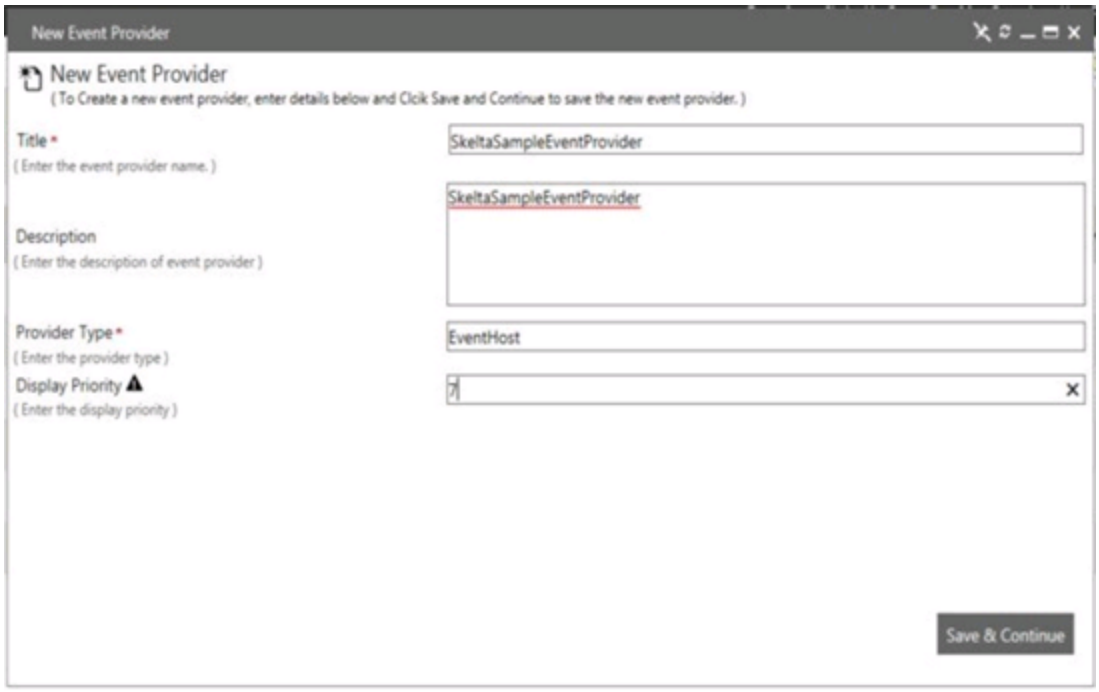
1. Go to Repository Settings from the top right menu bar.
2. Go to the Advanced Settings tab and click Manage Event Providers.
3. Click **New** in the Event Provider page and enter the details for the new Event Provider.

   The **Title** should besSame as that of the Event Provider being created.

   Provider Type should be given as **EventHost** if the event provider is going to hosted inside the Engine, or as Event if the Provider is going to be hosted externally.

   Display priority is the order in which the icon is going to be displayed on the top menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



4. Click **Save & Continue** which will successfully add a new event provider. A new entry will be made in the **SKEEventProviders** table for the Custom event provider.
5. Execute the following Query for the Repository database.

```
INSERT INTO SKAddInProviders (Type,[Name],ClassName,[Assembly],Settings,IsGacAssembly)
VALUES
('EventHost','SkeltaSampleEventProvider','SkeltaSampleEventProvider.EngineHostedProvider
','bin\SkeltaSampleEventProvider.dll','',1)
```

Here,

- **Type**- The field which decides the provider is hosted inside the Engine or externally.
- **Name**- Name of the provider being created
- **ClassName**- Namespace.Classname of the class in the class library file
- **Assembly**- Location of the dll created
- **Settings**- Settings for the provider, if any
- **IsGacAssembly**- Represents whether dll is registered in the GAC or not

A row will be added to the **AddInProviders** table. A new Event Provider has been added in the repository with name SkeltaSampleEventProvider.

Step 3: Verification

We can verify whether it has been correctly added by following the below procedure.

1. Go to the list of workflows.
2. Right-click any workflow.
3. Click **Create Associations** which would lead to a new page from where we can associate to different event providers.
4. With the list of default Event providers, the newly added event provider icon is also displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Code Sample - Class Library for Custom Event Provider

```
using System;
using System.Collections.Generic;
using System.Text;
using Skelta.Events.Interfaces;
using Skelta.Core;
using Workflow.NET;
using Skelta.Events;
using Workflow.NET.Engine;
using System.Collections;
namespace SkeltaSampleEventProvider
{
class SampleEventPort : IEventPort
{
string _EventName;
Guid _PortGuid;
string _PortName;
string _PortXmlString;
public SampleEventPort(string eventName)
{
_EventName = eventName;
 }
public SampleEventPort(string eventName, Guid portGuid, string portName, string portXml)
{
_EventName = eventName;
_PortGuid = portGuid;
_PortName = portName;
_PortXmlString = portXml;
}
#region IEventPort Members
/// <summary>
/// Get Event Provider name specific to provider.
/// </summary>
public string EventName
{
get
{
return _EventName;
}
}
/// <summary>
/// Internally used to identify the port
/// </summary>
public Guid PortGuid
{
 get
{
return _PortGuid;
}
}
/// <summary>
/// Internally used to display port names
 /// </summary>
public string PortName
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
get
{
return _PortName;
}
set
{
_PortName = value;
}
}

/// <summary>
/// Port level custom logic can be persisted inside this xmlstring.
/// </summary>
public string PortXmlString
{
get
{
return _PortXmlString;
}
set
{
_PortXmlString = value;
}
}
#endregion
}
public class EngineHostedProvider : ISkeltaAddInProvider, IEventClientServiceProvider,
IEventServiceProvider
{
#region ISkeltaAddInProvider Members
string _Settings;
Guid _Id;
/// <summary>
/// Return Id
/// </summary>
public Guid Id
{
get
{
return _Id;
}
}
/// <summary>
/// Initialize add in provider properties
/// </summary>
/// <param name="settings"></param>
/// <param name="id"></param>
public void InitializeProvider(string settings, Guid id)
{
_Settings = settings;
_Id = id;
//write code for initializing tasks required for this provider
}
/// <summary>
/// Return addin provider settings
/// </summary>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
public string Settings
{
get
{
return _Settings;
}
}
#endregion
#region IEventClientServiceProvider Members
string _EventName = "SkeltaSampleEventProvider";
/// <summary>
/// This method will get called while adding an eventbinding of this provider type.
/// </summary>
/// <param name="eventBinding"></param>
/// <returns></returns>
public bool EventBindingAdding(EventBinding eventBinding)
{
//write code required while adding an event binding
return true;
}
/// <summary>
/// This method will get called while deleting an eventbinding of this provider type.
/// </summary>
/// <param name="eventBinding"></param>
/// <returns></returns>
public bool EventBindingDeleting(EventBinding eventBinding)
{
//write code required while deleting an event binding
return true;
}
/// <summary>
/// This method will get called while updating an eventbinding of this provider type.
/// </summary>
/// <param name="eventBinding"></param>
/// <returns></returns>
public bool EventBindingUpdating(EventBinding eventBinding)
{
//write code required while updating an event binding
return true;
}
/// <summary>
/// Unique Event Provider Name. This name should be added in SKAddInProviders table.
/// </summary>
public string EventName
{
get
{
return _EventName;
}
}
/// <summary>
/// This method gives all correlation parameters used for filtering event bindings when an
event occurs.
/// </summary>
/// <param name="port"></param>
/// <returns></returns>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
public EventParameter[] GetAllEventParameters(IEventPort port)
{
ArrayList AL = new ArrayList();
EventParameter evntParam = new EventParameter("MandatoryParam1",
EventParameterTypes.StringType, "str1");
evntParam.IsMandatory = true;
AL.Add(evntParam);
evntParam = new EventParameter("OptionalParam2", EventParameterTypes.GUIDType, "GUID1");
AL.Add(evntParam);
evntParam.IsMandatory = false;
return (EventParameter[])AL.ToArray(evntParam.GetType());
}
/// <summary>
/// This method will give name for displaying purpose
/// </summary>
/// <param name="eventBinding"></param>
/// <returns></returns>
public string GetEventBindingNameForDisplay(EventBinding eventBinding)
{
return _EventName + "_" + eventBinding.BindingType.ToString() + "_" + eventBinding.str1;
}
/// <summary>
/// Gives new port specific to the provider.
/// </summary>
/// <returns></returns>
public IEventPort GetNewEventPort()
{
return new SampleEventPort(_EventName);
}
/// <summary>
/// Used to render forms for association.
/// </summary>
/// <param name="provisionType"></param>
/// <returns></returns>
public List<EventProvisionFormDetails> GetProvisionForms(EventProvisionFormType
provisionType)
{
//It should return list of EventProvisionFormDetails which can be used for rendering Event
association page
//This is the page which should appear when we are trying to associate a WF with the Custom
Created Event Provider. On the page we should specify the details of the event on which WF
has to be triggered.
//Before adding, need to set the following EventProvisionFormDetails properties
FormType(Web/Win/Skelta forms engine), FormURI(if web location of aspx file),
List<EventProvisionFormDetails> formDetails = new List<EventProvisionFormDetails>();
EventProvisionFormDetails allEventTypesNewEventForm = new EventProvisionFormDetails();
allEventTypesNewEventForm.FormType = "Web";
//Please click here for details of the aspx page being added
allEventTypesNewEventForm.FormURI = "Plugins/FileWatcher/EventProvider/
FileAssociateEC.aspx";

allEventTypesNewEventForm.Name = "Sample Associations";
allEventTypesNewEventForm.ProvisionType = EventProvisionFormType.NewEvent;
formDetails.Add(allEventTypesNewEventForm);
return formDetails;
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// <summary>
/// This method will be called from Engine after filtering the eventbindings for specific
events.
/// If it returns true then this eventbinding should be considered for executing/alerting
workflow.
/// Users business logic can be done here.
/// For Activity type eventbindings, variables can be updated here.
/// </summary>
/// <param name="eventItem"></param>
/// <param name="eventBinding"></param>
/// <param name="varCollection"></param>
/// <returns></returns>
public bool IsEventValidForTheTarget(EventItem eventItem, EventBinding eventBinding,
VariablesCollection varCollection)
{
varCollection.Add("MyVariable", "string", "Testing Skelta Event Provider");
return true;
}
/// <summary>
/// Returned icon should be shown in the event association page
/// </summary>
public string ListBigIcon
{
get
{
return "Plugins/FileWatcher/Images/icon-sampleprovider_Big.png";
}
}
/// <summary>
/// Returned icon should be shown in the event association page
/// </summary>
public string ListSmallIcon
{
get
{
return "Plugins/FileWatcher/Images/icon-sampleprovider_small.png";
}
}
/// <summary>
/// Internally used to get a specific event port.
/// </summary>
/// <param name="id"></param>
/// <param name="portName"></param>
/// <param name="portXml"></param>
/// <returns></returns>
public IEventPort LoadEventPort(Guid id, string portName, string portXml)
{
return new SampleEventPort(_EventName, id, portName, portXml);
}
#endregion
#region IEventServiceProvider Members
Dictionary<string, IEventPort> dicListeningPorts;
/// <summary>
/// Get all listening event ports
/// </summary>
/// <returns></returns>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
public Dictionary<string, IEventPort> GetListeningEventPorts()
{
return dicListeningPorts;
}
/// <summary>
/// Initialize the provider before starting the provider.
/// </summary>
public void Initialize()
{
dicListeningPorts = new Dictionary<string, IEventPort>();
}
/// <summary>
/// Internally used for identifying global providers
/// </summary>
/// <returns></returns>
public bool IsGlobal()
{
return false;
}
/// <summary>
///
/// </summary>
/// <param name="methodName"></param>
/// <param name="message"></param>
/// <returns></returns>
public object OnClientMessage(string methodName, object message)
{
return null;
//currently not implemented
}
/// <summary>
///
/// </summary>
public void OnStop()
{
//currently not implemented
}
/// <summary>
/// Will be called from Engine for re-starting an eventport
/// </summary>
/// <param name="eventPort"></param>
public void RefreshListeningPort(IEventPort eventPort)
{
//...
}
/// <summary>
/// Will be called from Engine for starting an eventport
/// </summary>
/// <param name="eventPort"></param>
public void StartListeningPort(IEventPort eventPort)
{
//...
dicListeningPorts.Add(eventPort.PortGuid.ToString(), eventPort);
}
/// <summary>
/// Will be called from Engine for stoping an eventport
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// </summary>
/// <param name="eventPort"></param>
public void StopListeningPort(IEventPort eventPort)
{
//...
dicListeningPorts.Remove(eventPort.PortGuid.ToString());
}
#endregion
#region OnEventReceived
/*****************************************************************
 * This region of code is completly sample.
 * This is for demonstrating how to react on events when they are received.
 *
*****************************************************************/
// Current port will be known to the provider when event occurs as this port is listening
for the event
//so initiation as in the next line is not required.
SampleEventPort currentPort = new SampleEventPort("SkeltaSampleEventProvider");
/// <summary>
/// Sample method for raising skelta client on event receive.
/// Normally there will be some objects as part of the event
/// For example
/// Email received, email is the eventMessage
/// File updated - file  is the eventMessage
/// SMS received - smd is the eventMessage
/// </summary>
/// <param name="Message"></param>
public void EngineHostedEventHappend(string eventMessage)
{
EventParameter[] eventParams = GetAllEventParameters(currentPort);
eventParams[0].Value = GetMandatoryParam1(eventMessage);
eventParams[1].Value = GetOptionalParam2(eventMessage);
EventItem evntItm = new EventItem(this, currentPort, eventParams, eventMessage);
VariablesCollection wfVarCol = new VariablesCollection();
wfVarCol.Add("SampleEventVariable", "string", "TestValue");
Client.RaiseEvent(evntItm, "<Test/>", wfVarCol, "",
Workflow.NET.Engine.Interfaces.ActionResult.Sleep);
}
private string GetMandatoryParam1(string EventMessage)
{
//...
return "";
}
private Guid GetOptionalParam2(string EventMessage)
{
//...
return new Guid();
}
#endregion
}

}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Code Sample - Files for File Watcher Event Binding Page

When creating a custom event provider, there should be an interface from which we should be able to associate to the event and set the properties.

For any custom Event Provider we are creating, we should design a similar page which can accept the settings for that provider and should be able to update into the database (SKEventBindings, SKEEventAssociations)

Here, we have given the sample page for creating **FileWatcher event association** page, which can be used as a reference to create a new aspx file with similar functionality.

The newly created code files for filewatcher is added in the folder.

Default\Plugins\FileWatcher\EventProvider

and Images in

Default\Plugins\FileWatcher\Images folder

In the class Library for Custom event provider, under the implementation of **EventClientServiceProvider** in function **GetProvisionForms()** we have to set the details of the page url that we have created above(for event binding) for the **EventProvisionFormDetail** object created.

The following aspx file is the interface from where association can be set for the file watcher event:

```
<%@ Page Language="C#" AutoEventWireup="true"
Inherits="Skelta.Repository.Web.CodeBehind.AssociateFileWatcher" %>
<%@ Register Assembly="RadTabStrip.Net2" Namespace="Telerik.WebControls" TagPrefix="radTS"
%>
<%@ Register TagPrefix="radCln" Namespace="Telerik.WebControls" Assembly="RadCalendar.Net2"
%>
<%@ Register Assembly="Skelta.Repository.Web" Namespace="Skelta.Repository.Web.Lookup"
TagPrefix="cc1" %>
<%@ Register Assembly="RadWindow.Net2" Namespace="Telerik.WebControls" TagPrefix="radW" %>
<%@ Register Assembly="RadComboBox.Net2" Namespace="Telerik.WebControls" TagPrefix="radC"
%>
<link href="<%=cssPath%>Common/StyleSheet/Global.css" rel="stylesheet" type="text/css" />
<html >
<head runat="server" id="Head1">
<title>Associate to File Watcher</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<table cellpadding="0" cellspacing="1" border="0" width="98%" align="center">
<tr>
<td class="pagetitle ">
<img src="../Images/filewatcher-icon.png" hspace="1" align="absmiddle"
/><%=GetLangSpecText("ec_filewatcher_titlename")%>
</td>
</tr>
<tr>
<br />
<td width="100%"  colspan="2">
<span class="description">
<%=GetLangSpecText("ec_filewatcher_helptext")%>
</span>
</td>
</tr>
</table>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<br />
<table cellpadding="7" cellspacing="1" border="0" class="tablebg" width="98%"
align="center">
<tr>
<td class="lefttdbg" style="width: 50%;">
<span class="subtitle">
<%=GetLangSpecText("ec_filewatcher_eventtype")%>
</span>
</td>
<td class="righttdbg" style="width: 50%;">
<asp:DropDownList ID="ListTypes" runat="server" AutoPostBack="false"  CssClass
="inputselect">
</asp:DropDownList>
</td>
</tr>
<tr>
<td class="lefttdbg" style="width: 50%;">
<span class="subtitle">
<%=GetLangSpecText("ec_filewatcher_eventport")%>
<img runat="server" id="imgMandatory" src="<%$ReplaceTemplateExpn:Skeltaforms/Images/
mandatoryicon.PNG%>"
visible="true" hspace="2" align="top" />
<img id="imgsiteUrlerror" alt="Select Folder" runat="server" src="<%$
ReplaceTemplateExpn:Skeltaforms/Images/erroricon.png%>"
hspace="2" align="top" visible="false" />
</span>
</td>
<td class="righttdbg" style="width: 50%;">
<radC:RadComboBox ID="EventsPort" runat="server" Skin="SkeltaCombobox"
RadControlsDir="<%$ReplaceTemplateExpn:telerik/radcontrols/%>"
MaxLength="100" DropDownWidth="303px" Height="150px" NoWrap="True" AutoPostBack="false">
</radC:RadComboBox>
</td>
</tr>
<tr>
<td class="lefttdbg" style="width: 50%;">
<span class="subtitle">
<%=GetLangSpecText("ec_filewatcher_filefullpath")%>
</span>
<br />
<span class="description">
<%=GetLangSpecText("ec_filewatcher_Pattern")%>
</span>
</td>
<td class="righttdbg" style="width: 50%;">
<asp:TextBox ID="txtFilePattern" runat="server" CssClass="inputtext"
Width="150px"></asp:TextBox>
</td>
</tr>
<tr>
<td class="lefttdbg" style="width: 50%;">
<span class="subtitle">
<%=GetLangSpecText("ecm_view_assn_alertver")%>
</span>
</td>
<td class="righttdbg" style="width: 50%;">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<asp:Panel ID="VersionPanel" runat="server">
</asp:Panel>
</td>
</tr>
</table>
<div id="footerdiv">
<table width="100%" align="center" border="0" height="100px">
<tr align="right">
<td width="50%">
</td>
<td width="50%">
<asp:Button ID="AttachEvent" runat="server" CssClass="inputbutton" CausesValidation="False"
/>
<asp:Button ID="CancelEvent" runat="server" CssClass="inputbutton" CausesValidation="False"
/>
</td>
</tr>
</table>
</div>
</div>
</form>
</body>
</html>
```

The below code behind file is responsible for associating a workflow with a File Watcher event.

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Xml;
using System.Xml.Schema;
using System.Text;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Collections.Generic;
using Workflow.NET.PropertyTypes;
using Skelta.Events;
using Workflow.NET;
using Skelta.Forms.Core.Controls;
using Skelta.Repository.Web.List;
using Skelta.Repository.Web;
using Skelta.Repository.List;
using Skelta.Repository.List.Action;
using Skelta.Entity;
using Skelta.Core;
using Workflow.NET.Web.Designer;
using System.IO;
using Skelta.Events.Interfaces;
using Workflow.NET.Interfaces;
using Workflow.NET.Storage;
using System.Xml.Serialization;
namespace Skelta.Repository.Web.CodeBehind
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
public partial class AssociateFileWatcher : System.Web.UI.Page
{
protected string applicationName;
protected ProcessDesigner _designerControl;
protected ProcessDesignerAdapter designerAdapter;
public string error = string.Empty;
protected Guid skListId;
protected Guid workflowId;
protected Guid skListItemId;
protected Guid stPort;
protected string workflowName;
protected string spScripts;
protected Skelta.Repository.List.ListItem item;
protected Skelta.Repository.Web.Lookup.VersionLookup WorkflowVersionLookUp;
protected string mode;
protected Guid eventBindId;
protected string eventListType;
protected string workflowVersion;
protected string effectiveFromDate = "";
protected string effectiveToDate = "";
protected string listName = "Workflow";
protected string eventType;
protected string FileName;
protected string FullPath;
protected string alertMessage;
protected Skelta.Repository.List.ListDefinition associationList;
protected Skelta.Repository.List.ListItem associationItem;
protected string Loggeduser;
protected System.Web.UI.WebControls.TextBox txtFilePattern;
protected System.Web.UI.HtmlControls.HtmlImage imgsiteUrlerror;
protected System.Web.UI.WebControls.DropDownList ListTypes;
protected Telerik.WebControls.RadComboBox EventsPort;
protected System.Web.UI.WebControls.Panel VersionPanel;
protected Button AttachEvent;
protected Button CancelEvent;
ListDefinition _List;
//Themes Implementation changes
protected string cssPath = "";
protected string imgPath1 = "";
protected string imgPath2 = "";
protected object styleUrl = Workflow.NET.TemplateExpressionBuilder.GetUrl("");
protected object imgUrl1 = Workflow.NET.TemplateExpressionBuilder.GetUrl("Plugins/
FileWatcher/Images");
protected object imgUrl2 = Workflow.NET.TemplateExpressionBuilder.GetUrl("Images");
#region "Event enums"
public enum FileEventType
{
Created = 0,
Changed = 1,
Renamed = 2,
Deleted = 3
}
#endregion
protected void Page_Load(object sender, EventArgs e)
{
cssPath = styleUrl.ToString();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
imgPath1 = imgUrl1.ToString();
imgPath2 = imgUrl2.ToString();
ListPageParameters listParams = new ListPageParameters();
listParams.ResolveCurrentUri();
applicationName = listParams.ApplicationName;
if (listParams.ListName.ToLower() == "workflow")
workflowId = listParams.ListItemId;
else
skListItemId = listParams.ListItemId;
ViewState["workflowId"] = workflowId;
ViewState["skListItemId"] = skListItemId;
ViewState["skListId"] = skListId;
ViewState["applicationName"] = applicationName;
ViewState["PageMode"] = mode;
ViewState["UserId"] = listParams.LoggedInUserId;

Loggeduser = listParams.LoggedInUserId;


EventsPort.Width = 320;
{
if (ViewState["workflowId"] != null)
workflowId = (Guid)ViewState["workflowId"];
if (ViewState["skListItemId"] != null)
skListItemId = (Guid)ViewState["skListItemId"];
if (ViewState["skListId"] != null)
skListId = (Guid)ViewState["skListId"];
if (ViewState["applicationName"] != null)
applicationName = ViewState["applicationName"].ToString();
if (ViewState["PageMode"] != null)
mode = ViewState["PageMode"].ToString();
AttachEvent.Click += new EventHandler(AttachEvent_Click);
CancelEvent.Click += new EventHandler(CancelEvent_Click);
AttachEvent.Text = GetLangSpecText("ecm_filewatcher_save");
CancelEvent.Text = GetLangSpecText("ecm_filewatcher_cancel");

}
if (!string.IsNullOrEmpty(applicationName))
{
this.LoadEvents(ListTypes.Text);
}
PopulateWorkflowVersion();
if (!IsPostBack)
{
this.LoadListTypes();
if (!string.IsNullOrEmpty(applicationName))
{
_List = ListDefinition.GetList(new Skelta.Core.ApplicationObject(applicationName),
"FileWatcher List");
LoadSecurityEnblePorts();
}
}
}
protected void AttachEvent_Click(object sender, EventArgs e)
{
CreateXMLVaribale();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
SaveEventBinding();
}
private void SaveEventBinding()
{
string filename;
string filepath;
String eventtype;
string EventType;
FullPath = txtFilePattern.Text;
EventType = ListTypes.SelectedItem.Text;
ListDataHandler wfhandler = new ListDataHandler(applicationName, "Workflow");
Skelta.Repository.List.ListItem wfitem;
wfitem = wfhandler.GetListItem(workflowId);
workflowName = wfitem.Title;
EventBinding eventBind;
eventBind = new EventBinding(new Skelta.Core.ApplicationObject(applicationName));
eventBind.BindingType = EventBindingType.Workflow;
eventBind.Disabled = false;
eventBind.Disable_After_Expiry = false;
eventBind.NoofEvents = 0;
eventBind.EventName = "FileWatcher";
eventBind.Alert_ApplicationName = applicationName;
eventBind.Alert_Workflow = workflowName;//the current list item
eventBind.Alert_Version = wfitem.CurrentVersion.VersionStamp.ToString();
eventBind.Alert_UserId = ViewState["UserId"];
int flag = 0;
if (EventsPort.Items.Count != 0)
{
//
bool chkFolder = true;
string strPort = EventsPort.SelectedItem.Text;
 if (chkFolder == true)
{

stPort = new Guid(EventsPort.SelectedValue);
eventBind.PortGuid = stPort;
flag = 1;
}
else
{
error = GetLangSpecText("ec_filewatcher_portRemoved");
Page.ClientScript.RegisterClientScriptBlock(typeof(Page), "regJs", "<script>alert(\"" +
error + "\");</script>");
}
}
else
{
imgsiteUrlerror.Visible = true;
}
if (flag == 1)
{
string[] strFileType = txtFilePattern.Text.Split('.');
if (strFileType.Length > 1)
{
if (strFileType[0] == string.Empty && strFileType[1] != string.Empty)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
eventBind.str2 = strFileType[1].ToString();
}
if (strFileType[0] != string.Empty && strFileType[1] != string.Empty)
{
if (strFileType[0] == "*" && strFileType[1] == "*")
{
}
else if (strFileType[0] == "*" && strFileType[1] != string.Empty)
{
eventBind.str2 = strFileType[1];
}
else
{
eventBind.str1 = txtFilePattern.Text;
}
}
}
else if (strFileType[0] != string.Empty)
{
eventBind.str1 = strFileType[0];
}
eventBind.str3 = EventType;
//New code
string SqlQuery = "";
//stPort
if (eventBind.str1 == "" && eventBind.str2 == "")
{
SqlQuery = "select * from SKEventBindings ";
SqlQuery += "where str1 is null ";
SqlQuery += "and str2 is null ";
SqlQuery += "and str3='" + EventType + "'";
SqlQuery += "and PortGuid=@PortGuid ";
SqlQuery += "and Alert_Workflow='" + workflowName + "' ";
}
else if (eventBind.str1 != "" && eventBind.str2 == "")
{
SqlQuery = "select * from SKEventBindings ";
SqlQuery += "where str1='" + eventBind.str1 + "' ";
SqlQuery += "and str2 is null ";
SqlQuery += "and str3='" + EventType + "' ";
SqlQuery += "and PortGuid=@PortGuid ";
SqlQuery += "and Alert_Workflow='" + workflowName + "' ";
}
else if (eventBind.str1 == "" && eventBind.str2 != "")
{
SqlQuery = "select * from SKEventBindings ";
SqlQuery += "where str1 is null ";
SqlQuery += "and str2='" + eventBind.str2 + "'";
SqlQuery += "and str3='" + EventType + "' ";
SqlQuery += "and PortGuid=@PortGuid ";
SqlQuery += "and Alert_Workflow='" + workflowName + "' ";
}
else
{
SqlQuery = "select * from SKEventBindings ";
SqlQuery += "where str1='" + eventBind.str1 + "' ";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
SqlQuery += "and str2='" + eventBind.str2 + "'";
SqlQuery += "and str3='" + EventType + "' ";
SqlQuery += "and PortGuid=@PortGuid ";
SqlQuery += "and Alert_Workflow='" + workflowName + "' ";
 }
Guid portGuid = Guid.Empty;
portGuid = stPort;
DataTable temp = null;
IDataHandler dbhandler = DataHandlerFactory.GetDataHandler(wfitem.List.Configuration);
IDataParameter Param = dbhandler.GetParameter("@PortGuid", portGuid);
try
{
using (dbhandler)
{
temp = dbhandler.ReadData(SqlQuery, Param);
}
}
catch { }
if (temp == null)
return;
if (temp.Rows.Count != 0)
{
error = GetLangSpecText("ecm_assn_multiplebindings");
Page.ClientScript.RegisterClientScriptBlock(typeof(Page), "regJs", "<script>alert(\"" +
error + "\");</script>");
return;
}
eventBind.Save();
eventBindId = eventBind.Id;
filename = txtFilePattern.Text;
filepath = EventsPort.SelectedValue;
eventtype = EventType;
alertMessage = EventsPort.SelectedItem.Text  + "  " + filename + "  " + eventtype;
StringBuilder eventXml = new StringBuilder();
eventXml.Append("<EventBind id=\"" + eventBindId + "\">");
eventXml.Append("<ApplicationName>" + applicationName + "</ApplicationName>");
eventXml.Append("<Workflow>");
eventXml.Append("<WorkflowName>" + workflowName + "</WorkflowName>");
eventXml.Append("<WorkflowAlert>" + alertMessage + "</WorkflowAlert>");
eventXml.Append("<WorkflowEffectiveFrom>" + effectiveFromDate +
"</WorkflowEffectiveFrom>");
eventXml.Append("<WorkflowEffectiveTo>" + effectiveToDate + "</WorkflowEffectiveTo>");
eventXml.Append("</Workflow>");
eventXml.Append("<Parameters>");
eventXml.Append("<File FileName=\"" + filename + "\" name=\"" + listName + "\"></File>");
eventXml.Append("<File FilePath=\"" + filepath + "\" name=\"" + listName + "\"></File>");
eventXml.Append("<File EventType=\"" + eventtype + "\" name=\"" + listName + "\"></File>");
eventXml.Append("</Parameters>");
eventXml.Append("</EventBind>");
SaveAssociationList(eventXml.ToString());
}
}
private void SaveAssociationList(string eventXml)
{
if (associationList == null)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
associationList = ListDefinition.GetList(new
Skelta.Core.ApplicationObject(applicationName), "Event Associations");
}
if (associationList != null)
{
Guid webApplicationItemReferenceId = Guid.Empty;
ListDefinition webApplicationList = ListDefinition.GetList(new
Skelta.Core.ApplicationObject(applicationName), "Event Associations");
Skelta.Repository.List.ListItemCollection webApplicationListItems = new
Skelta.Repository.List.ListItemCollection(webApplicationList);
Hashtable webApplicationTableColumnNames = new Hashtable();
SortedDictionary<string, Skelta.Forms.Core.Controls.BaseDataControl> fields =
webApplicationListItems.GetRecordsFieldMapping();
foreach (KeyValuePair<string, Skelta.Forms.Core.Controls.BaseDataControl> info in fields)
{
webApplicationTableColumnNames.Add(info.Value.Name, info.Key);
}
Skelta.Repository.List.ListItem webApplicationListItem = null;
if (webApplicationItemReferenceId == Guid.Empty)
webApplicationListItem = new Skelta.Repository.List.ListItem(webApplicationList);
else
webApplicationListItem = new Skelta.Repository.List.ListItem(webApplicationList,
webApplicationItemReferenceId);
if (webApplicationListItem != null)
{
if (associationItem == null)
{
associationItem = new Skelta.Repository.List.ListItem(associationList);
}
ListMainForm associationMainForm = (ListMainForm)associationItem.ListForm.Records[0];
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_evntbndxml")).Value
= eventXml;
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_wrkflwname")).Value
= workflowName;
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_evntprvdrname")).Va
lue = "FileWatcher";
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_evnttype")).Value =
"FileWatcher";
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_evntstatus")).Value
= "New";
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_desc")).Value =
alertMessage;
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_title")).Value =
ListTypes.Text;
((ListGuidDataItem)associationMainForm.FindControlByID("_sys_sk_evntassbnd_id")).Value =
eventBindId;
associationItem.LoggedInUserId = Loggeduser;
associationItem.Save(ListItemVersionStatus.Published);
error = string.Empty;
string strSuccessText = GetLangSpecText("ecm_filewatcher_savesuccess");
Page.ClientScript.RegisterClientScriptBlock(typeof(Page), "regJs",
"<script>alert(\""+strSuccessText+"\");</script>");
}
else
{
throw new Exception("Not able to fetch main Reference Id");
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
}
}
private void LoadListTypes()
{
string listTypesSelectedValue = ListTypes.SelectedValue;
this.ListTypes.DataSource = Enum.GetNames(typeof(FileEventType));
this.ListTypes.DataBind();

if (listTypesSelectedValue != null && listTypesSelectedValue != "")
this.ListTypes.SelectedValue = listTypesSelectedValue;
}
private void LoadEvents(string listType)
{
}
private void PopulateWorkflowVersion()
{
if (!string.IsNullOrEmpty(applicationName) && workflowId != Guid.Empty)
{
if (WorkflowVersionLookUp == null)
{
WorkflowVersionLookUp = new Skelta.Repository.Web.Lookup.VersionLookup();
WorkflowVersionLookUp.ApplicationName = applicationName;
WorkflowVersionLookUp.ID = "WorkflowVersionLookUp";
WorkflowVersionLookUp.ListName = "Workflow";
WorkflowVersionLookUp.ListItemId = workflowId;// set the listitem id of the workflow..
WorkflowVersionLookUp.IsLatestVisible = true;
VersionPanel.Controls.Clear();
VersionPanel.Controls.Add(WorkflowVersionLookUp);
}
}
}
protected string GetLangSpecText(string keyName)
{
SkeltaResourceSetManager resManager = new SkeltaResourceSetManager();
return resManager.GlobalResourceSet.GetString(keyName);
}
protected void CancelEvent_Click(object sender, EventArgs e)
{
txtFilePattern.Text = string.Empty;
}
private void LoadPorts()
{
Dictionary<string, IEventPort> eventPorts = GetAllEventPorts(applicationName,
"FileWatcher");
foreach (string portGuid in eventPorts.Keys)
{
Telerik.WebControls.RadComboBoxItem lst = new Telerik.WebControls.RadComboBoxItem();
lst.Text = eventPorts[portGuid].PortName;
lst.Value = eventPorts[portGuid].PortGuid.ToString();
EventsPort.Items.Add(lst);
}
}
private void LoadSecurityEnblePorts()
{
DataTable dtTable = null;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Skelta.Repository.List.ListItemCollection objSecPorts = new
Skelta.Repository.List.ListItemCollection(_List);
dtTable = objSecPorts.GetRecordsForConsume(Loggeduser, Skelta.Repository.Display.AllItems);
if (dtTable.Rows.Count > 0)
 LoadPorts();
}
private Dictionary<string, IEventPort> GetAllEventPorts(string applicationName, string
eventProviderName)
{
//If it is not given here, for every serviceprovider class, geteventports method should be
implemented.
// it will be duplication of code everywhere.
Dictionary<string, IEventPort> colEventPorts = new Dictionary<string, IEventPort>();

if (applicationName != "")
{
FILEWATCHER myObject;
EventServicePortCollection colEventServicePorts = new EventServicePortCollection(new
ApplicationObject(applicationName), eventProviderName);
foreach (EventServicePort evntPort in colEventServicePorts.Items)
{
myObject = new FILEWATCHER();
System.IO.StringReader read = new StringReader(evntPort.PortXml);
System.Xml.Serialization.XmlSerializer serializer = new
System.Xml.Serialization.XmlSerializer(myObject.GetType());
System.Xml.XmlReader reader = new XmlTextReader(read);
myObject = (FILEWATCHER)serializer.Deserialize(reader);
if (myObject.REPOSITORY.Trim() == applicationName)
colEventPorts.Add(evntPort.Name, evntPort.EventPort);
}
}
return colEventPorts;
}
private void CreateXMLVaribale()
{
try
{
ListDataHandler wfhandler = new ListDataHandler(applicationName, "Workflow");
Skelta.Repository.List.ListItem wfitem;
wfitem = wfhandler.GetListItem(workflowId);
workflowName = wfitem.Title;
Skelta.Core.AddInProvider provider = new Skelta.Core.AddInProvider(new
Skelta.Core.ApplicationObject(applicationName), GetProviderId("XmlDocument",
applicationName));
Workflow.NET.Interfaces.IXmlStorageType storageHandler =
(Workflow.NET.Interfaces.IXmlStorageType)provider.CreateInstance(true);
Workflow.NET.Diagram _diagram =
Workflow.NET.Diagram.GetDiagramCreateIfNotFound(applicationName, workflowName,
wfitem.CurrentVersion.VersionStamp.ToString());
Skelta.Enterprise.CommonFunctions commonFunctions = new
Skelta.Enterprise.CommonFunctions();
_diagram.PreProcessDiagaram();
bool flag = true;
foreach (XmlVariable var in _diagram.XmlVariables.Values)
{
if (var.Name.Equals("FileWatcherXMLData"))
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
flag = false;
}
}
if (flag)
{
Workflow.NET.XmlVariable XVariable = new Workflow.NET.XmlVariable();
XVariable.Name = "FileWatcherXMLData";
XVariable.Description = "File watcher XML Data";
XVariable.StorageType = "XmlDocument";
string eventdataFilePath =
System.Web.HttpContext.Current.Server.MapPath(System.Web.HttpContext.Current.Request.Applic
ationPath + "/" + Workflow.NET.Config.GetTemplateDirectoryFromWebConfig() + "Default/
XMLSchemas/FileWatcherEventData.xml");
XVariable.Schema = ParseXMLForData(eventdataFilePath);
XVariable.StorageHandler = storageHandler;
((Workflow.NET.XmlStorageType.XmlDocument)(XVariable.StorageHandler)).StorageData =
"variables";
_diagram.XmlVariables.Add(XVariable.Name, XVariable);
commonFunctions.WriteDiagram(_diagram, applicationName, workflowName,
wfitem.CurrentVersion.VersionStamp.ToString());
_diagram.Dispose();
}
}
catch (Exception)
{
}
}
private Guid GetProviderId(string storageType, string applicationName)
{
Skelta.Core.AddInProviderCollection _Provider = new Skelta.Core.AddInProviderCollection(new
Skelta.Core.ApplicationObject(applicationName));
string FilterExp = "where Type =@Type and Name=@Name";
Skelta.Core.DS.SkeltaParameter param1 = new Skelta.Core.DS.SkeltaParameter("@Type",
"XmlStorageType");
Skelta.Core.DS.SkeltaParameter param2 = new Skelta.Core.DS.SkeltaParameter("@Name",
storageType);
System.Data.DataSet DsProvider = _Provider.GetMatchingRecords(FilterExp, param1, param2);
string s = "";
if (DsProvider.Tables[0].Rows.Count > 0)
{
s = DsProvider.Tables[0].Rows[0].ItemArray.GetValue(0).ToString();
}
return new Guid(s);
}
private string ParseXMLForData(string file)
{
TextWriter txt = new StringWriter();
StreamReader sReader = new System.IO.StreamReader(file);
System.Xml.XmlTextReader reader = new System.Xml.XmlTextReader(sReader);
XmlDocument XDoc = new XmlDocument();
XDoc.Load(reader);
reader.Close();
sReader.Close();
StringReader stringReader1 = new StringReader(this.AddNameSpacePrefixIntoHolder(XDoc));
XmlTextReader xmlTextReader1 = new XmlTextReader(((TextReader)stringReader1));
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
XmlSchemaSet schemaSet = new XmlSchemaSet();
XmlSchemaInference schema = new XmlSchemaInference();
schemaSet = schema.InferSchema(xmlTextReader1);
foreach (XmlSchema s in schemaSet.Schemas())
{
s.Write(txt);
}
string xsdstring = txt.ToString();
return xsdstring;
}
private string AddNameSpacePrefixIntoHolder(XmlDocument oxmlDoc)
{
foreach (XmlNode xmlNode1 in oxmlDoc.ChildNodes)
{
if (xmlNode1.NamespaceURI.Length > 0)
this.AddNameSpaceHolder(xmlNode1.NamespaceURI, xmlNode1.Prefix);
if (xmlNode1.ChildNodes.Count > 0)
{
string string2 = this.FindNextNode(oxmlDoc, xmlNode1);
}
}
 return oxmlDoc.OuterXml;
}
private string FindNextNode(XmlDocument oxmlDoc, XmlNode oNode)
{
string string1 = null;
foreach (XmlNode xmlNode1 in oNode.ChildNodes)
{
if (xmlNode1.NamespaceURI.Length > 0)
{
this.AddNameSpaceHolder(xmlNode1.NamespaceURI, xmlNode1.Prefix);
if (xmlNode1.ChildNodes.Count > 0)
{
string string2 = this.FindNextNode(oxmlDoc, xmlNode1);
}
}
else
{
if (xmlNode1.ChildNodes.Count > 0)
{
string string3 = this.FindNextNode(oxmlDoc, xmlNode1);
}
continue;
}
}
return string1;
}
public void AddNameSpaceHolder(string NameSpaceURI, string Prefix)
{
Hashtable nameSpaceHolder = new Hashtable();
bool b1 = false;
NameSpaceInfo nameSpacePrefix1 = null;
Exception exception1 = null;
IEnumerator iEnumerator1 = null;
try
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
b1 = false;
if (nameSpaceHolder.Count <= 0)
{
nameSpacePrefix1 = new NameSpaceInfo();
nameSpacePrefix1.NameSpaceURI = NameSpaceURI;
nameSpacePrefix1.NameSpacePrefix = Prefix;
nameSpaceHolder.Add(Prefix, nameSpacePrefix1);
goto L_00DA;
}
try
{
iEnumerator1 = nameSpaceHolder.Values.GetEnumerator();
while (iEnumerator1.MoveNext())
{
nameSpacePrefix1 = ((NameSpaceInfo)iEnumerator1.Current);
}
}
finally
{
if (iEnumerator1 is IDisposable)
{
((IDisposable)iEnumerator1).Dispose();
}
}
L_0093:
{
}
if (b1)
{
return;
}
nameSpacePrefix1 = new NameSpaceInfo();
nameSpacePrefix1.NameSpaceURI = NameSpaceURI;
nameSpacePrefix1.NameSpacePrefix = Prefix;
if (!nameSpaceHolder.Contains(Prefix))
nameSpaceHolder.Add(Prefix, nameSpacePrefix1);
return;
L_00DA:
{
}
}
catch (Exception exception2)
{
exception1 = exception2;
throw new Exception(exception1.Message);
}
}
}
public class FILEWATCHER
{
public string PORTNAME;
public string REPOSITORY;
}
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create File Watcher Initiated Workflow or Custom Workflow

AVEVA Work Tasks allows users to create Custom event provider initiated workflows. To implement it, first a custom event provider should be created.

After successfully creating a Custom Event Provider, we will see the steps for how to create an option for FileWatcher initiated WorkFlow.

Step 1:

1. Place an image icon under *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**BPMUITemplates**Default**Repository**ListControl**images**icons-ribbon* folder. This icon will be displayed in the ribbon bar for associating with custom event provider.
2. Add an entry in the *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**BPMUITemplates**Default**Repository**ListControl**WorkflowRibbonBar.ascx* page as shown below.

   The following code should be added to the location where other event based workflow icons are defined (like Forms initiated WF). Find out the exact location and add the below code with the other similar kind of TDs.

   ```
   <td height="25" align="left">
    <span><a href="javascript:ExecuteAction('_sys_newfilewatcherworkflow')"
   class="sectionitem">
    <img alt="<%=GetLangSpecText("ec_rbn_wf_bamworkflow_alttext")%>"
   src="<%=templateThemeDirectory%>Repository/ListControl/images/icons-ribbon/custom.PNG"
   width="23" height="23" align="absmiddle" border="0" />
    <%=GetLangSpecText("Customworkflow") %>
    </a></span>
   </td>
   ```

   The above code will display the icon for the custom initiated workflow and it is linked to a javascript function.

   If you check the WorkFlow list now, we can see that a new icon has been added to the ribbon bar.

   The functionality for the new icon will be implemented in the following steps.
3. Create a workflow form aspx page in *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**BPMUITemplates**Default**Plugins**FileWatcher**Workflow* directory. That aspx name can be specified in the function coming up in the next step.
4. To add a List Definition execute the following code by creating a project which has reference to the Workflow.net.dll and has the following using statements.
   - using Skelta.Repository.List;
   - using Skelta.Repository.List.Action;

     Given below is the complete code file to get the function executed. The function InsertFileWatcherWorkflow() adds the list definition. This function accepts repository name as its parameter.

     ```
     using System;
     using System.Collections.Generic;
     using System.ComponentModel;
     using System.Data;
     using System.Drawing;
     using System.Text;
     using System.Windows.Forms;
     using Workflow.NET;
     using Skelta;
     ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using Skelta.Repository.List;
using Skelta.Repository.List.Action;


namespace WindowsApplication1
{
public partial class Form1 : Form
{
public Form1()
{
InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
//The function to add a new
 InsertFileWatcherWorkflow("W4ELoadRepo");
}
public void InsertFileWatcherWorkflow(string RepositoryName)
{

ListDefinition objList = new ListDefinition(new
Skelta.Core.ApplicationObject(RepositoryName), "Workflow");
ActionNewWorkflow newFileWatcherWorkflow = new ActionNewWorkflow();
newFileWatcherWorkflow.Id = "_sys_newfilewatcherworkflow";
newFileWatcherWorkflow.Name = "ecm_list_cm_newfilewatcherworkflow";
newFileWatcherWorkflow.ShowWhenSecurityAllowsRights = string.Empty;
newFileWatcherWorkflow.ValidOnlyForItemStatus = string.Empty;
//newFileWatcherWorkflow.ValidOnlyForItemTypes = "0";
newFileWatcherWorkflow.PagePath = "../../Plugins/FileWatcher/Workflow/
NewWorkflowForm.aspx";
//newFileWatcherWorkflow.ItemType = 1;
newFileWatcherWorkflow.ItemType = 5;
newFileWatcherWorkflow.Title = "New FileWatcher Workflow";
newFileWatcherWorkflow.Type = ListActionType.Task;
newFileWatcherWorkflow.WindowType = ListUIActionWindowTypes.Popup;
newFileWatcherWorkflow.WindowPosition = ListUIActionWindowPosition.Center;
newFileWatcherWorkflow.WindowSize = ListUIActionWindowSize.ScreenPropotional;
newFileWatcherWorkflow.DisplayValidRightValues =
"ActionCreateFileWatcherWorkflow<True>";
newFileWatcherWorkflow.ActionAllowValidRightValues =
"ActionCreateFileWatcherWorkflow<True>";
newFileWatcherWorkflow.ApprovalRequiredValidRightValues = string.Empty;
newFileWatcherWorkflow.ApprovalRequiredWithPromptValidRightValues = string.Empty;
// myList.Add(newFileWatcherWorkflow);
objList.Operations.Actions.Add(newFileWatcherWorkflow);
objList.Save();

}
}
}
```

Execute the following code once.

5. Run the script given below. Make an entry in the SKESecurityRigths Table.

```
-- First you need to make an entry in the SKESecurityRigths Table
DECLARE @Id AS UNIQUEIDENTIFIER
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
DECLARE @RepositoryName AS VARCHAR(100)
DECLARE @ParentItemId AS UNIQUEIDENTIFIER
DECLARE @UserId AS UNIQUEIDENTIFIER
DECLARE @EmptyId AS UNIQUEIDENTIFIER
DECLARE @TableId AS UNIQUEIDENTIFIER


SET @RepositoryName = 'Dec10Repo' -- Specify the Required Repostiory Name
DECLARE @Count as INT
Select @Count = Count(Id) from SKESecurityRights where
RightCode='ActionCreateFileWatcherWorkflow' and application=@RepositoryName
IF(@Count = 0)
BEGIN
 SET @Id = newid()
 SET @EmptyId = NULL
 SELECT @ParentItemId=Id,@UserId=CreatedBy,@TableId=TableId FROM SKESecurityRights WHERE
RightCode='WorkflowListRights' and Application=@RepositoryName

 INSERT INTO [dbo].[SKESecurityRights]
 ([Id],[Application],[CreatedBy],[CreatedDateTime],[LastUpdatedDateTime],[Title],[Descri
ption],[WIP],[WIPItemId],[MainItemId]
 ,[ParentItemId],[RecordParentItemId],[TableId],[ItemRowIndex],[IsLatest],[Version],[Ver
sionHistory],[Status],[LockedBy],[LockedOn]
 ,[ListID],[SecuritySettings],[ItemId],[ItemType],[SecurityItemId],[Owner],[ModifiedBy],
[RightCode],[Scope],[IsApplicableToFollower]
 ,[ControlType],[ControlFollowerType],[TypeFlag],[FollowerTypeFlag],[AuditTrail],[IsDisa
bled],[WorkflowStatus],[IsStandardRight]
 ,[SecurityCustomizationId])
  VALUES
 (@Id,@RepositoryName,@UserId,getdate(),getdate()
 ,'Can Create FileWatcher
Workflow',NULL,0,NULL,@Id,@ParentItemId,NULL,@TableId,0,1,NULL,'|',3,NULL,NULL,NULL,NULL
 ,@Id,1,@EmptyId,@UserId,@UserId
 ,'ActionCreateFileWatcherWorkflow','ListSpecificForListLevel',1,'AllowDisallow','AllowD
isallow',1,1,NULL,0,'',1,@EmptyId)


 -- Next making entry into the SKERightsListMapping table

 Declare @RightId as uniqueidentifier
 Declare @ListId as Uniqueidentifier

 select @ListId = Id from SKEListDefinition WHERE Name='Workflow' and
Application=@RepositoryName

 SET @Id = newid()

 SELECT @RightId = Id FROM SKESecurityRights
 WHERE RightCode = 'ActionCreateFileWatcherWorkflow' AND Application=@RepositoryName

 INSERT INTO [dbo].[SKERightsListMapping]
 ([Id], [RightId], [ListID], [Application])
  VALUES
 (@Id, @RightId, @ListId, @RepositoryName)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 -- Next making entries in the SKESGRightsMapping table for different security groups
 -- Getting Default Security group id's
 Declare @Admin as UniqueIdentifier
 Declare @Contributor as uniqueIdentifier
 declare @Reader as uniqueidentifier
 declare @LimitedAccess as uniqueidentifier

 select @Admin = Id from SKESecurityGroup where Title='Administrator' and
application=@RepositoryName
 select @Contributor = Id from SKESecurityGroup where Title='Contributor' and
application=@RepositoryName
 select @Reader = Id from SKESecurityGroup where Title='Reader' and
application=@RepositoryName
 select @LimitedAccess = Id from SKESecurityGroup where Title='Limited Access' and
application=@RepositoryName


 SET @Id = newid()
 INSERT INTO [dbo].[SKESGRightsMapping]
 ([Id],[SecurityGroupId],[RightId],[ListID],[ListItemId],[Value]
 ,[FollowerValue],[EffectiveSGCustomizationId],[Application])
  VALUES
 (@Id, @Admin, @RightId, @EmptyId, @EmptyId
 ,'true', 'true', @EmptyId, @RepositoryName)

 SET @Id = newid()
 INSERT INTO [dbo].[SKESGRightsMapping]
 ([Id],[SecurityGroupId],[RightId],[ListID],[ListItemId],[Value]
 ,[FollowerValue],[EffectiveSGCustomizationId],[Application])
  VALUES
 (@Id, @Contributor, @RightId, @EmptyId, @EmptyId
 ,'true', 'true', @EmptyId, @RepositoryName)

 SET @Id = newid()
 INSERT INTO [dbo].[SKESGRightsMapping]
 ([Id],[SecurityGroupId],[RightId],[ListID],[ListItemId],[Value]
 ,[FollowerValue],[EffectiveSGCustomizationId],[Application])
  VALUES
 (@Id, @Reader, @RightId, @EmptyId, @EmptyId
 ,'false', 'false', @EmptyId, @RepositoryName)

 SET @Id = newid()
 INSERT INTO [dbo].[SKESGRightsMapping]
 ([Id],[SecurityGroupId],[RightId],[ListID],[ListItemId],[Value]
 ,[FollowerValue],[EffectiveSGCustomizationId],[Application])
  VALUES
 (@Id, @LimitedAccess, @RightId, @EmptyId, @EmptyId
 ,'false', 'false', @EmptyId, @RepositoryName)
END
ELSE
BEGIN
 PRINT ('Entries already existing for Can Create AX Workflow right in the repository')
END
```

After all the steps to create the custom event provider initiated workflow are completed, verify the functionality

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

by clicking the custom icon in the top menu. This should pop-up a new window where we can create a workflow. As we have illustrated with the filewatcher example, our custom event provider is file watcher only.

Click **Save & Continue** which will lead to the Process Designer where we can design the workflow. Publish the newly created workflow and try generating the event (creating a .doc file in this case) that is set to trigger the workflow.

Code Sample - File Watcher Initiated Workflow

The below is the Aspx file for the File Watcher initiated WorkFlow

```
<%@ Page Language="C#" AutoEventWireup="true"
Inherits="Skelta.Repository.CodeBehind.NewFileWatcherWorkflow" %>
<%@ Register Assembly="Skelta.Forms.Web" Namespace="Skelta.Forms.Web" TagPrefix="cc1" %>
<%@ Register Assembly="RadAjax.Net2" Namespace="Telerik.WebControls" TagPrefix="radAjax" %>
<%@ Register Assembly="RadInput.Net2" Namespace="Telerik.WebControls" TagPrefix="radInput"
%>
<%@ Register Assembly="RadCombobox.Net2" Namespace="Telerik.WebControls" TagPrefix="radC"
%>
<%@ Register Assembly="RadWindow.Net2" Namespace="Telerik.WebControls" TagPrefix="radW" %>
<%@ Register Assembly="RadCalendar.Net2" Namespace="Telerik.WebControls" TagPrefix="radCln"
%>
<%=_docType %>
<link href="<%=cssPath2%>Common/StyleSheet/Global.css" rel="stylesheet" type="text/css" />
<html>
<head id="Head1">
<title>
<%=rs.GetString("ec_filewatcher_Workflow")%>
</title>

<script language="javascript" type="text/javascript">

function OpenRadWindowOverDue()
{
var WorkflowName = document.getElementById("<%=hidWorkflow.ClientID%>").value;

var ApplicationName = document.getElementById("<%=hidApplication.ClientID%>").value;
var oWindow1 = radopen("../EventProvider/BAMOverDueLookup.aspx?ApplicationName="
+ApplicationName+"&WorkflowName=" + WorkflowName, null);
oWindow1.SetModal(true);

oWindow1.SetSize(465,408);
oWindow1.IconUrl="..\images\Icon-ListExplorer.png";
oWindow1.ClientCallBackFunction  = fillOverDue;
}

function OpenRadWindowKpi()
{
var WorkflowName = document.getElementById("<%=hidWorkflow.ClientID%>").value;
var ApplicationName = document.getElementById("<%=hidApplication.ClientID%>").value;
var oWindow1 = radopen("../EventProvider/BAMKpiLookup.aspx?ApplicationName="
+ApplicationName+"&WorkflowName=" + WorkflowName, null);
oWindow1.SetModal(true);

oWindow1.SetSize(455,400);
oWindow1.IconUrl="..\images\Icon-ListExplorer.png";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
oWindow1.ClientCallBackFunction  = fillKpi;
}

function fillOverDue(radWindow, returnlistid)
{
var hidSlots =  document.getElementById("<%=hidSlots.ClientID%>");
hidSlots.value = returnlistid;
}

function fillKpi(radWindow, returnlistid)
{
var hidKpis =  document.getElementById("<%=hidKpis.ClientID%>");
hidKpis.value = returnlistid;
}
function ReloadPage()
{

}

function SizeToFit()
{
 var oWnd = GetRadWindow();
 oWnd.SetSize(700,580);
}
function GetRadWindow()
{
var oWindow = null;
 oWindow = window.frameElement.radWindow;
return oWindow;
}
</script>

</head>
<body style="margin: 0; left: 0; overflow: hidden; height: 100%" onload="ReloadPage()">
<form id="form1" runat="server">
<div id="ParentDiv">
<br />
<asp:Panel ID="Panel1" runat="server" Height="100%">
<div id="HeaderDiv">
<table width="94%" align="center" border="0">
<tr>
<td colspan="3">
<img src="<%=cssPath2%>/Repository/ListControl/images/icons-ribbon/icon-FileWatcher-
Workflow.png"
hspace="4" align="absmiddle" />
<span class="pagetitle">
<%=rs.GetString("ec_filewatcher_NewWorkflow", true)%>
</span>
<br />
<span class="description" colspan="2">
<%= rs.GetString("ec_filewatcher_desc_new", true)%>
</span>
</td>
</tr>
</table>
</div>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<br />
<div id="BodyDiv" style="overflow: hidden;">
<table width="94%" cellpadding="7" cellspacing="1" border="0" class="tablebg" width="94%"
align="center">
<tr>
<td class="lefttdbg" width="50%">
<span class="subtitle">
<%=rs.GetString("ec_filewatcher_PlcTitle", true)%>
<img runat="server" id="imgMandatory" src="<%$ReplaceTemplateExpn:Skeltaforms/Images/
mandatoryicon.PNG%>"
visible="true" hspace="2" align="top" />
<asp:RequiredFieldValidator ID="ReqSiteUrl" runat="server"
ControlToValidate="WorkflowNameInput"
Display="Dynamic" Enabled="true">
<img id="imgsiteUrlerror" alt="" runat="server" src="<%$ ReplaceTemplateExpn:Skeltaforms/
Images/erroricon.png%>"
hspace="2" align="top" visible="true" />
</asp:RequiredFieldValidator>
<asp:CustomValidator ID="WorkflowNameValidator" runat="server"
ControlToValidate="WorkflowNameInput"
SetFocusOnError="true" Display="Dynamic" ValidateEmptyText="true"
OnServerValidate="WorkflowNameValidator_ServerValidate"
Enabled="true">
<img id="imgCustom" alt="" runat="server" src="<%$ ReplaceTemplateExpn:Skeltaforms/Images/
erroricon.png%>"
hspace="2" align="top" visible="true" />
</asp:CustomValidator>
</span>
<br />
<span class="description">
<%=rs.GetString("ec_filewatcher_EnterWfName", true)%>
</span>
</td>
<td class="righttdbg" width="50%">
<radInput:RadTextBox ID="WorkflowNameInput" runat="server" Skin="SkeltaInput"
CausesValidation="true"
MaxLength="100" Width="50%" RadControlsDir="<%$ReplaceTemplateExpn:telerik/radcontrols/%>">
</radInput:RadTextBox>
</td>
</tr>
<tr>
<td class="lefttdbg" width="50%">
<span class="subtitle">
<%=rs.GetString("ec_filewatcher_PlcDesc", true)%>
</span>
<br />
<span class="description">
<%=rs.GetString("ec_filewatcher_EnterWfDesc", true)%>
</span>
</td>
<td class="righttdbg" width="50%">
<radInput:RadTextBox ID="WorkflowDescriptionInput" Skin="SkeltaInput" runat="server"
TextMode="MultiLine" Width="99%" Height="55px"
RadControlsDir="<%$ReplaceTemplateExpn:telerik/radcontrols/%>">
</radInput:RadTextBox>
</td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>
</table>
<table width="94%" align="center" border="0">
<tr>
<td width="50%">
<img src="<%=imgPath2%>/icon-associate-events.png" hspace="4" align="absmiddle" /><span
class="pagetitle">
<%=rs.GetString("ec_filewatcher_Associate", true)%>
</span>
<br />
<span class="description" colspan="2">
<%= rs.GetString("ec_filewatcher_new_desc", true)%>
</td>
</tr>
</table>
<radW:RadWindowManager runat="server" Behavior="Default" SingleNonMinimizedWindow="true"
RadControlsDir="<%$ReplaceTemplateExpn:telerik/radcontrols/%>" ID="RadWindowManager1"
Skin="SkeltaWindow" VisibleStatusbar="true" IconUrl="..\images\Icon-ListExplorer.png">
</radW:RadWindowManager>
<table width="94%" cellpadding="7" cellspacing="1" border="0" class="tablebg" width="94%"
align="center">
<tr>
<td class="lefttdbg" width="50%">
<span class="subtitle">
<%=rs.GetString("ec_filewatcher_Comboeventtype",true )%>
</span>
<img id="imgType" alt="" runat="server" src="<%$ ReplaceTemplateExpn:Skeltaforms/Images/
erroricon.png%>"
hspace="2" align="top" visible="false" />
</td>
<td class="righttdbg" width="50%">
<asp:DropDownList ID="cmbEventType" runat="server" AutoPostBack="false"
CssClass="inputselect"
Width="200px">
</asp:DropDownList>
</td>
</tr>
<tr>
<td class="lefttdbg" width="50%">
<span class="subtitle">
<%=rs.GetString("ec_filewatcher_Comboeventfolder")%>
</span>
<img id="imgFolder" alt="" runat="server" src="<%$ ReplaceTemplateExpn:Skeltaforms/Images/
erroricon.png%>"
hspace="2" align="top" visible="false" />
</td>
<td class="righttdbg" width="50%">
<radC:RadComboBox ID="cmbFolderList" runat="server" Skin="SkeltaCombobox"
RadControlsDir="<%$ReplaceTemplateExpn:telerik/radcontrols/%>"
MaxLength="100" DropDownWidth="283px" Height="100px" NoWrap="false" AutoPostBack="false">
</radC:RadComboBox>
</td>
</tr>
<tr>
<td class="lefttdbg" width="50%">
<span class="subtitle">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<%=rs.GetString("ec_filewatcher_filefullpath")%>
</span>
<br />
<span class="description">
<%=rs.GetString("ec_filewatcher_Pattern")%>
</span>
</td>
<td class="righttdbg" width="50%">
<asp:TextBox ID="txtFilePattern" runat="server" CssClass="inputtext"
Width="50%"></asp:TextBox>
<br />
<span class="description"></span>
</td>
</tr>
<tr id="TrSharePointEventsNote" runat="server">
<td class="lefttdbg" colspan="3">
<img id="Img1" runat="server" src="<%$ ReplaceTemplateExpn:/Common/Images/icon-note.png%>"
align="absmiddle">
<span class="note">
<%=rs.GetString("_NewWFForm_EvtNoteTitle", true)%>
<%=rs.GetString("_NewWFForm_EvtNotes", true)%>
</span>
</td>
</tr>
<input type="hidden" id="hidApplication" runat="server" />
<input type="hidden" id="hidSlots" runat="server" />
<input type="hidden" id="hidWorkflow" runat="server" />
<input type="hidden" id="hidKpis" runat="server" />
</table>
</div>
<div id="FooterDiv">
<table width="98%" border="0" cellpadding="3">
<tr>
<td colspan="1" align="right" valign="top">
<asp:Button ID="SaveButton" runat="server" CssClass="inputbutton" />
</td>
</tr>
<tr>
<td height="50px">
</td>
</tr>
</table>
</div>
</asp:Panel>
</div>
</form>
</body>
</html>

<script type="text/javascript">

function AddEventHandler(elementObject, eventName, functionObject)
{
if(document.addEventListener)
elementObject.addEventListener(eventName, functionObject, false);
else
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if(document.attachEvent)
elementObject.attachEvent("on" + eventName, functionObject);
}

AddEventHandler(window,"resize",resizefunction);
AddEventHandler(window,"load",resizefunction);

function resizefunction()
{
 var bodydiv = document.getElementById('BodyDiv');
 var headerdiv = document.getElementById('HeaderDiv');
 var footerdiv = document.getElementById('FooterDiv');
var parentdiv = document.getElementById('ParentDiv');
if (parentdiv.offsetParent==document.body)
 document.body.style.height = '100%';

 var parentdivheight = parentdiv.offsetParent.offsetHeight;
parentdiv.style.height =  parentdivheight ;
 try{bodydiv.style.height = parentdivheight - headerdiv.offsetHeight -
footerdiv.offsetHeight ;}catch(e){}//headerdiv.offsetHeight -
 bodydiv.style.width =  '100px';
 bodydiv.style.width =  parentdiv.clientWidth;
bodydiv.style.overflow =  'auto';
 parentdiv.style.overflow = 'hidden';
}
</script>
```

The below is the code file for the File watcher initiated WF aspx file.

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using Skelta.Forms.Core.Controls;
using Skelta.Repository.Web.List;
using Skelta.Repository.List;
using Skelta.Repository.List.Action;
using Skelta.Core;
using Workflow.NET;
using Telerik.WebControls;
using System.Collections.Generic;
using Workflow.NET.PropertyTypes;
using Skelta.Events;
using Skelta.Events.Interfaces;
using System.Text.RegularExpressions;
using Workflow.NET.Interfaces;
using Workflow.NET.Storage;
using System.Text;
using System.Xml;
using System.Xml.Schema;
using System.IO;
using Skelta.Repository.List.EventProvider;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using Skelta.Repository.Security;
public partial class NewWorkflowForm : System.Web.UI.Page
{
protected string TemplateDirectory;
protected SkeltaResourceSet rs = new SkeltaResourceSetManager().GlobalResourceSet;
protected string _docType = string.Empty;
protected string eventsComboId = string.Empty;
protected Workflow.NET.Config objConfig = new Config();
int _ItemTemplateWidth = 0;
string alertMessage = string.Empty;
protected Guid eventBindId;
protected Skelta.Repository.List.ListDefinition associationList;
protected Skelta.Repository.List.ListItem associationItem;
string _applicationName = string.Empty;
string workflowName = string.Empty;
string _LoggedInUser = string.Empty;
string versionStamp = string.Empty;
string _ListName = string.Empty;
string effectiveFromDate = string.Empty;
string effectiveToDate = string.Empty;
Guid parentItemId = Guid.Empty;
Guid listGuid = Guid.Empty;
protected Guid stPort;
string listName = string.Empty;
const string workflowListName = "Workflow";
const string workflowAssociationListName = "Event Associations";
public string error;
string EventType;
string FullPath = string.Empty;
protected Skelta.Repository.List.ListItemCollection licollection = null;
protected Skelta.Repository.List.ListDefinition listdefinition = null;
Skelta.Repository.List.ListItemCollection _menuItems;
ListDefinition _List;
ListDefinition List;
Hashtable columnNames = new Hashtable();
//Themes Implementation changes
protected string cssPath1 = "";
protected string cssPath2 = "";
protected string imgPath1 = "";
protected string imgPath2 = "";
protected object styleUrl1 = Workflow.NET.TemplateExpressionBuilder.GetUrl("SkeltaForms/
StyleSheet");
protected object styleUrl2 = Workflow.NET.TemplateExpressionBuilder.GetUrl("Repository/
Site/styles");
protected object imgUrl=Workflow.NET.TemplateExpressionBuilder.GetUrl("SkeltaForms/
Images");
public enum FileEventType
{
```

```
                Created = 0,
                Changed = 1,
                Renamed = 2,
                Deleted = 3

                }
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
protected void Page_Load(object sender, EventArgs e)
{
```

```
if ((this.Request.Browser.Browser.ToString().ToLowerInvariant() == "firefox") &&
(this.Request.Browser.Version.ToString() == "1.5"))
_docType = "<!DOCTYPE html PUBLIC \"-//W3C//DTD XHTML 1.0 Transitional//EN\"
\"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd\">";
else
_docType = "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 Transitional//EN\">";

//Themes Implementation changes
cssPath1 = styleUrl1.ToString();
cssPath2 = styleUrl2.ToString();
imgPath1=imgUrl.ToString();
imgPath2 = imgUrl.ToString();
WorkflowNameInput.EmptyMessage = rs.GetString("_NewWFForm_WorkflowNameInput_EmptyMsg");
SaveButton.Text = rs.GetString("_NewWFForm_SaveButton_Text");
Guid itemGuid = Guid.Empty;
ListPageParameters listParams = new ListPageParameters();

listParams.ResolveCurrentUri();
_LoggedInUser = listParams.LoggedInUserId;
_ListName = listParams.ListName;
_applicationName = listParams.ApplicationName;
LoadEventTypes();
cmbEventType.Width = 100;
cmbFolderList.Width = 300;
if (!this.Page.IsPostBack)
{
listGuid = listParams.ListId;
parentItemId = listParams.ParentItemId;
_applicationName = listParams.ApplicationName;
versionStamp = listParams.VersionStamp;

ViewState["itemGuid"] = itemGuid;
ViewState["listGuid"] = listGuid;
ViewState["parentItemId"] = parentItemId;
ViewState["ApplicationName"] = _applicationName;
InitializeComponent();
 // LoadPorts();
List = new ListDefinition(new Skelta.Core.ApplicationObject(_applicationName), "FileWatcher
List");
LoadSecurityEnblePorts();
}
else
{
itemGuid = (Guid)ViewState["itemGuid"];
listGuid = (Guid)ViewState["listGuid"];
parentItemId = (Guid)ViewState["parentItemId"];
_applicationName = ViewState["ApplicationName"].ToString();
}

_List = new ListDefinition(new Skelta.Core.ApplicationObject(_applicationName), listGuid);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
workflowName = Request["WorkflowName"];
workflowName = (workflowName == null) ? "" : workflowName.ToString();
SaveButton.Click += new EventHandler(SaveButton_Click);
}

private void LoadEventTypes()
{
string EventTypesSelectedValue = cmbEventType.SelectedValue;
this.cmbEventType.DataSource = Enum.GetNames(typeof(FileEventType));
this.cmbEventType.DataBind();
cmbEventType.Items.Insert(0, "--Select--");
if (EventTypesSelectedValue != null && EventTypesSelectedValue != string.Empty)
this.cmbEventType.SelectedValue = EventTypesSelectedValue;
}

void Alert(string message)
{
Page.ClientScript.RegisterStartupScript(this.GetType(), "alertwindow", "<script>alert('" +
message.Replace("'", "\\'") + "');</script>");
}


private void InitializeComponent()
{
try
{
string applicationName = "";
string workflowname = "";
Guid ListItemId = Guid.Empty;

ListPageParameters listParams = new ListPageParameters();
listParams.ResolveCurrentUri();
applicationName = listParams.ApplicationName;

if (listParams.ListName.ToLower() == "workflow")
ListItemId = listParams.ListItemId;

ListDataHandler ObjListDataHandler = new ListDataHandler(listParams.ApplicationName,
"Workflow");
Skelta.Repository.List.ListItem wfitem;

ViewState["ApplicationName"] = applicationName;
ViewState["UserId"] = listParams.LoggedInUserId;

objConfig = new Config(ViewState["ApplicationName"].ToString().Trim());


}
catch (Exception)
{

}

}

protected string GetLangSpecText(string keyName)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
SkeltaResourceSetManager resManager = new SkeltaResourceSetManager();
return resManager.GlobalResourceSet.GetString(keyName);
}


protected void WorkflowNameValidator_ServerValidate(object source, ServerValidateEventArgs
args)
{
WorkflowNameValidator.ErrorMessage = string.Empty;
if (WorkflowNameInput.Text == string.Empty)
{
args.IsValid = false;
WorkflowNameValidator.ErrorMessage = rs.GetString("_NewWFForm_WorkflowNameInput_EmptyMsg");
}
else
{
string patternMatch = @"[#~&+\\""'<>|]";
Regex reg = new Regex(patternMatch, RegexOptions.CultureInvariant);
if (Regex.IsMatch(args.Value, patternMatch))
{
args.IsValid = false;
WorkflowNameValidator.ErrorMessage = GetLangSpecText("ec_filewatcher_Workflow_Name");
}
else
{
args.IsValid = true;
}
}
}


/// <summary>
/// Method to load filewatcher ports
/// </summary>
private void LoadPorts()
{
Dictionary<string, IEventPort> eventPorts = GetAllEventPorts(_applicationName,
"FileWatcher");
Telerik.WebControls.RadComboBoxItem lstIt = new Telerik.WebControls.RadComboBoxItem();
lstIt.Text = GetLangSpecText("ec_filewatcher_option");
lstIt.Value = "0";
cmbFolderList.Items.Add(lstIt);
foreach (string portGuid in eventPorts.Keys)
{
Telerik.WebControls.RadComboBoxItem lst = new Telerik.WebControls.RadComboBoxItem();
lst.Text = eventPorts[portGuid].PortName;
lst.Value = eventPorts[portGuid].PortGuid.ToString();
cmbFolderList.Items.Add(lst);
}

 }
/// <summary>
/// New Method to load all filewatcher ports which secuity enabled.
/// </summary>
```

private void LoadSecurityEnblePorts()

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
DataTable dtTable = null;

Skelta.Repository.List.ListItemCollection objSecPorts = new Skelta.Repository.List.ListItemCollection(List);
dtTable = objSecPorts.GetRecordsForConsume(_LoggedInUser, Skelta.Repository.Display.AllItems);
if (dtTable.Rows.Count > 0)
 LoadPorts();



}
 /// <summary>
/// Check whether the events are selected or not. If it is not selected eventbinding table
saving part skiped.
/// </summary>
/// <returns></returns>
private bool EventsSelected()
{
bool flag = false;
try
{
if (cmbEventType.SelectedIndex != 0 && cmbFolderList.SelectedIndex != 0)
{
flag = true;
}
else if ((cmbEventType.SelectedIndex != 0 && cmbFolderList.SelectedIndex == 0) ||
(cmbEventType.SelectedIndex == 0 && cmbFolderList.SelectedIndex != 0))
{
flag = false;
error = GetLangSpecText("ec_filewatcher_portnotempty");
}
else
{
flag = true;
}

}
catch (Exception)
{
}
return flag;
}

/// <summary>
/// Method to get all filewatcher folders
/// </summary>
/// <param name="_applicationName"></param>
/// <param name="eventProviderName"></param>
/// <returns></returns>
private Dictionary<string, IEventPort> GetAllEventPorts(string _applicationName, string
eventProviderName)
{
//If it is not given here, for every serviceprovider class, geteventports method should be
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
implemented.
Dictionary<string, IEventPort> colEventPorts = new Dictionary<string, IEventPort>();
if (_applicationName != string.Empty)
{
EventServicePortCollection colEventServicePorts = new EventServicePortCollection(new
ApplicationObject(_applicationName), eventProviderName);
foreach (EventServicePort evntPort in colEventServicePorts.Items)
colEventPorts.Add(evntPort.Name, evntPort.EventPort);
}
return colEventPorts;
}


/// <summary>
/// Button click event
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void SaveButton_Click(object sender, EventArgs e)
{
string err = WorkflowNameValidator.ErrorMessage.ToString().Trim();

if (cmbEventType.SelectedValue  == "--Select--" && cmbFolderList.SelectedValue != "0")
{
error = GetLangSpecText("ec_filewatcher_event");
err = "Error";
return;

}
if (cmbEventType.SelectedValue != "--Select--" && cmbFolderList.SelectedValue == "0")
{
error = GetLangSpecText("ec_filewatcher_Port");
err = "Error";
return;
}

if (err == string.Empty)
{

Skelta.Repository.List.ListItem item = new Skelta.Repository.List.ListItem(_List,
"<Workflow Title=\"" + WorkflowNameInput.Text.ToString().Trim() + "\" Description=\"" +
WorkflowDescriptionInput.Text.ToString().Trim() + "\" />");
item.ParentItemId = parentItemId;
item.LoggedInUserId = _LoggedInUser;
item.ItemType = 5;
try
{
item.Save();
}
catch (Exception ex)
{
err = item.LastSaveError;
Alert(err.Trim());
return;
}
//if any exeception there at creating the item it will be stored in LastSaveError property
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
of ListItem Object.it may not throw any exception hence check for this
if (item.LastSaveError != null)
{
err = item.LastSaveError;
Alert(err.Trim());
return;
}

Guid WFId = item.Id;
string workflowName = WorkflowNameInput.Text.ToString().Trim();
versionStamp = item.CurrentVersion.VersionStamp;
if (string.IsNullOrEmpty(versionStamp))
{
versionStamp = "1";
}

ListPageParameters listParams = new ListPageParameters();

listParams.ListId = listGuid;
listParams.ApplicationName = _applicationName;
listParams.ListName = _ListName;
listParams.ListItemId = WFId;
listParams.LoggedInUserId = _LoggedInUser;

string uri = listParams.GetSecuredUri();

bool blAssociate = false;

if (EventsSelected() == true)
{
if (HasSecurityRights() == true)
{
SaveEventBinding();
}
else
{
return;
}
}
else
{
return;
}

CreateXMLVaribale();
if (item.LastSaveError == null)
{

Response.Redirect("../../../Repository/Site/WorkflowTasks.aspx?" + uri + "&WFName=" +
workflowName, true);
}
Page.ClientScript.RegisterStartupScript(typeof(NewWorkflowForm), "",
"<script>this.close();</script>");
}
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
private bool HasSecurityRights()
{
bool blRight = true;
try
{
//Check whether it is required to check for event association
//Save the Event Bindings here
ListDataHandler ldh = new ListDataHandler(_applicationName, "Workflow");
Skelta.Repository.List.ListItem _ListItem = ldh.GetListItem(WorkflowNameInput.Text.Trim());
_ListItem.LoggedInUserId = _LoggedInUser;
//Commented by Hemanth to get SecuityFlag Information from SecurityManager class
//Skelta.FarmManager.Repository repository = new
Skelta.FarmManager.Repository(_applicationName);
bool securityflag = SecurityManager.GetSecurityFlag(_applicationName);
if (securityflag == true) //repository.Properties["enablesecurity"].ToLowerInvariant()
{
//Checking whether the user has rights to add an item in association list in workflow list

Skelta.Repository.List.ListDefinition skWorkflowList = new
Skelta.Repository.List.ListDefinition(new Skelta.Core.ApplicationObject(_applicationName),
"Workflow");
Skelta.Repository.Security.SecurityManager securityMgr = new
Skelta.Repository.Security.SecurityManager(skWorkflowList, _LoggedInUser);

if ((securityMgr.GetValueForRightCode("ActionCreateAssociations",
Skelta.Repository.EntityType.ListItem, _ListItem.Id)).ToUpperInvariant() != "TRUE")
{
blRight = false;
SecurityRightsHandler SecurityHandler = new SecurityRightsHandler(_applicationName);
ListAction action = _ListItem.List.Operations.GetActionById("_sys_delete");
SecurityHandler.ExecuteAction(action, _ListItem, ListItemEvents.OnDelete);
Alert("You dont have permission to create the associations.");
 // return;
}

bool blRightProvider = true;
if (blRight)
{
Skelta.Repository.List.ListDefinition eventProvidersList = new
Skelta.Repository.List.ListDefinition(new Skelta.Core.ApplicationObject(_applicationName),
"Event Providers");
Skelta.Repository.List.ListItemCollection eventProvidersCollection = new
Skelta.Repository.List.ListItemCollection(eventProvidersList);
System.Data.DataTable providersTable =
eventProvidersCollection.GetRecordsForConsume("Title", "FileWatcher",
Skelta.Repository.Display.MainItems, _LoggedInUser);
if (providersTable.Rows.Count == 0)
{
blRight = false;
SecurityRightsHandler SecurityHandler = new SecurityRightsHandler(_applicationName);
ListAction action = _ListItem.List.Operations.GetActionById("_sys_delete");
SecurityHandler.ExecuteAction(action, _ListItem, ListItemEvents.OnDelete);
Alert("You dont have permission to consume the File Watcher Event Provider. So cannot
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
create association.");
 // return;
}
}


}

}
catch (Exception Ex)
{
throw Ex;
}
return blRight;
}


/// <summary>
/// Method to save the entry to the event binding table
/// </summary>
private void SaveEventBinding()
{
try
{
string filename;
string filepath;
String eventtype;
FullPath = txtFilePattern.Text;
EventType = cmbEventType.SelectedItem.Text;
```

```
ListDataHandler wfhandler = new ListDataHandler(_applicationName, "Workflow");
Skelta.Repository.List.ListItem wfitem;
wfitem = wfhandler.GetListItem(WorkflowNameInput.Text);


workflowName = WorkflowNameInput.Text.Trim();

EventBinding eventBind;

eventBind = new EventBinding(new Skelta.Core.ApplicationObject(_applicationName));
eventBind.BindingType = EventBindingType.Workflow;
eventBind.Disabled = false;
eventBind.Disable_After_Expiry = false;
eventBind.NoofEvents = 0;

eventBind.EventName = "FileWatcher";
eventBind.Alert_ApplicationName = _applicationName;
eventBind.Alert_Workflow = workflowName;//the current list item
eventBind.Alert_Version = versionStamp;
eventBind.Alert_UserId = _LoggedInUser;
int flag = 0;
if (cmbFolderList.SelectedValue != "0")
{
bool chkFolder = true;

string strPort = cmbFolderList.SelectedItem.Text;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
DirectoryInfo drInfo = new DirectoryInfo(strPort);
if (!drInfo.Exists)
{
string[] strFolders = strPort.Split('\\');
if (strFolders.Length > 0)
{
for (int i = 0; i < strFolders.Length; ++i)
{
if (strFolders[i] == "DESKTOP")
{
chkFolder = true;
break;
}
}
if (strFolders[0] == string.Empty)
{
chkFolder = true;
}

}


}

if (chkFolder == true)
{

stPort = new Guid(cmbFolderList.SelectedValue);

eventBind.PortGuid = stPort;
flag = 1;
}
else
{
error = GetLangSpecText("ec_filewatcher_portRemoved");

}
}
else
{
error = GetLangSpecText("ec_filewatcher_portnotempty");
}
if (flag == 1)
{
string[] strFileType = txtFilePattern.Text.Split('.');
if (strFileType.Length > 1)
{
if (strFileType[0] == string.Empty && strFileType[1] != string.Empty)
{
eventBind.str2 = strFileType[1].ToString();
}
if (strFileType[0] != string.Empty && strFileType[1] != string.Empty)
{
if (strFileType[0] == "*" && strFileType[1] == "*")
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
else if (strFileType[0] == "*" && strFileType[1] != string.Empty)
{
eventBind.str2 = strFileType[1];
}
else
{
eventBind.str1 = txtFilePattern.Text;
}

}
}
else if (strFileType[0] != string.Empty)
{
eventBind.str1 = strFileType[0];
}

eventBind.str3 = EventType;

 //
eventBind.Save();

eventBindId = eventBind.Id;

filename = txtFilePattern.Text;
filepath = cmbFolderList.SelectedItem.Text;
eventtype = EventType;
alertMessage = filepath + " " + filename + " " + eventtype;
StringBuilder eventXml = new StringBuilder();
eventXml.Append("<EventBind id=\"" + eventBindId + "\">");
eventXml.Append("<ApplicationName>" + _applicationName + "</ApplicationName>");
eventXml.Append("<Workflow>");
eventXml.Append("<WorkflowName>" + workflowName + "</WorkflowName>");
eventXml.Append("<WorkflowAlert>" + alertMessage + "</WorkflowAlert>");
eventXml.Append("<WorkflowEffectiveFrom>" + effectiveFromDate +
"</WorkflowEffectiveFrom>");
eventXml.Append("<WorkflowEffectiveTo>" + effectiveToDate + "</WorkflowEffectiveTo>");
eventXml.Append("</Workflow>");
eventXml.Append("<Parameters>");
eventXml.Append("<File FileName=\"" + filename + "\" name=\"" + listName + "\"></File>");
eventXml.Append("<File FilePath=\"" + filepath + "\" name=\"" + listName + "\"></File>");
eventXml.Append("<File EventType=\"" + eventtype + "\" name=\"" + listName + "\"></File>");
eventXml.Append("</Parameters>");
eventXml.Append("</EventBind>");
SaveAssociationList(eventXml.ToString());
}
}
catch (Exception)
{

}
}
/// <summary>
/// Method to add the eventassociation table entry
/// </summary>
/// <param name="eventXml"></param>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
private void SaveAssociationList(string eventXml)
{
if (associationList == null)
{
associationList = new ListDefinition(new Skelta.Core.ApplicationObject(_applicationName),
"Event Associations");
}
if (associationList != null)
{

//Get the reference id from the main list []

Guid webApplicationItemReferenceId = Guid.Empty;

ListDefinition webApplicationList = new ListDefinition(new
Skelta.Core.ApplicationObject(_applicationName), "Event Associations");
//File association details Details
Skelta.Repository.List.ListItemCollection webApplicationListItems = new
Skelta.Repository.List.ListItemCollection(webApplicationList);
Hashtable webApplicationTableColumnNames = new Hashtable();
SortedDictionary<string, Skelta.Forms.Core.Controls.BaseDataControl> fields =
webApplicationListItems.GetRecordsFieldMapping();
foreach (KeyValuePair<string, Skelta.Forms.Core.Controls.BaseDataControl> info in fields)
{
webApplicationTableColumnNames.Add(info.Value.Name, info.Key);
}

//Create a new record If web application does not exist on that list

Skelta.Repository.List.ListItem webApplicationListItem = null;
if (webApplicationItemReferenceId == Guid.Empty)
webApplicationListItem = new Skelta.Repository.List.ListItem(webApplicationList);
else
webApplicationListItem = new Skelta.Repository.List.ListItem(webApplicationList,
webApplicationItemReferenceId);

if (webApplicationListItem != null)
{
if (associationItem == null)
{
associationItem = new Skelta.Repository.List.ListItem(associationList);
}

ListMainForm associationMainForm = (ListMainForm)associationItem.ListForm.Records[0];
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_evntbndxml")).Value
= eventXml;
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_wrkflwname")).Value
= workflowName;
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_evntprvdrname")).Va
lue = "FileWatcher";
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_evnttype")).Value =
"FileWatcher";
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_evntstatus")).Value
= "New";
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_desc")).Value =
alertMessage;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
((ListTextDataItem)associationMainForm.FindControlByID("_sys_sk_evntass_title")).Value =
EventType;
((ListGuidDataItem)associationMainForm.FindControlByID("_sys_sk_evntassbnd_id")).Value =
eventBindId;
associationItem.LoggedInUserId = _LoggedInUser;
associationItem.Save(ListItemVersionStatus.Published);
Page.ClientScript.RegisterClientScriptBlock(typeof(Page), "regJs", "<script>alert('File
association saved successfully');</script>");
}
else
{
throw new Exception("Not able to fetch main Reference Id");
}
}

}




private void CreateXMLVaribale()
{
try
{

workflowName = WorkflowNameInput.Text.Trim();
Skelta.Core.AddInProvider provider = new Skelta.Core.AddInProvider(new
Skelta.Core.ApplicationObject(_applicationName), GetProviderId("XmlDocument",
_applicationName));
Workflow.NET.Interfaces.IXmlStorageType storageHandler =
(Workflow.NET.Interfaces.IXmlStorageType)provider.CreateInstance(true);

Workflow.NET.Web.Designer.ProcessDesigner pd = new
Workflow.NET.Web.Designer.ProcessDesigner();
pd.ID = "pd";
pd.ApplicationName = _applicationName;
pd.WorkflowName = workflowName;
pd.FileName = "1";
pd.UseTopmostWindowForRenderingOnClientSide = false;

pd.Height = System.Web.UI.WebControls.Unit.Percentage(100);
pd.Width = System.Web.UI.WebControls.Unit.Percentage(100);
pd.InitializeValues();


Workflow.NET.Diagram _diagram =
Workflow.NET.Diagram.GetDiagramCreateIfNotFound(_applicationName, workflowName,
versionStamp);
_diagram.PreProcessDiagaram();
bool flag = true;
foreach (XmlVariable var in _diagram.XmlVariables.Values)
{
if (var.Name.Equals("FileWatcherXMLData"))
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
flag = false;
}
}
if (flag)
{
Workflow.NET.XmlVariable XVariable = new Workflow.NET.XmlVariable();
XVariable.Name = "FileWatcherXMLData";
XVariable.Description = "File watcher XML Data";
XVariable.StorageType = "XmlDocument";

string eventdataFilePath =
System.Web.HttpContext.Current.Server.MapPath(System.Web.HttpContext.Current.Request.Applic
ationPath + "/" + Workflow.NET.Config.GetTemplateDirectoryFromWebConfig() + "Default/
XMLSchemas/FileWatcherEventData.xml");

XVariable.Schema = ParseXMLForData(eventdataFilePath);

XVariable.StorageHandler = storageHandler;
((Workflow.NET.XmlStorageType.XmlDocument)(XVariable.StorageHandler)).StorageData =
"variables";
_diagram.XmlVariables.Add(XVariable.Name, XVariable);
 commonFunctions.WriteDiagram(_diagram, _applicationName, workflowName, versionStamp);
_diagram.Dispose();

Workflow.NET.Designer workflowDesigner = new Workflow.NET.Designer(_applicationName,
workflowName, "1");
workflowDesigner = new Workflow.NET.Designer(_applicationName, workflowName, "1");
Workflow.NET.XmlVariable eventDataVariable =
Skelta.Events.SKEventData.GetXmlVariable(_applicationName, workflowName, "1");
workflowDesigner.XMLVariables.Add(eventDataVariable.Name, eventDataVariable);
workflowDesigner.WriteToDefinition();
}
}
catch (Exception)
{

}


}
private Guid GetProviderId(string storageType, string applicationName)
{
Skelta.Core.AddInProviderCollection _Provider = new Skelta.Core.AddInProviderCollection(new
Skelta.Core.ApplicationObject(_applicationName));

string FilterExp = "where Type =@Type and Name=@Name";

Skelta.Core.DS.SkeltaParameter param1 = new Skelta.Core.DS.SkeltaParameter("@Type",
"XmlStorageType");
Skelta.Core.DS.SkeltaParameter param2 = new Skelta.Core.DS.SkeltaParameter("@Name",
storageType);

System.Data.DataSet DsProvider = _Provider.GetMatchingRecords(FilterExp, param1, param2);
string s = "";
if (DsProvider.Tables[0].Rows.Count > 0)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
s = DsProvider.Tables[0].Rows[0].ItemArray.GetValue(0).ToString();
}
return new Guid(s);
}



private string ParseXMLForData(string file)
{

TextWriter txt = new StringWriter();


StreamReader sReader = new System.IO.StreamReader(file);

System.Xml.XmlTextReader reader = new System.Xml.XmlTextReader(sReader);

XmlDocument XDoc = new XmlDocument();

XDoc.Load(reader);

reader.Close();

sReader.Close();



StringReader stringReader1 = new StringReader(this.AddNameSpacePrefixIntoHolder(XDoc));

XmlTextReader xmlTextReader1 = new XmlTextReader(((TextReader)stringReader1));

XmlSchemaSet schemaSet = new XmlSchemaSet();
XmlSchemaInference schema = new XmlSchemaInference();

schemaSet = schema.InferSchema(xmlTextReader1);


foreach (XmlSchema s in schemaSet.Schemas())
{
s.Write(txt);
}
string xsdstring = txt.ToString();

return xsdstring;

}
private string AddNameSpacePrefixIntoHolder(XmlDocument oxmlDoc)
{
foreach (XmlNode xmlNode1 in oxmlDoc.ChildNodes)
{
if (xmlNode1.NamespaceURI.Length > 0)
{
this.AddNameSpaceHolder(xmlNode1.NamespaceURI, xmlNode1.Prefix);
}
if (xmlNode1.ChildNodes.Count > 0)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
string string2 = this.FindNextNode(oxmlDoc, xmlNode1);
}
}
return oxmlDoc.OuterXml;
}
private string FindNextNode(XmlDocument oxmlDoc, XmlNode oNode)
{
string string1 = null;
foreach (XmlNode xmlNode1 in oNode.ChildNodes)
{
if (xmlNode1.NamespaceURI.Length > 0)
{
this.AddNameSpaceHolder(xmlNode1.NamespaceURI, xmlNode1.Prefix);
if (xmlNode1.ChildNodes.Count > 0)
{
string string2 = this.FindNextNode(oxmlDoc, xmlNode1);
}
}
else
{
if (xmlNode1.ChildNodes.Count > 0)
{
string string3 = this.FindNextNode(oxmlDoc, xmlNode1);
}
continue;
}
}
return string1;
}
public void AddNameSpaceHolder(string NameSpaceURI, string Prefix)
{
Hashtable nameSpaceHolder = new Hashtable();
bool b1 = false;
NameSpaceInfo nameSpacePrefix1 = null;
Exception exception1 = null;
IEnumerator iEnumerator1 = null;
try
{
b1 = false;
if (nameSpaceHolder.Count <= 0)
{
nameSpacePrefix1 = new NameSpaceInfo();
nameSpacePrefix1.NameSpaceURI = NameSpaceURI;
nameSpacePrefix1.NameSpacePrefix = Prefix;
nameSpaceHolder.Add(Prefix, nameSpacePrefix1);
goto L_00DA;
}
try
{
iEnumerator1 = nameSpaceHolder.Values.GetEnumerator();
while (iEnumerator1.MoveNext())
{
nameSpacePrefix1 = ((NameSpaceInfo)iEnumerator1.Current);
//if (StringType.StrCmp(nameSpacePrefix1.NameSpaceURI.Trim().ToUpperInvariant(),
NameSpaceURI.Trim().ToUpperInvariant(), false) == 0)
//{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//  b1 = true;
//goto L_0093;
//}
}
}
finally
{
if (iEnumerator1 is IDisposable)
{
((IDisposable)iEnumerator1).Dispose();
}
}

L_0093:
{
}
if (b1)
{
return;
}
nameSpacePrefix1 = new NameSpaceInfo();
nameSpacePrefix1.NameSpaceURI = NameSpaceURI;
nameSpacePrefix1.NameSpacePrefix = Prefix;

if (!nameSpaceHolder.Contains(Prefix))
nameSpaceHolder.Add(Prefix, nameSpacePrefix1);
return;

L_00DA:
{
}
}
catch (Exception exception2)
{
exception1 = exception2;
throw new Exception(exception1.Message);
}
}
}
```

Import BPMN Process Models from Microsoft Visio

The Business Process Modeling Notation (BPMN) is a published standard that enables standardization in the notations used for business process modeling.

Overview of Visio Process Import

BPMN Compatibility

The Business Process Modeling Notation (BPMN) is a published standard that enables standardization in the notations used for business process modeling. It defines a comprehensive set of notations for different elements in a process model such as actions, events and links. This in turn makes it possible for business process models

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

created in one BPM application to be used in another, provided both applications support, and are compatible with BPMN standards.

Support for Process Models created using IT Pearls BPMN Stencil for Microsoft Visio

The IT Pearls BPMN Stencil for Microsoft Visio from ITP-Commerce adds process modeling functionality to Visio. Business Analysts can use this familiar tool to model their business processes and export them as XML files, which in turn can be used in applications that automate those processes. AVEVA Work Tasks has a provision to import and integrate the process models exported from Visio and execute them.

Import of XML file Exported from Microsoft Visio

The import functionality (found in the Process Designer's Settings menu) has a new option to import Visio Process Designs in addition to the earlier option to import existing Processes. Selecting this option displays the open file dialog box. You can select the Visio process file here and import it. AVEVA Work Tasks will map the BPMN events and activities to corresponding activities and recreate the process design. In case the Visio event or activity does not have a corresponding activity then it will be retained in the redrawn process design as a BPMN event or activity.

There is only UI provided for conversion of Visio workflow to AVEVA Work Tasks. AVEVA Work Tasks provides API, which gives the entire structure of workflow definition within AVEVA Work Tasks. These APIs can be used to import workflow definition built in other tool after converting the same to AVEVA Work Tasks understandable format.

Note: From Visio, you cannot import workflow XML which is generated by AVEVA Work Tasks. The XML generated by AVEVA Work Tasks is native which can be easily understood. The third party tool is required to convert the same to Visio to understand the format.

Realigning the Imported Process Model

The imported process will have to be realigned to ensure that it reflects the source Visio process model. The Realign Activity is available to enable this process. You can execute this Activity to automatically realign the imported process design.

Handling Unlinked Process Flows in the Visio Process Model

The original Visio process model may have had process flows that were not linked to any other event/activity. During import into the Process Designer these process flows are linked to the Start activity. An appropriate message will be displayed on the Import window, mentioning the activity name and stating that it was found unlinked in the Process Model and so it is being linked to the Start activity by default. Such process flows need to be manually linked to the appropriate point in the main process flow.

Note: Importing "Visio 2013 exported file" is not supported.

Map BPMN Events to AVEVA Work Tasks Actions

You can map BPMN End Events to AVEVA Work Tasks. The End Event represents the type of event that stops a process flow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

BPMN End Events

The End Event represents the type of event that stops a process flow. This event can have several possible results such as Message, Timer, Exception, Link, etc. It can also end the process flow without generating any result.

End Event

BPMN End Event

The End Event represents the type of event that stops a process flow. This event can have several possible results such as Message, Timer, Exception, Link, etc. It can also end the process flow without generating any result. The End Event is denoted by a circle with a thick outline. It is empty if the event does not have any result. If there are results they are represented by different symbols for each result attribute, which will be superimposed on the End Event symbol.

The handling of this event with different results during the import of a process model into AVEVA Work Tasks is described in subsequent topics in this section.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The BPMN End Event is mapped to the End Activity, which is represented by a circle with a thick outline, filled with red, with the image of a stop sign superimposed on it to denote the end.

The following mapping will be used during the conversion:

- Only the End Events without any result (without any result attributes superimposed) or with the Message result attribute (represented by an envelope) superimposed will be mapped to the End Activity.
- The End Event with the Link result attribute will be automatically handled by the Link Originator connector and converted to a physical link.
- In AVEVA Work Tasks, the End Activity will not do anything and will be treated as the logical end of the execution thread of the process. (In the case of processes running in parallel this would mean that the thread which reaches the End Activity will end.)

  Note: The End Activity is different from the Terminate Activity.

- Use of this Activity is optional and has been provided mainly for modeling purposes. Even if this activity is not placed at the end of the process thread, it will conclude if there are no further links to execute.

Validation

In AVEVA Work Tasks, outlinks (links originating from the activity) are not allowed from the End Activity.

Raise Exception

BPMN End Event with Exception

The End Event with Exception represents the type of event that stops a process flow and generates an exception as the result. It is denoted by the symbol for electric current (indicating hazard) superimposed on the End Event

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

symbol, i.e. the circle with thick outline.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The End Event with the Exception result attribute is mapped to the Raise Exception Activity, which is represented by a red circle similar to the End Activity superimposed with the symbol for electric current.

The following mapping will be used during the conversion:

- Only End Events with the Exception result attribute and no outlinks will be mapped to the Raise Exception Activity. It is important that the End Event with Exception have no outlinks to be mapped.
- On reaching this activity a named exception will be generated, which will have to be caught by the Exception Handler(s) of the workflow.

Validation

Outlinks (links originating from the activity) are not allowed from the Raise Exception Activity.

Cancel Transaction

BPMN End Event with Cancel

The End Event with Cancel represents the type of event that stops a process flow and cancels the transaction that is running at that time. It is denoted by an 'X' superimposed on the End Event symbol, i.e. the circle with thick outline.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The End Event with the Cancel result attribute is mapped to the Cancel Transaction Activity, which is represented by a red circle, similar to the End Activity, superimposed with an 'X'.

The following mapping will be used during the conversion:

- Only the End Events with the Cancel result attribute and no outlinks will be mapped to the Cancel Transaction Activity.
- On reaching this activity the current running transaction is canceled (rolled back).
- The parameter for this Activity is the transaction name.

Validation

- Outlinks (links originating from the activity) are not allowed from the Cancel Transaction Activity.
- This activity can be used only within a transaction block.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Compensate Transaction

BPMN End Event with Compensation

The End Event with Compensation represents the type of event that stops a process flow, cancels the transaction that is running at that time, generates the compensation event, and performs a full roll back. It is denoted by a left-pointing double arrowhead symbol (similar to the symbol for rewind) superimposed on the End Event symbol, i.e. the circle with thick outline.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The End Event with the Compensation result attribute is mapped to the Compensate Transaction Activity, which is represented by a red circle similar to the End Activity superimposed with a left-pointing double arrowhead symbol.

The following mapping will be used during the conversion,

- Only the End Events with the Compensation result attribute and no outlinks will be mapped to the Compensate Transaction Activity.
- On reaching this activity the current running transaction is canceled, the compensation event is generated, and full rollback is performed.

Validation

- Outlinks (links originating from the Activity) are not allowed from the Cancel Transaction Activity.
- This activity can be used only within a transaction block.

Terminate Workflow

BPMN End Event with Terminate

The End Event with Terminate represents the type of event that kills a workflow process. It is denoted by a filled circle superimposed on the End Event symbol, i.e. the circle with thick outline.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The End Event with the Terminate result attribute is mapped to the Terminate Workflow Activity, which is represented by a red circle similar to the End Activity superimposed with a solid red circle similar to the corresponding BPMN notation.

The following mapping will be used during the conversion:

- Only the End Events with the Terminate result attribute and no outlinks will be mapped to the Terminate Workflow Activity.
- Terminate Workflow action takes the Workflow ID as an optional parameter. If no ID is specified the currently executing workflow is terminated.
- If the Workflow ID is specified the corresponding workflow is killed immediately.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Validation

Outlinks (links originating from the activity) are not allowed from the Terminate Workflow Activity.

Link Source

BPMN End Event with Link

The End Event with Link represents the type of event that creates a link. It is denoted by a right-pointing arrow superimposed on the End Event symbol, i.e. the circle with thick outline.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The End Event with the Link result attribute is mapped to the Link Source Activity, which is represented by a circle similar to the End Activity but filled with yellow and superimposed with a right-pointing arrow similar to the corresponding BPMN notation.

The following mapping will be used during the conversion:

- Only the End Events with the Link result attribute and no outlinks will be mapped to the Link Source Activity.
- The Link ID is mapped to the Link Identifier parameter of the Link Source Activity, which is used by the Link Target Activity to make the connection.

Validation

- Outlinks (links originating from the activity) are not allowed from the Link Source Activity.
- Link Identifier is mandatory.

End With Message

BPMN End Event with Message

The End Event with Message represents the type of event that generates a message after ending the process flow. It is denoted by an envelope symbol superimposed on the End Event symbol, i.e. the circle with thick outline.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The End Event with the Message result attribute is mapped to the End with Message Activity, which is represented by a red circle similar to the End Activity superimposed with an envelope symbol similar to the corresponding BPMN notation. The End with Message Activity is identical to the Email Activity, and is used to end the process logically while notifying some user about the end.

The following mapping will be used during the conversion:

Only the End Events with the Message result attribute and no outlinks will be mapped to the End with Message Activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Validation

Outlinks (links originating from the activity) are not allowed from the End with Message Activity.

Multiple (Not Mapped)

BPMN End Event with Multiple results

The End Event with Multiple results represents the type of event that generates a number of results after ending the process flow. It is denoted by a six-pointed star symbol superimposed on the End Event symbol, that is the circle with thick outline.

The Multiple End Event is resolved in AVEVA Work Tasks having different results mapped to different End Events in parallel.

BPMN Intermediate Events

Intermediate Events are usually placed at the boundary of an activity. They denote the triggers that can interrupt the activity. The activity will stop executing and the process flow will be rerouted from the Event. The different types of Intermediate Event are Timer, Exceptions, Messages.

Intermediate Events Overview

BPMN Intermediate Events

Intermediate Events are usually placed at the boundary of an activity. They denote the triggers that can interrupt the activity. The activity will stop executing and the process flow will be rerouted from the Event. The different types of Intermediate Event are Timer, Exceptions, Messages. The Intermediate Event is represented by a circle with a double outline. It is empty if the event does not have any result. If there are results they are represented by different symbols for each result attribute, which will be superimposed on the Intermediate Event symbol.

On Exception

BPMN Intermediate Event with Exception

The Intermediate Event with Exception represents the type of event that interrupts a process flow when an exception occurs. It is denoted by the symbol for electric current (indicating hazard) superimposed on the Intermediate Event symbol, that is the circle with the double outline.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The Intermediate Event with the Exception trigger is mapped to the On Exception Activity, which is represented by the standard notation for Intermediate Events, that is a circle with a double outline filled with orange, with the symbol for electric current superimposed on it to denote the exception.

The following mapping will be used during the conversion:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Intermediate Event with Exception trigger will be mapped to the On Exception Activity.
- If the Target attribute is specified then the activity should not be mapped to the Wait for Exception activity; instead, it should be converted to an On Exception as the Trigger for the Link.

Validation

On Exception activity may be allowed to have no inlinks; in such a case it will be immediately executed after the Start Activity before the Start Activity proceeds to the next logical link.

Wait for Event

BPMN Intermediate Event with Message

The Intermediate Event with Message represents the type of event that interrupts a process flow when a notification/message is received. It is denoted by the envelope symbol superimposed on the Intermediate Event symbol, that is. the circle with the double outline.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The Intermediate Event with the Message trigger is mapped to the Wait for Event Activity, which is represented by the standard notation for Intermediate Events, that is a circle with a double outline filled with orange, with the envelope symbol superimposed on it.

The following mapping will be used during the conversion:

- Intermediate Event with Message trigger will be mapped to the Wait for Event Activity.
- If the Target attribute is specified then the activity should not be mapped to the Wait for Event activity; instead, it should be converted to a Link with Wait for Event as the Trigger for the event.

Validation

Wait for Event activity may be allowed to have no inlinks; in such a case it will be immediately executed after the Start Activity before the Start Activity proceeds to the next logical link.

Wait Timer

BPMN Intermediate Event with Timer

The Intermediate Event with Timer represents the type of event that interrupts the process flow at a scheduled time. It is denoted by a clock symbol superimposed on the Intermediate Event symbol, that is the circle with the double outline.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The Intermediate Event with the Timer trigger is mapped to the Wait Timer Activity, which is represented by the standard notation for Intermediate Events, that is a circle with a double outline filled with orange, with the clock

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

symbol superimposed on it.

The following mapping will be used during the conversion:

- Intermediate Event with Timer trigger will be mapped to the Wait Timer Activity.
- If the Target attribute is specified then the activity should not be mapped to the Wait Timer activity; instead, it should be converted to a Link with Wait Timer as the Trigger for the event.

Validation

Wait Timer activity may be allowed to have no inlinks; in such a case it will be immediately executed after the Start Activity before the Start Activity proceeds to the next logical link.

On Transaction Canceled

BPMN Intermediate Event with Cancel

The Intermediate Event with Cancel represents the type of event that interrupts the process flow when a transaction is canceled. It is denoted by an 'X' superimposed on the Intermediate Event symbol, that is the circle with the double outline.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The Intermediate Event with the Cancel trigger is mapped to the On Transaction Canceled Activity, which is represented by the standard notation for Intermediate Events, that is a circle with a double outline filled with orange, with an 'X' superimposed on it.

The following mapping will be used during the conversion:

- Intermediate Event with Cancel trigger will be mapped to the On Transaction Canceled Activity.
- If the Target attribute is specified then the activity should not be mapped to the On Transaction Canceled activity; instead, it should be converted to a Link with On Transaction Canceled as the Trigger for the event.

Validation

- On Transaction Canceled action may be allowed to have no inlinks; in such a case it will be immediately executed after the Start Activity before the Start Activity proceeds to the next logical link.
- On Transaction Canceled must be placed in a process that is marked as transactionable.
- On Transaction Canceled, if used as an event trigger, must be attached to a task or activity that is marked as transactionable.

On Compensation

BPMN Intermediate Event with Compensation

The Intermediate Event with Compensation represents the type of event that interrupts the process flow when a transaction compensation event is encountered. It is denoted by a left-pointing double arrowhead symbol

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

superimposed on the Intermediate Event symbol, that is the circle with the double outline.

Mapping of Visio BPMN Event to the AVEVA Work Tasks Activity

The Intermediate Event with the Compensation trigger is mapped to the On Compensation Activity, which is represented by the standard notation for Intermediate Events, that is a circle with a double outline filled with orange, with a left-pointing double arrowhead symbol superimposed on it.

The following mapping will be used during the conversion:

- Intermediate Event with Compensation trigger will be mapped to the On Compensation Activity.
- If the Target attribute is specified then the activity should not be mapped to the On Compensation activity; instead, it should be converted to a Link with On Compensation as the Trigger for the event.

Validation

- On Compensation activity may be allowed to have no inlinks; in such a case it will be immediately executed after the Start Activity before the Start Activity proceeds to the next logical link.
- On Compensation must be placed in a process that is marked as transactional.
- On Compensation, if used as an event trigger, must be attached to a task or activity that is marked as transactional.

Map BPMN Activities to AVEVA Work Tasks Activities

You can map BPMN activities to AVEVA Work Tasks activities. The BPMN Task Activity is represented by a rounded rectangle shape. It is used to denote a logical task and may not do anything at runtime.

BPMN Task

BPMN Task Activity

The BPMN Task Activity is represented by a rounded rectangle shape. It is used to denote a logical task and may not do anything at runtime. For actual execution or implementation it has to be replaced by an appropriate human activity or any other suitable action in the AVEVA Work Tasks process definition.

This action will have the following properties:

- Task Type value, which is a string property where the appropriate value could be typed
- Marker Attributes, which will be used to show appropriate markers on the task shape
  - Loop Type: One, Standard, Multiinstance
  - Adhoc Task: Yes/No
  - Compensating Task: Yes/No
- Possible Outputs, which is a grid of strings
- Force Output During Run, which determines if the output link is to be traversed during run time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Mapping of Visio BPMN Notation to AVEVA Work Tasks Activities

- The imported BPMN Task Activity is generally converted to an appropriate action in AVEVA Work Tasks.
- The following mapping will be used during the conversion:
  - Receive Task will be converted to Host Web Service and Wait Call Activity.
  - Send Task will be converted to Call Web Service Activity.
  - User Task will be converted to Task Activity.
  - Script Task will be converted to Script Activity.
  - Manual Task will remain a BPMN Task activity within the process.
  - Reference Task will also remain a BPMN Task activity within the process.
  - Any other value will be mapped to see if a corresponding activity exists in AVEVA Work Tasks and if it does the said Activity will be created during import.
  - In case the value does not match any of the above, it will remain a BPMN Task within AVEVA Work Tasks.

Validation

No specific validations will be performed on the BPMN Task Activity in AVEVA Work Tasks.

Sub-Process

BPMN Sub-Process

The BPMN Sub-Process can be of two types -

- Expanded, which is represented by the Task symbol superimposed with its own set of process items forming a separate process flow.
- Collapsed, which is represented the Task symbol with a node symbol [+] on the lower boundary.

Mapping of Visio BPMN Notation to AVEVA Work Tasks Activities

While importing a sub-process from the Visio BPMN Modeler, it will be created as a separate process in AVEVA Work Tasks, and will be referenced to the parent process automatically.

- Sub-Processes will be treated as separate Process Definitions within AVEVA Work Tasks with a Sub-Process Marker in the Process Definition Storage and the definition.
- Sub-Processes will automatically get access to all the Variables and Content of the parent process.
- Sub-Processes cannot be invoked directly since their contextual information will be completely dependent upon the parent process.
- The Sub-Process Task will be mapped to the Workflow Activity, which will be mapping to above defined Sub-Process.
- The Sub-Process Activity will provide a way to easily jump to a sub-process view by using right-click on the properties.
- A Sub-Process will be able to return to a particular output by setting a variable called Sub Process_Output

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

and based on this it can be accessed directly within the parent process flow.

- An Expanded Sub-Process is mapped to a Workflow Activity that is assigned the name of the Sub-Process. The elements inside the sub-process are mapped to the appropriate AVEVA Work Tasks activities and a new XML process definition is created with all these elements.
- A Collapsed Sub-Process is mapped to a Workflow Activity with the name of the Sub-Process assigned to the Workflow activity.

Map Connectors

You can map connectors in AVEVA Work Tasks with a BPMN Sequence Flow. The BPMN Sequence Flow is denoted by a solid line with a terminal arrow. It represents the direction of process flow by linking other elements.

Sequence Flows

BPMN Sequence Flow

The BPMN Sequence Flow is denoted by a solid line with a terminal arrow. It represents the direction of process flow by linking other elements.

Mapping of AVEVA Work Tasks BPMN Element

The Sequence Flow is identical to the AVEVA Work Tasks Link Activity and it will be mapped to the appropriate Link Activity based on the source element and target element.

Message Logs

The following functions are available in using message logs:

- Setting the debug mode level
- Viewing message log files
- Using the logger console
- Disabling logging of strings

Set Message Log Debug Mode Level

You can specify the debug mode in **Farm Configuration Wizard**. The level of detail in the information that is shown in the Logger Console depends on the value that is set for this property. The Workflow Execution information is displayed in the Logger Console.

The debug levels are:

- Level 1 shows minimal information in the Logger Console, giving only information on the start and end of the workflow. Error messages are also shown in this mode. AVEVA Work Tasks recommends this Debug Level on production environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Level 2 shows detailed information of what is happening in the background. It will show all the messages logged from the actions.
- Level 3 shows only messages indicating whether the Activity is executed.

Log Message Categories

There are three categories of messages logged in the log files.

- **Information** – This type of message is preceded by a lower case 'i'. It indicates messages containing general information about the events being logged. Each message of this type has four parts – the event information, the module generating the event, the machine on which the event took place, and the date and time of the event.
- **Warning** – This type of message is preceded by a question mark '?'. It indicates messages containing warning notes to the users about the events being logged. The messages of this type have a structure similar to the 'Information' messages with the same four parts mentioned earlier.
- **Error** – This type of message is preceded by an exclamation mark '!'. It indicates messages containing details about errors that occur. The messages of this type have the four parts found in the other two message categories. In addition it has the 'Exception Stack Trace', which occurs before the machine name and date/time. It can be identified by the keyword 'Source' and it can be used to trace the source of the error.
- **LogConfirmation** - This displays confirmation messages and such messages are prefixed c -.

View Message Log Files

In this section you will see how to view and use the Log files created by AVEVA Work Tasks. Two types of Log files are created.

- The first type of Log file, for generic log messages, will be created in the following path:

  [AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\\Logs

  The Log files are saved in this folder using the date of logging as the filename with the format "YYYYMMDD" (there will be one Log file for each day).
- The second type of Log file is application-specific, it will log messages related to applications. These Log files are created in separate folders corresponding to each application. The folders are located in the following path:

  [AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Logs

  These log files are saved with names in the format "<application name>_YYYYMMDD".

To view the Log files

1. Browse to the location of the log files using the relevant path, that is, "\Logs" or "\Logs\<application name>" depending on the type of Log file you wish to view. Select the log file to view by looking for the date of logging.
2. Open the log file in a text editor. It will display all the events of the day using the three types of log

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

categories to identify the type of information.

Logger Console

In this section you will see how to view and use the Logger Console that is part of AVEVA Work Tasks. The Logger Console is a utility that shows a real-time log of events during the execution of AVEVA Work Tasks. It is automatically installed in the *C:\Program files\AVEVA\Work Tasks\\Bin* folder when AVEVA Work Tasks is installed.

You can also access Logger Console by selecting the **Logger** option from the App bar menu items at the top-left corner of the application window.

1. Browse to the location of the Logger Console and double-click the Logger Console executable.
2. This opens the Logger Console log in a new window. You can view the events as they occur when AVEVA Work Tasks is running.
3. The Logger Console will display the events using the three types of log categories to identify the type of information.

Option to Disable Logging of Strings

You can control the string messages getting logged into the logger files without making any change to the debug level. Suppose your debug level is set as 2 and you want to restrict some messages containing text. For Example, **"Inserting local variables"** from getting logged into the logger file without changing your current debug level. Following are the steps to achieve it.

1. Create a New Key in AVEVA Work Tasks Registry named "DoNotLogMessages" of type Multi-String value.
2. Specify the text of the messages which are to be ignored while writing to the log files.
3. Restart services.

Manage and Synchronize Threads in Engine

Overview

This section provides information on how to manage and synchronize threads in AVEVA Work Tasks Engine.

Thread

A Thread is a small set of executable instructions, which can be used to isolate a task from a process. Multiple threads are efficient way to obtain parallelism of hardware and give interactive user interaction to your applications. Threading enables your C# program to perform concurrent processing so you can do more than one operation at a time. For example, you can use threading to monitor input from the user, perform background tasks, and handle simultaneous streams of input.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

ThreadPool Class

Provides a pool of threads that can be used to post work items, process asynchronous I/O, wait on behalf of other threads, and process timers. Thread pooling enables you to use threads more efficiently by providing your application with a pool of worker threads that are managed by the system. One thread monitors the status of several wait operations queued to the thread pool. When a wait operation completes, a worker thread from the thread pool executes the corresponding callback function.

The threads in the managed thread pool are background threads. This means that a **ThreadPool** thread will not keep an application running after all foreground threads have exited.

Engine and Threads

The engine is run as a Windows Service on AVEVA Work Tasks server installed machines. Once the service is started the engine listens on a particular port waiting for requests from clients to be processed. The communication between the engine and client (Example: w3wp process) is achieved using TCP/IP communication protocol. On a request from client the engine verifies the requests and places it on to a thread pool and then gives a notification back to client on successful delivery. From then on the engine processes the request in an asynchronous fashion. This is a bit different for synchronous execution which is detailed below.

The engine adopts two modes of execution for workflow steps or activities.

1. Asynchronous mode
2. Synchronous mode

For **Asynchronous** mode, the engine uses predefined set of threads (Custom thread pool) which will run/execute the workflow steps in parallell. Based on the number of threads available the number of workflow steps which can execute will vary. The number of threads for the custom thread pool is configurable and the performance on higher thread count will vary based on the operating system and hardware configuration.

For **Synchronous** mode, the TCP/IP connection between the client and engine is placed on hold and the client gets a reply only when a logical end point is reached within the engine. So the wait period for client to get a reply from engine on successful delivery will happen only after logical end is reached.

Note: On heavy load environment, it is always advisable to use asynchronous mode of execution.

Limit In-Memory Activity/Steps Count for Repository

You can control the number of **In Memory** workflow steps that can exist for any workflow instance. **In Memory** refers to the workflow steps which execute continuously in engine. For example, a workflow activity with a loop or a workflow which does not have any waiting steps like Human Activities.

This feature can be used to stop workflow instances with infinite looping behavior. All workflow instances which cross the **In Memory** activity count will be removed and gets marked as aborted.

SKInMemoryActivityCount Table

The table SKInMemoryActivityCount with columns **Application**, **Workflow**, and **InMemoryCount** is added to Farm Configuration database for configuring the **In Memory** concurrent steps. These settings will take effect only

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

when engine is restarted.

The default **In Memory** workflow steps for any workflow instance is 500. To change the default In Memory concurrent activity count, create an entry in *SKInMemoryActivityCount* table with application name as "defaultinmemorycount".

Thread Pools at Repository Level

Use this feature to set the default number of threads which will be running in background to execute the workflow steps for a repository. The feature can be very useful in a hosted environment where there will be predefined number of threads which will carry out the activity execution for each repository. So, if a workflow within a repository has time consuming steps it is in no way going to affect the workflows running in another repository.

SKConcurrentThreadPools Table

The table SKConcurrentThreadPools with columns Application and ThreadCount is added to farm configuration database for configuring the concurrent threads for executing workflow steps for a repository. These settings will take effect only when engine is restarted.

Note: If Service Monitoring is enabled, it is mandatory to restart Client Service which, in turn, restarts the engine. Hence, there is no need to restart the engine manually.

The default repository thread pool for executing workflow steps will have 30 threads running in parallel. To change the default application thread pool size, create an entry in SKConcurrentThreadPools table with application name as "DefaultPool".

Activity Concurrency Limit

Activity concurrency limit defines the feature to control the number of Activity Instances* or Activity Type which can run in parallel. Two tables are added to Farms database for this configuration. It is mandatory to restart the engine for these configuration changes to take effect.

SKActivityTypeConcurrency table: The columns available in the table are **Application**, **ActivityType** and **Concurrency Limit**. Two levels of concurrency can be set using this configuration.

- **Application and Activity Type level**: You can control the number of activity types which will run in parallel at the same time for a repository for a step type.
  Example: Within an application there can be only 10 script activity which will run in parallel.
- **Activity Type level**: You can control the number of activity types which will run in parallel at the same time.
  Example: Across all repositories there can be only 10 script activity which will run in parallel.

SKActivityNameConcurrency table: The columns available in the table are Application, Workflow, ActivityName, and ConcurrencyLimit. One level of concurrency can be set using this configuration.

Application, Workflow and ActivityName level: You can control the number of activity names (instances) which will run parallelly at the same time for a repository within a workflow.
Example: Within an application and workflow there can be only 10 MoveCopyFolder activity which will run in parallel.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The properties which can be configured in the farm database for Workflow Engine service is:

- **maxruntimethreadsforconcurrentexecution**: Number of threads which will be used to queue up the activities which timeouts the waiting periods to execute. The timeout activities will have Pending ('P') status for the activity in SWExecutionDetails table or BAM report.
- **concurrentactivityexecutionwaittime**: The wait time for an activity to start executing in synchronous mode. It is useful only when concurrency limits are set in farm configuration database for SKActivityTypeConcurrency or SKActivityNameConcurrency tables.

```
<properties>
<property key="loadbalanced"><![CDATA[False]]></property>
<property key="loadbalanceserverport"><![CDATA[8853]]></property>
<property key="loadbalanceclientport"><![CDATA[8859]]></property>
<property key="port"><![CDATA[8855]]></property>
<property key="maxenginethreads"><![CDATA[100]]></property>
<property key="maxruntimethreadsforconcurrentexecution"><![CDATA[20]]></property>
<property key="concurrentactivityexecutionwaittime"><![CDATA[15000]]></property>
<property key="lastupdateddatetime"><![CDATA[]]></property>
<property key="serviceexe"><![CDATA[WFEngine.NET2.exe]]></property>
<property key="displayname"><![CDATA[Skelta - Workflow Engine]]></property>
</properties>
```

Note: Based on the type of workflows, load and scenario, it is advisable to configure these parameters during load testing to reach an optimum number and performance. Activity Instance* is a single step which is in executing/running stage. For example, there are 2 **MoveCopy** steps named **MoveCopy1** and **MoveCopy** 2 placed in parallel with in a workflow. After the workflow is triggered, it would be one instance of **MoveCopy1** activity and one instance of MoveCopy2 activity running at the same time. If the workflow is triggered twice it would be two instance of MoveCopy1 activity and two instance of **MoveCopy2** activity running at the same time.

Entities

Application users are the most integral part of any system. The storage type used to store this user information can vary based on the Enterprise. Examples could be SQL database or Active Directory. AVEVA Work Tasks can be used to build Human centric workflow solutions. The users required to be part of this workflow solution can exist in different data stores.

A Repository contains Workflow Definitions, Queues, Calendars, Workflow execution data, and Business Activity Monitoring (BAM) data. The User Provider for the Repository is also defined while creating a Repository. Each Repository could potentially have its own user database, for example, one repository may connect to Active Directory, and the other may connect to theDatabases and others to Lightweight Directory Access Protocol (LDAP) or Active Directory Application Mode (ADAM).

The user integration needs to be outside the Repository boundary since users can span across multiple Repositories. AVEVA Work Tasks has the concept known as User Entity and Role Entity for managing users across different platforms.

User Entity manages all the information for the users across different Data sources. Data Source may be Active Directory, SQL providers, and so on. The Role Entity manages users in roles.

Data bridge framework is the common framework that bridges the entity item and the corresponding Data source providers. Each entity item will internally communicate with the Data bridge framework to know from which Data Source provider the entity details should be retrieved.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Basic Understanding of the Need for Data Bridge Interface

- Enterprise Console will be deployed as a central server, which can serve multiple applications simultaneously.
- Each application could potentially have its own Users Database. For example, one of the applications might be connecting to Active Directory, other might be connecting to Database, something to LDAP or ADAM.
- When AVEVA Work Tasks deals with each of these applications, it needs to ensure that users of these applications can work with AVEVA Work Tasks without any complicated configurations, and also in a secure manner which ensures only authorized functionality is available to each and every resource.
- Enterprise Console will also act as a Work Bench for all the workflow participants, hence it needs to allow user authentication from variety of sources. For example, an Application spanning internal as well as external vendors and customers, needs to have variety of database access requirements possible.
- Different applications might be using the same Authentication source, but might have different Metadata and grouping information with them. For example, the authentication source could be an active directory, but there could be five other applications that are using the same user and maintain application-specific user metadata. Hence, in reality if all the information is needed about the user it needs to be collected from across all the application as well as active directory.
- Sometimes it is expensive to query every time on the user database, so the results are cached heavily.
- The user integration is outside the Application boundary since the users can span across multiple applications.

DataBridge Provider

AVEVA Work Tasks follows a concept known as User Entity and Role Entity for managing users across different platforms.

User Entity manages all the information for the users across different data sources. Here data source may be active directory and SQL providers.

Role entity manages users in roles.

Data bridge framework is the common framework that bridges between the entity item & the corresponding data source providers. Here each entity item will internally communicate with the Data bridge framework to know from which data source provider the entity details should be retrieved.

User Entity

User Entity makes possible to use different data source providers to be used simultaneously to extend the Meta data and provide more number of unique user stores. This provider will maintain its own Universal Provider Actor table, which will map each data source provider and allow unique user ids to be stored in the mapping table while giving a unique user id which will be further used by the rest of AVEVA Work Tasks functionality.

Role Entity

Role entity allows getting all the users based on their roles. This will again communicate with the Data bridge frame work that will get the users in Roles based on the data source provider.

All the Role information will be stored in the SKVirtualRole table along with the role name and data source provider name.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

User Integration in AVEVA Work Tasks

To add users, all the user properties should be configured correctly. The configurations are done in the XML data. Here user properties are name of the user, email id, unique identifier and other user related details. Along with the user properties, the data source for users must be specified whether if the users are from the data base, active directory etc.

All the entity information is stored in the SKEntity and SKEntityDataSourceProvider tables.

SKEntity table stores all the user properties. These properties will be displayed in the UI for the user to query the resources.

SKEntityDataSourceProvider table stores the data source provider information for each of the entity item. Here all the user properties are defined and bounded to the user properties defined in the SkEntity table.

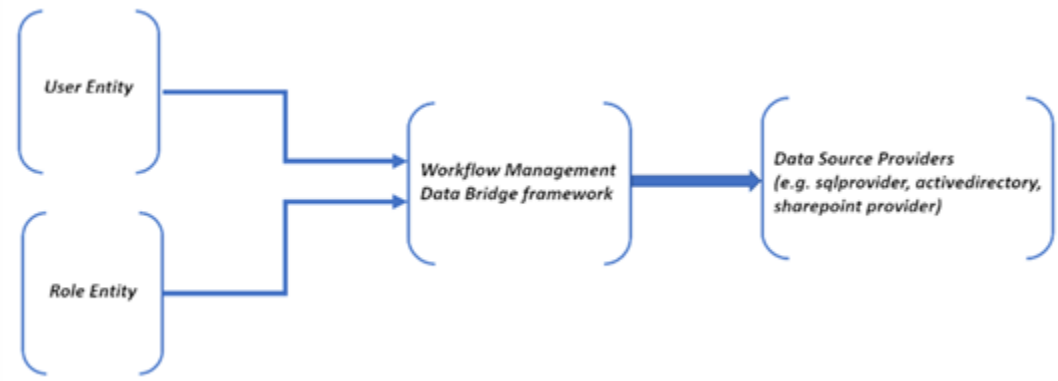
The VirtualActor table is used to map real user ids to virtual ids used by AVEVA Work Tasks.

User Entity

User Entity makes possible to use different data source providers simultaneously to extend the Meta data and provide more number of unique user stores. This provider will maintain its own Universal Provider Actor table, which will map each data source provider and allow unique user ids to be stored in the mapping table while giving a unique user id which will be further used by the rest of AVEVA Work Tasks functionality.

User Entity provider information will be stored in the **SKEntity** table. For each of this user entity, there will be unique settings that are stored in the DefinitionXml column. Along with these setting, repository name, mapping table info etc are stored. Settings for each user entity are stored in XML format. These include the user properties with details like the name of the property, its purpose, and display name. The mapping table used for storing the user information will be **SKVirtualActor** table. This table contains data related to user unique identifier, name, and email.

As shown in the image below, an entity can be mapped to different data source providers. For User Entity the providers mapped are called as user entity data source provider. The configuration for user entity data source provider is persisted in **SKEntityDataSourceProvider** table. For each of this user entity data source provider, there will be unique settings that are stored in the DefinitionXml column.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

XML Schema for User Entity and User Entity Data Source Provider

User Entity XML Schema for Definition XML

The XML Schema for user entity consists of collection of properties which defines the user. The properties defined for user entity is a clubbed collection of properties defined for the entity user data source providers.

Example

For example, let us consider the user data store to be SQL. The SQL server contains a database with table named as Users. Refer to the image below for the table columns specified which contain information regarding the user.

Table Name: Users

UserId	Name	Role	Email	Password
d51-036d0145c547	Michelle	Admin	michelle@skelta....	michelle
c6a7170a-6a35-4f79-8688-150d0ac8fe15	Kevin	Developer	kevin@skelta.com	kevin
ce24b3d0-21dc-4821-8d55-afe4c8581716	John	Developer	john@skelta.com	john
9d3eebb3-ebbe-4415-9d18-df4205af5c46	Elaine	Product Manager	elaine@skelta.com	elaine

The "UserId" column in the table contains the unique identifier value for the user. The "Name" and "Email" column contains the name & email information for the user. Let us see how the properties need to be defined in the definition XML for user entity & user entity data source for the above columns.

The User Entity Data Source Provider definition XML in **SKEntityDataSourceProvider** table will be as shown below:

```
 <dataproviders>
<provider name="authenticationuserprovider" type="Skelta List">
<properties>
<property  name="UserId"  purpose="identifier"  type="guid" />
<property  name="Name"  purpose="name"  type="string"/>
<property  name="Email"  purpose="email" type="string" />
<property  name="Role" type="string"/>
</properties>
</provider>
</dataproviders>
```

The XML nodes are **case sensitive** and should be added "as is". Please find below the description for the node and its attributes:

The **provider** node contains information about the type of data source provider used.

Attribute Name	Description
name	Defines the name of the data source provider type. This filed should match the name column value provided in SKAddInProvider table while configuring the entity data source provider. The names used for default provides shipped are given below. *Active Directory: activedirectory* *SQL Server: sqlprovider* *ASP.NET Membership: membershipprovider*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Attribute Name	Description
	AVEVA Work Tasks List : authenticationuserprovider
type	Defines the type of the provider. The value for this column can be any value which is descriptive. The default values used for the out of box providers are listed below. The default provider values should not be changed if used. *Active Directory: activedirectory* *SQL Server: sql server* *ASP.NET Membership: aspnet membership* *AVEVA Work Tasks List: Skelta List*
providerprefix	Value provided in this attribute will be prefixed to the unique identifier value when persisted in SKVirtualActor table. This value is referred only for data base operation like update or select on SKVirtualActor table and does not reflect in the UI when unique identifier value is referenced. AVEVA Work Tasks active directory provider uses this field to append domain name with the unique identifier value for AD [samAccountName is used as unique column in AD]. If this attribute is used it is assumed that the entity data source provider unique identifier column type is string. For the above sample this should not be used as the unique identifier type is Guid.
authenticationsource	Determines if the provider can be used as authentication source for logging in to EC. If set as false this provider does not get displayed for Log In option for EC.

This **property** node contains information related to the user. The property node is case sensitive. Each property represents a column/ field of the data source provider which contains the user information. For example, if the data source is SQL the property will represent table column and if it's AD the property column represents the active directory fields.

Attribute Name	Description
name	Column/ Field of the data source provider which defines a property of the user. In the above example name attribute value contains the column name of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	table which contains user information. UserId, Name, Email & Role are the columns defined in table "Users". **Note**: The name attribute value should contain only alphabets without spaces or special characters.
type	Defines the column type of the data base column / field. In the above example the UserId column is defined as guid type so the attribute type for property UserId is set as "guid".
purpose	Purpose defines the type of information available for a property (column/field). It's mandatory to configure purpose attribute for the properties. The must set purpose attributes are **identifier**: Mandatory fields required for user entity data source provider. This setting identifies the column from which the unique identifier value needs to be fetched. This setting is very critical for the data source provider configuration. The column value configured as identifier gets persisted in UserIDString or UserID or UserGUID column of SKVirtualActor table, based on the type. name: This setting identifies the column from which the name of user will be fetched. Gets persisted in Name column of SKVirtualActor table. **email**: This setting identifies the column from which the email of user will be fetched. Gets persisted in Email column of SKVirtualActor table. **groupid**: This setting identifies the column from which the groupid of user will be fetched. Does not get persisted in SKVirtualActor table. **groupname**: This setting identifies the column from which the groupname of user will be fetched. Does not get persisted in SKVirtualActor table. **manager**: This setting identifies the column from which the name of manger will be fetched. Does not get persisted in SKVirtualActor table. **password**: Used to set the column which contains password value while using forms provider. Can be used only with forms provider.
persist	Boolean flag which can be set as true/false to decide if the column value needs to be persisted in SKVirtualActor table. The default entries in SKVirtualActor table are for purpose identifier, name and email.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

persistcolumn	SKVirtualActor table contains generic columns like nvarchar1, datetime1 etc which can be used to persist custom information in the table. This attribute can be used to set the column to which the value needs to be persisted for a user. Only the generic columns should be used to persist custom info.
entitymappedproperty	Attribute used to map the user entity data source provider property to an entity property. An example will be provided below on how to map a property in user entity data source provider to entity provider.
displayname	Does not have any specific functionality and can be ignored. Was maintained for back ward compatibility in XML Schema and will be removed in future release.
suppressdn	This attribute is used only with active directory. Based on this setting the value fully qualified name returned by AD is trimmed to return exact required value. Normally used with Role and Manager property in Active Directory provider.

Note: The property field set as purpose=" identifier" should be passed as the real user id prefixed with the provider instance name & "::". For the above example the real user id for Michelle will be "<Provider Instance Name>:: e9d21049-e3cf-4f4d-9d51-036d0145c547". A sample is provided after the description section for better clarity.

The entity definition XML in SKEntity table will be as below:

```
<entity type="user">
<properties>
<property name="UserId" displayname="Id" purpose="identifier"  type="string" />
<property name="UserName"  displayname="Display Name" purpose="name"  type="string"/>
<property name="UserEmail" displayname="Email"  purpose="email" type="string"/>
<property name="Role" displayname="Designation" type="string"/>
</properties>
</entity>%%$#@@!!!*&
```

This property node contains information related to the user entity data source provider. The properties defined in entity should match the columns defined for the user entity data source provider. In the above example 2 properties (UserName, UserEmail) does not exactly match the properties defined in the sample for user entity data source provider, this will be explained after the attribute description.

The special characters (%%$#@@!!!*&) shown at the end of entity definition XML is the separation string to include multiple providers mapping XML along with entity definition XML.

Attribute Name	Description
name	Name of the column used to build filter condition for building query to filter out required users.
type	Type defines the value type, you can see that the type defined in user entity data source provider was "int" but in entity it is set as string. This is because the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	unique identifier at entity level will always have the user entity data source provider instance name prefixed, which makes it a string always.

purpose	Purpose defines the type of information available for a property. Based on the purpose set for a property the resource object provides specific info as below.
	//Gets the resource object based on the real user id.
	Workflow.NET.Resource resourceProperties = new Skelta.HWS.Actor(new ApplicationObject("RepositoryName"), "skeltalist::9F67E912-5E5B-42C5-97A0-53462374A3B3").Resource;
	//Return the property value with purpose set as identifier
	resourceProperties.Properties.Identifier;
	//Return the property value with purpose set as name
	resourceProperties.Properties.Name;
	//Return the property value with purpose set as email
	resourceProperties.Properties.Email;
	//Return the property value with purpose set as groupid
	resourceProperties.Properties.GroupID;
	//Return the property value with purpose set as groupname
	resourceProperties.Properties.GroupName;
	//Return the property value with purpose set as manager
	resourceProperties.Properties.Manager;
	
	In the example you can see the real user id getting prefixed to the instance. In the sample the user entity data source provider is AVEVA Work Tasks List and the instance name is set as "skeltalist". The unique identifier type is guid for users in AVEVA Work Tasks List.

displayname	The display name of the property which will be displayed to users while building filter condition to query users. An example of filter UI is the "To" property available for Human Activities.

In the above definition, XML 2 properties (UserName, UserEmail) does not match with the user entity data source provider columns listed. Since the names do not match the attribute "entitymappedproperty" for user entity data source provider can be used to do the mapping. This property gets used when configuring multiple providers for the same repository. The definition XML will be as below with "entitymappedproperty":

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

User Entity Data Source Provider definition XML:

```
<dataproviders>
<provider name="authenticationuserprovider" type="Skelta List">
<properties>
<property  name="UserId" purpose="identifier" type="guid" />
<property  name="Name"  purpose="name" type="string"  entitymappedproperty ="UserName" />
<property  name="Email"  purpose="email" type="string"  entitymappedproperty = "UserEmail"
/>
<property  name="Role" type="string"/>
</properties>
</provider>
</dataproviders>
```

Entity Provider definition XML:

```
<entity type="user">
<properties>
<property name="UserId" displayname="Id"  purpose="identifier"  type="string" />
<property name="UserName"  displayname="Display Name" purpose="name"  type="string"/>
<property name="UserEmail" displayname="Email"  purpose="email" type="string"/>
<property name="Role" displayname="Designation" type="string"/>
</properties>
</entity>%%$#@@!!!*&
```

Find below definition XML for a few default User Entity Data Source Providers

Active Directory

```
<dataproviders>
<provider name="activedirectory" type="activedirectory">
<properties>
<property name="name" type="string"  entitymappedproperty="UserName"/>
<property name="sAMAccountName" persistcolumn="UserIDString" type="string"
purpose="identifier" entitymappedproperty="UserId"/>
<property name="displayName" type="string" purpose="name" persistcolumn="Name"
entitymappedproperty="UserDisplayName"/>
<property name="mail" type="string" purpose="email" entitymappedproperty="UserEmail"/>
<property name="title" type="string"  entitymappedproperty="UserDesignation"/>
<property name="department" type="string"  entitymappedproperty="UserDepartment"/>
<property name="manager" type="string"  suppressdn="false" purpose="manager"
entitymappedproperty="UserManager"/>
<property name="memberof" type="string" purpose="groupname"  suppressdn="true"
entitymappedproperty="UserRole"/>
<property name="distinguishedname" type="string"
entitymappedproperty="UserDistinguishedName"/>
</properties>
<Preferences>
<SearchScope>SubTree</SearchScope>
<ReferralChasing>External</ReferralChasing>
<TimeOut>60</TimeOut>
<SizeLimit>0</SizeLimit>
<AuthenticationUser></AuthenticationUser>
<AuthenticationPassword ></AuthenticationPassword>
</Preferences>
<LDAPServer>LDAP://</LDAPServer>
<LDAPSearchBase>DC=SK,DC=Skelta,DC=net</LDAPSearchBase>
<LDAPBaseFilter>(objectCategory=Person)</LDAPBaseFilter>
<DomainName>NG\</DomainName>
</provider>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</dataproviders>
AVEVA Work Tasks List
<dataproviders>
<provider name="authenticationuserprovider" type="Skelta List">
<properties>
<property name="Title"  type="string" purpose="name" entitymappedproperty="UserDisplayName,
UserName"/>
<property name="Id" type="guid" purpose="identifier" entitymappedproperty="UserId"/>
<property name="Role" type="string" entitymappedproperty="UserRole"/>
<property name="Password" type="string" purpose="password"/>
<property name="Description"  type="string" />
<property name="Email"  type="string" purpose="email" entitymappedproperty="UserEmail"/>
<property name="Department" type="string" entitymappedproperty="UserDepartment"/>
<property name="Manager"  type="string" entitymappedproperty="UserManager"/>
</properties>
</provider>
</dataproviders>
Forms
<dataproviders>
<provider name="sqlprovider" type="sql server">
<properties>
<property name="UserId"  purpose="identifier"  type="string" />
<property name="Name"  purpose="name"  type="string"/>
<property name="Email" purpose="email" type="string"/>
<property name="Role"  type="string"/>
<property name="Password"  purpose="password" type="string"/>
</properties>
<datasource>server=localhost;database=SkeltaUsers;User Id=sa;pwd=XXX;</datasource>
<SourceTable>SqlUsers</SourceTable>
<loginproperty>Name</loginproperty>
</provider>
</dataproviders>
```

Configure Forms Provider for Authentication

You can create a Repository with Forms Provider for Authentication. Repository with Forms authentication implies that the user details are fetched from a user defined table. These details can be used for various purposes like login authentication into the Enterprise Console, assigning actors to the human activities and others.

Following are the steps that are required to create a repository with forms authentication:

1. Prerequisite - Create a table that contains the user details. The table could be in the same database that is mapped to the repository or in a different database. For example:

   ```
   Create table Users(UserIdentifier int,FirstName varchar(50),PasswordValue varchar(50),
   LastName varchar(50), Email varchar(50),ManagerId int)
   ```

2. In Central Configuration, click the link Repository Settings that is under the title Repository. Now the repository list page is displayed.
3. Click the link named "Create Repository" that is just below the repository list. Manage Repository page is displayed.
4. The Manage Repository page contains fields on the right side of it that needs to be filled. The mandatory fields that need to be configured are:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

a. Repository Data Source – Select data source listed in this drop-down. If selected data source is not listed in the drop-down then click the link "Create data source" below the drop-down to create a new data source.

b. Attach Data Source – To attach the data source, that is being used by any other repositories then check this option. Then a drop-down with all the repositories will be listed and user can select one from them.

c. Repository Name – Provide a repository name in this field.

d. Description - Provide a description for repository in this field. [optional]

e. Servers – Select server from the list box provided.

f. Resource Provider – In this drop-down, select the provider as Forms.

g. User entity Properties – Enter the XML code that identifies user properties.

The XML code consists of the main tag<properties> followed by the sub tags <property > and its attributes. The property tags are case sensitive. The property attributes should match the columns that are present in the user table newly created.

For example, the columns of the table created by executing the above sample query consists of the columns matching the property attributes in the below sample code.

Sample code

```
<properties>
<property name="UserIdentifier" type="int" purpose="identifier"
displayname="Useridentifier" />
  <property name="FirstName" type="string" purpose="name" displayname="First Name" />
  <property name="PasswordValue" type="string" purpose="password"
displayname="PasswordValue" />
  <property name="LastName" type="string" displayname="Last Name" />
  <property name="Email" type="string" displayname="Email ID" purpose="email" />
  <property name="ManagerId" type="string" displayname="ManagerId" purpose="manager"
/>
</properties>
```

5. User Provider Properties – Enter the XML code to map user properties in the provider to the user Entity Properties.

Sample code

```
<properties>
<property name="UserIdentifier" type="int" purpose="identifier"
displayname="Useridentifier" />
  <property name="FirstName" type="string" purpose="name" displayname="First Name" />
  <property name="PasswordValue" type="string" purpose="password"
displayname="PasswordValue" />
  <property name="LastName" type="string" displayname="Last Name" />
  <property name="Email" type="string" displayname="Email ID" purpose="email" />
  <property name="ManagerId" type="string" displayname="ManagerId" purpose="manager" />
</properties>
```

Data Source – Provide the connection string to the data source.

Data Table/View – In this field, specify the table that provides the user details. For example, created a table by name Users, specify the same name here. i.e Users. In case the user details has to be fetched from than one table then user can specify view in this field.

User login Property – Specify the column name of the field which has to be authenticated for login. For example: UserIdentifier

Click Test Sql Connection to check the status of the connection. If the message displayed is "Test Connection

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Succeeded" then click OK button that is at the bottom of the page.

Map Active Directory and SQL Provider

Overview of mapping between Active Directory and SQL Provider

Mapping is required, if same user exists in more than one provider to avoid duplicate entries in database.

Advantage of Mapping

1. While mapping, if user exists in more than one provider then user can login or act on the given task from any of the providers.
2. Duplication of same user from different providers can be avoided.

Scenario: Log on to the EC site using domain login and use the resource properties defined in the sqlprovider (form) in workflow design.

To set the Mapping

Identify the column which is unique for all the providers i.e. user has to identify the common column which shares the same information for the given user across the providers. In below scenario, considered email as the common column

Active directory provider properties

```
<dataproviders>
<provider name="activedirectory" type="activedirectory">
<properties>
<property name="sAMAccountName"  type="string" purpose="identifier" displayname="Account
Name" entitymappedproperty="AD userid" />
<property name="displayName" type="string" purpose="name" displayname="Resource Name"
entitymappedproperty="name" />
<property name="mail" type="string" purpose="email" displayname="Email"
entitymappedproperty="email"/>
</properties>
<Preferences>
<SearchScope>SubTree</SearchScope>
<ReferralChasing>External </ReferralChasing> <TimeOut>60</TimeOut>
<SizeLimit>0</SizeLimit>
<AuthenticationUser>skelta\kalaiyar</AuthenticationUser>
<AuthenticationPassword IsEncrypted="true"> MqXasBHPAOsX6NU+oNwPSQ==
</AuthenticationPassword>
</Preferences><LDAPServer>LDAP://</LDAPServer>
<LDAPSearchBase>DC=skelta,DC=dom </LDAPSearchBase>
<LDAPBaseFilter>(objectCategory=Person)</LDAPBaseFilter>
<DomainName>SKELTA\</DomainName>
</provider>
</dataproviders>
```

sqlprovider Properties

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
dataproviders>
<provider name="sqlprovider" type="sql server" authenticationsource="false"> <properties>
<property name="SQLuserid" type="int" purpose="identifier" />
<property name="name" type="string"  purpose="name" display="true"/>
<property name="email" type="string" purpose="email"/>
</properties>
<datasource>server=SKELTADTP112;database=TestMappingAd-SQL;Integrated
security=true</datasource>
<SourceTable>users1</SourceTable>
</provider>
</dataproviders>
```

To Map between Active directory and SQL Provider

The **'mail'** property from active directory and email property from DB has the common field to map i.e. they both have the same information for the given user mail is active directory property. email is a column in sqlprovider.

Default Instance name for ActiveDirectory is activedirectory

Default Instance name for sql is sqlprovider

```
<<mappingfields>
 <mappingfield mappedfromprovider="ActivedirectoryInstance name">
 <fields mappedtoprovider= "sql provider instance name" ><field name="mail"
mappedto="email" />
 </fields>
 </mappingfield>
 <mappingfield mappedfromprovider="sql provider instance name">
 <fields mappedtoprovider ="ActivedirectoryInstance name" ><field name="email"
mappedto="mail"/>
 </fields>
 </mappingfield>
</mappingfields>
```

Note: User can map between any number of providers and on any common columns. Provider instance name are taken from SKEntityDatasourceprovider table.

To map Active Directory Provider with SQL Provider

1. Create repository with active directory as resource provider.
2. Insert a record to SKEntityDataSourceProvider with Application set to Repository name, EntityID taken from the SKEntity table (Userentity for mapping users), Provider to sqlprovider and Instancename with any UserdefinedName.
3. Update the DefnitionXML column for the added record with appropriate resource XML data.

   Example:

   ```
   <dataproviders>
   <provider name="sqlprovider" type="sql server" authenticationsource="false">
   <properties>
   <property name="SQLuserid" type="int" purpose="identifier" />
   <property name="name" type="string"  purpose="name" display="true"/>
   <property name="email" type="string"  purpose="email" persistcolumn="Email"/>
   </properties>
   <datasource>server=SKELTADTP112;database=TestMappingAd-SQL;Integrated security=true
   </datasource>
   SourceTable>users</SourceTable>
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</provider>
</dataproviders>
```

Note: Add persistcolumn attribute for the common column in both the providers. Authenticationsource="false" at the provider level disable the login to the repository from EC site for this provider and Connection string should be changed.

4. Update DefinitionXML column of the active directory by adding persistcolumn attribute for the common property.

   Example:

```
<dataproviders>
<provider name="activedirectory" type="activedirectory">
<properties>
<property name="sAMAccountName"  type="string" purpose="identifier"
displayname="Account Name" entitymappedproperty="AD userid" />
<property name="displayName" type="string" purpose="name" displayname="Resource Name"
entitymappedproperty="name" />
<property name="mail" type="string" purpose="email"
displayname="Email" persistcolumn="Email" entitymappedproperty="email"/>
</properties>
<Preferences>
<SearchScope>SubTree</SearchScope>
<ReferralChasing>External </ReferralChasing> <TimeOut>60</TimeOut>
<SizeLimit>0</SizeLimit>
<AuthenticationUser>skelta\kalaiyar</AuthenticationUser>
<AuthenticationPassword IsEncrypted="true"> MqXasBHPAOsX6NU+oNwPSQ==
</AuthenticationPassword>
</Preferences>
<LDAPServer>LDAP://</LDAPServer>
<LDAPSearchBase>DC=skelta,DC=dom </LDAPSearchBase>
<LDAPBaseFilter>(objectCategory=Person)</LDAPBaseFilter>
<DomainName>SKELTA\</DomainName>
</provider>
</dataproviders>
```

Note: Active directory preferences should be changed.

5. Update DefinitionXml of SKEntity table for the user entity.

   Example:

```
entity type="user">
<properties>
<property name="AD userid" type="string" purpose="identifier"/>
<property name="SQLuserid" type="int" purpose="identifier"/>
<property name="name" type="string" displayname="Name" purpose="name"
display="true"/>
<property name="email" type="string" displayname= "Email ID" purpose ="email" />
</properties>
</entity>%%$#@@!!!*&
<mappingfields>
 <mappingfield mappedfromprovider="activedirectory">
<fields mappedtoprovider= "AD_SQLprovider" >
 <field name="mail" mappedto="email" />
</fields>
 </mappingfield>
 <mappingfield mappedfromprovider="AD_SQLprovider">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<fields mappedtoprovider ="activedirectory" >
 <field name="email" mappedto="mail"/>
</fields>
 </mappingfield>
</mappingfields>
```

Note: The properties configured in DefinitionXml of SKEntity table are available to create the query for assigning the activities to the resources. mappedfromprovider and mappedtoprovider are the instance name taken from the SKEntityDataSourceProvider.

6. Create the table named "users". This is the SourceTable for sqlprovider. The sample table is given below.

```
CREATE TABLE [dbo].[users](
 [SQLuserid] [int] NULL,
 [name] [varchar](50) NULL,
 [email] [varchar](50) NULL
 ) ON [PRIMARY]
GO
```

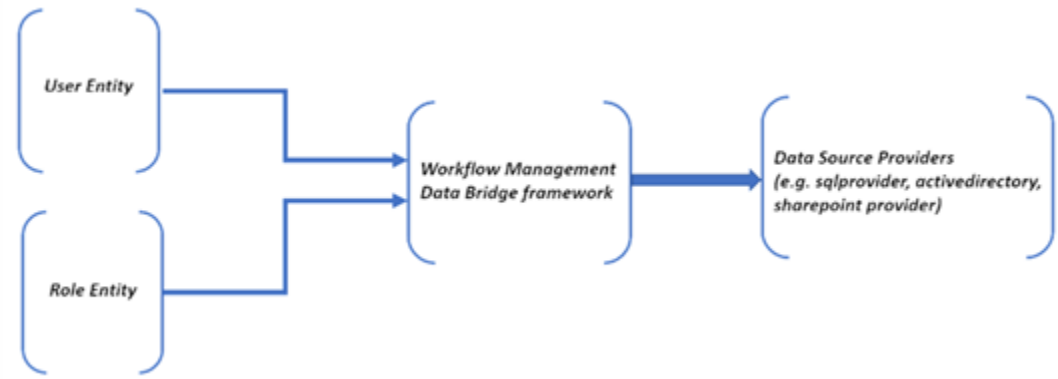
7. When the Activities are assigned to resources, it will look for the resource in both the providers and if common field found for the same, it will assign the activity to the user based on the mapped id.

When the user with common field found and if that user's details are already created in SKVirtualActor table, AVEVA Work Tasks automatically maps that user using MappedToId.

Role Entity

Role entity is used to fetch the role information for the data source provider. Role Entity provides information of roles and the associated users in the role. This will again communicate with the data bridge frame work that will get the users in Roles based on the role entity data source provider.

The mapping table used for storing the role information will be **SKVirtualRole** table. The table contains data related to role.



XML Schema Used by Role Entity Data Source Provider

A Role entity data source provider cannot exist on its own and needs to be associated always with a User entity data source provider. The reason being, the users who belong to the role have to be mapped to a user entity data source provider to be used in the system. The mapping between the role entity data source provider and user entity data source provider is achieved by maintaining the same instance name for the providers in the **SKEntityDataSourceProvider** table.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Now let us take a look at the Definition XML schema for the Role entity.

Role Entity Data Source Provider definition XML in SKEntityDataSourceProvider table will be as shown below.

```
<dataproviders>
<provider name="customsqlprovider" type="Custom Role Provider">
<properties>
<property name="Role" type="string" purpose="identifier"/>
<property name="RoleName" type="string" persist="true" persistcolumn="rolename"/>
</properties>
<datasource>
<datasource>server=localhost;database=SkeltaUsers;User Id=sa;pwd=XXX;</datasource>
<SourceTable>SqlUsers</SourceTable>
</datasource>
</provider>
</dataproviders>
```

The properties defined above are not used in any of the UI. To comply with the entity data source provider infrastructure this has been provided. One important thing to be taken care for Role entity data source provider is to ensure that the identifier value and Role Name getting persisted in SKVirtualRole database. If this is taken care by the code then there is no need for this definition XML. However the database connection string info can be useful. The type for the identifier property should be string and should always contain the Role name itself as the value. This is mandatory as the Role name itself is considered as unique identifier for the role provider. A sample for role provider value would be sqlproviderCustom::Admin (<Provider Instance Name> + "::"+ RoleName) here admin is the unique identifier value which is the role name.

Entity definition XML in SKEntity table will be as below.

<entity type="role"></entity>%%$#@@!!!*&

There are no properties defined at entity level as there is no UI, where the Role properties get listed. As the name implies the primary property provided by the Role provider is the Role name.

Synchronizing Roles of Queue Participants

When a user role is first made a participant in a queue, AVEVA Work Tasks internally stores all users with that role for allocation of work items assigned to the queue. As the resource database changes, with new users added to the role and changes in roles, the repository of users for the role need to be synchronized with the resource database. This synchronization is done based on an entry in the table SKROLEREFRESH for the resource handler.

The columns in this table are:

- **ID** - a system-generated field.
- **Resource handler** - the name of the resource handler for which the refresh setting is specified.
- **RoleRefreshInterval** - the interval specified in dd.hh:mm:ss at which the synchronization must be done.
- **IsActive** - This field must be initially set to False; This is later updated to True by the system when a Queue is created or edited.
- **TaskId** - the corresponding ID of the refresh task in the task scheduler. This field is updated by the system.
- **LastUpdatedDateTime** - The time stamp of the last update. This field is updated by the system.

When roles are refreshed, and there is a change in the users of a role that is a queue participant, all open items of users who are not longer a part of the role which the user has not taken ownership of are automatically moved back to the queue. Items that the user has taken ownership continue to stay in his work list. However, these items will no longer be considered a part of the queue. For example, a queue for processing Invoice

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Payments has the role Process Associate as a participant and John Smith as a Process Associate has 5 open items from this queue. John Smith is promoted to Manager and the role change is effected in the resource database. When the synchronization happens, John has 5 open items 3 of which has already started to process. The 2 open items that has not taken ownership of are moved back to the queue for reallocation while the 3 items that has started to process are not complete continue to stay in his work list.

Deleting Synchronization Settings

To temporarily discontinue refresh or to delete the setting, delete the entry in the table.

Entity Data Source Provider Code Samples

Introduction

In this section, you will learn:

1. How to create User and Role Entity Data source provider?
2. How to create Custom Role Provider Fetching Roles from SQL and Users from Active directory?

Create User Entity Data Source Provider

Using this API a custom User Entity Data Source Provider can be created.

Note: The Repository for which you are going to create the User Entity Data Source Provider, should have Authentication provider as "AVEVA Work Tasks List".

- **Create a table named SqlUsers**

The custom entity provider fetches the user information from a table called **SqlUsers**.

Script to create this table is provided below:

```
GO
/****** Object:  Table [dbo].[SqlUsers]  ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SqlUsers](
[UserId] [uniqueidentifier] ROWGUIDCOL  NOT NULL CONSTRAINT [DF_SqlUsers_Id]  DEFAULT
(newid()),
 [Name] [nvarchar](100) NULL,
 [Role] [nvarchar](200) NULL,
 [Email] [nvarchar](75) NULL,
 [Password] [nchar](10) NULL,
 CONSTRAINT [PK_SqlUsers_1] PRIMARY KEY CLUSTERED
(
[UserId] ASC
) WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

- **Insert default users to SqlUsers Table**

Script to insert default users to SqlUsers Table:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
INSERT INTO [dbo].[SqlUsers]
 ([Name]
 ,[Role]
 ,[Email]
 ,[Password])
 VALUES
 ('Michelle','Admin','michelle@skelta.com','michelle'),
 ('kevin','Developer','kevin@skelta.com','kevin'),
 ('john','Developer','john@skelta.com','john'),
 ('Elaine','Product Manager','elaine@skelta.com','elaine')
GO
```

- **Create a dll named CustomUserEntityDataSourceProvider**

PROCEDURE

DLL REFERENCE

```
Workflow.NET.NET2
System.Transactions.dll
System.Web.dll
```

NAMESPACE USED

```
System
System.Collections.Generic
System.Text
System.Xml
System.Xml.XPath
System.Transactions
System.Data
Workflow.NET
Workflow.NET.Interfaces
Skelta.Entity
```

API INFORMATION

```
The User Entity Data Source Provider should
Inherit Skelta.Entity.DataSourceProviderBase class
```

or

```
Implement Skelta.Entity.IEntityDataSourceProvider, Skelta.Core.ISkeltaAddInProvider
Implement interface Skelta.Entity.UserEntityProvider.IUserEntityDataSourceProvider
Implement interface Skelta.Entity.IEntityAdminDataSourceProvide.
```

Skelta.Entity.DataSourceProviderBase: Base class used by the default providers, this class implements the interfaces Skelta.Entity.IEntityDataSourceProvider, Skelta.Core.ISkeltaAddInProvider. All the methods in this class are declared as virtual. The implementers should ideally implement the interfaces implemented by the base class directly. The base class is used in the sample trying to make it easier for the implementer with some of the methods already implemented with default functionality. The method which needs to be mandatorily overridden throws exception if relied on the base class. This class exposes methods for initializing the provider, fetching user information & querying users based on filters.

Skelta.Entity.UserEntityProvider.IUserEntityDataSourceProvider: Gets called during EnterpriseConsole authentication from the Log on page. This interface provides methods which gives username & password entered in the login page of EnterpriseConsole. All the user authentication code must be implemented in these methods.

Skelta.Entity.IEntityAdminDataSourceProvider: Required for setting the default administrator for a Repository.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

This administrator will have complete rights on the EC site. This administrator will be responsible for setting security and adding users to EC.

SAMPLE CODE:

```
namespace CustomUserEntityDataSourceProvider
{
/// <summary>
/// Custom entity data source  provider which queries sql database table/view returning
user information.
/// </summary>
public class UserEntityDataSourceProvider : DataSourceProviderBase,
Skelta.Entity.UserEntityProvider.IUserEntityDataSourceProvider,
Skelta.Entity.IEntityAdminDataSourceProvider
{
private string sConnectionString;
private string sDataSourceType;
private string sSourceTable;
private string sCols = "";
private Log log = null;
private string loginProperty = "";
```

```
public UserEntityDataSourceProvider() { }

/// <summary>
/// Gets called  while provider is initialized.
/// </summary>
/// <param name="settings">Definition XML configured for SKEntityDataSourceProvider
table.</param>
public override void Initialize(string settings)
{
log = new Log();
log.FileName = "CustomUserEntityDataSourceProvider.cs";
XmlDocument doc = new XmlDocument();
try
{
doc.LoadXml(settings);
// Get the Connection String from the definition xml file.
XmlNode DsConnNode = doc.GetElementsByTagName("datasource").Item(0);
sConnectionString = DsConnNode.InnerText;
sDataSourceType = "sql server";
sSourceTable = doc.GetElementsByTagName("SourceTable").Item(0).InnerText;
}
catch (Exception exp)
{
log.LogError(exp, "Could not read the xml data while initializing the provider");
throw;
}
try
{
if (doc.GetElementsByTagName("loginproperty") != null)
{
XmlNode qxn = doc.GetElementsByTagName("loginproperty").Item(0);
loginProperty = qxn.InnerText;
}
}
catch (Exception ex)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
log.LogError(ex, "Could not get loginproperty column set.");
throw;
}
}
```

```
/// <summary>
/// Called when delayed initialization on a resource property is required.
/// For example if there are 2 providers and the initial property collection for the
resource
/// has a property which is empty and if this property is requested a delayed
initialization is
/// triggered on the  provider based on the mapping  fields configured.
/// </summary>
/// <param name="filter">Filter which is internally built based on the mapping fields
configured.</param>
/// <param name="dataparams">Parameters used in the filter.</param>
/// <returns>Entity item with the delayed initialization property  value.</returns>
public override EntityItem GetEntityItem(string filter, object[] dataParams)
{
if(string.IsNullOrEmpty(sCols))
BuildPropertyColumn();
List<IDataParameter> dParam = new List<IDataParameter>();
string sql = string.Empty;
using (IDataHandler dbHandler =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString, sDataSourceType))
{
sql = "SELECT " + sCols + " FROM " + sSourceTable + " WHERE " + filter;
try
{
foreach (object[] values in dataParams)
{
dParam.Add(dbHandler.GetParameter(values[0].ToString(), values[1]));
}
return GetEntityItem(sql, dParam.ToArray());
}
catch (Exception exp)
{
log.LogError(exp, "Could not fetch entity item with filter (" + filter + ").");
throw;
}
}
}

/// <summary>
/// Called when actor object is created or any resource details are requested.
/// </summary>
/// <param name="entityItemId">Guid of the resource based on the current sample.</param>
/// <returns>User Entity Details.</returns>
public override EntityItem GetEntityItem(object entityItemId)
{
if (string.IsNullOrEmpty(sCols))
BuildPropertyColumn();
string sql = string.Empty;
IDataParameter dParam = null;
using (IDataHandler dbHandler =
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString, sDataSourceType))
{
try
{
sql = "SELECT " + sCols + " FROM " + sSourceTable + " WHERE " +
EntityItemProperties.Identifier.PropertyName + " = @resourceId";
dParam = dbHandler.GetParameter("@resourceId", entityItemId);
return GetEntityItem(sql, dParam);
}
catch (Exception exp)
{
log.LogError(exp, "Could not fetch entity item for entity item id (" + entityItemId +
").");
throw;
}
}

}

EntityItem GetEntityItem(string filter, params IDataParameter[] dataparams)
{
IDataReader idr = null;
EntityItem res;
Workflow.NET.Interfaces.IDataHandler ConnectionWrapper = null;
using (ConnectionWrapper =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString, sDataSourceType))
{
idr = ConnectionWrapper.ExecuteReader(filter, dataparams);
//Create a empty EntityItem object
res = new EntityItem();
res.Properties = EntityItemProperties.CreateDeepClone();
string fieldname;
EntityItemProperty rp;
int loopcount = 0;
while (idr.Read())
{
//For every record, update the properties values
for (int i = 0; i < idr.FieldCount; i++)
{
fieldname = idr.GetName(i);
if (res.Properties.ContainsKey(fieldname))
{
rp = ((EntityItemProperty)res.Properties[fieldname]);
if (!rp.Values.Contains(idr.GetValue(i)) && idr.GetValue(i) != System.DBNull.Value &&
idr.GetValue(i).ToString() != "") //Add all unique values of a given field in Values
collection
rp.Values.Add(idr.GetValue(i));
}
else
{  //if( loopcount==0 )
log.LogWarning("Field " + fieldname + " not found in EntityItem Properties");
}
}
loopcount++;
}
idr.Close();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
idr.Dispose();
}
return res;
}
```

```
/// <summary>
/// Called  when  an filter is executed in  Runtime, for example when Human activity is
executed.
/// This method  also gets called when user look search is performed
/// </summary>
/// <param name="filter">Filter for querying the resource.</param>
/// <param name="query"></param>
/// <param name="FullyProcessedQuery"></param>
/// <param name="Params"></param>
/// <returns></returns>
public override object[] QueryEntityItem(string filter, string query, string
FullyProcessedQuery, object[] Params)
{
// When QueryEntityItem is called from Runtime (Engine),
//there is an option to get the executing context details using GetEntityExecutionContext
method.
Skelta.Entity.EntityExecutionContextInfo entityExecutionContextInfo =
Skelta.Entity.EntityExecutionContext.GetEntityExecutionContext();
//if (entityExecutionContextInfo != null)
//{
//entityExecutionContextInfo.ApplicationName;
//entityExecutionContextInfo.WorkflowName;
//entityExecutionContextInfo.ExecutionId
//}
IDataReader idr = null;
Workflow.NET.Interfaces.IDataHandler ConnectionWrapper = null;
try
{
using (ConnectionWrapper =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString, sDataSourceType))
{
string sql = "SELECT DISTINCT " + EntityItemProperties.Identifier.PropertyName + " FROM " +
sSourceTable;
System.Collections.ArrayList EntityItems = new System.Collections.ArrayList();
if (FullyProcessedQuery != null && FullyProcessedQuery.Trim().Length > 0)
{
sql += " WHERE " + FullyProcessedQuery;
}
else
{
log.LogWarning("EntityItem filter is emtpy.");
return EntityItems.ToArray();
}
```

```
System.Collections.ArrayList alParams = new System.Collections.ArrayList();
foreach (object Param in Params)
{
object[] obj = (object[])Param;
alParams.Add(ConnectionWrapper.GetParameter((string)obj[0], obj[1]));
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
IDataParameter[] Parameters = (IDataParameter[])alParams.ToArray(typeof(IDataParameter));
idr = ConnectionWrapper.ExecuteReader(sql, Parameters);

while (idr.Read())
{
EntityItems.Add(idr.GetValue(0));
}
idr.Close();
idr.Dispose();
return EntityItems.ToArray();

}
}
catch (Exception exp)
{
if (entityExecutionContextInfo != null)
log.LogError(exp, "Could not fetch entity item while querying entity item,  filter(" +
filter + ").", entityExecutionContextInfo.ApplicationName,
entityExecutionContextInfo.WorkflowName, entityExecutionContextInfo.ExecutionId);
else
log.LogError(exp, "Could not fetch entity item while querying entity item,  filter(" +
filter + ").");
throw;
}
}
void BuildPropertyColumn()
{
if (string.IsNullOrEmpty(sCols))
{
foreach (EntityItemProperty prop in EntityItemProperties)
{
sCols += (sCols == "" ? "" : ",") + prop.PropertyName;
}
}
}

/// <summary>
/// Creates a clone  of the user entity provider.
/// </summary>
/// <returns></returns>
public override IEntityDataSourceProvider Clone()
{
UserEntityDataSourceProvider customProvider = new UserEntityDataSourceProvider();
customProvider.sCols = this.sCols;
customProvider.Name = this.Name;
customProvider.sConnectionString = this.sConnectionString;
customProvider.sDataSourceType = this.sDataSourceType;
customProvider.sSourceTable = this.sSourceTable;
EntityItemPropertiesCollection entityItems = this.EntityItemProperties.CreateDeepClone();
foreach (EntityItemProperty eProp in entityItems)
{
customProvider.EntityItemProperties.Add(eProp.PropertyName, eProp);
}
foreach (string key in this.DataSourceMappingFields.Keys)
{
customProvider.DataSourceMappingFields.Add(key, this.DataSourceMappingFields[key]);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
foreach (string key in this.DataSourceProperties.Keys)
{
customProvider.DataSourceProperties.Add(key, this.DataSourceProperties[key]);
}
foreach (string key in this.RuleParts.Keys)
{
customProvider.RuleParts.Add(key, this.RuleParts[key]);
}
customProvider.IsPrimaryDataSource = this.IsPrimaryDataSource;
customProvider.DataSourceType = this.DataSourceType;
customProvider.loginProperty = this.loginProperty;
customProvider.Name = this.Name;
customProvider.CurrentEntity = this.CurrentEntity;
customProvider.DisplayName = this.DisplayName;
customProvider.RepositoryName = this.RepositoryName;
customProvider.IsAuthenticationSource = this.IsAuthenticationSource;
customProvider.ProviderPrefix = this.ProviderPrefix;
return customProvider;
}
/// <summary>
/// Converts to provider specific identifier type;
/// </summary>
/// <param name="identifierId">real entity id converted as string.</param>
/// <returns>Id which  is provider specific type converted.</returns>
public override object ConvertToIdentifierType(string identifierId)
{
switch (EntityItemProperties.Identifier.PropertyType)
{
case "string": return identifierId;
case "int": return int.Parse(identifierId);
case "guid": return new Guid(identifierId);
}
return null;
}

/// <summary>
/// Return the unique identifier for the user. The default return value is user name
itself.
/// The actual unique identifier for the user could be different from the property used for
login. For  the sample though the unique identifier is guid we have provided email as the
login property for ease of use.
/// </summary>
/// <param name="userName">Value entered in the login page against user name field.</param>
/// <param name="password">Password entered in the login page against password
field.</param>
/// <returns>Unique identifier  value. Return value type is object as the identifier type
could be guid, int or string.</returns>
public override object GetUniqueIdentifier(string userName, string password)
{
if (string.IsNullOrEmpty(loginProperty))
return userName;
string passwordColumn = GetPasswordColumnName();
Workflow.NET.Interfaces.IDataHandler ConnectionWrapper = null;
string sql = "";
IDataParameter param1 = null;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
IDataParameter param2 = null;
IDataReader idr = null;
try
{
ConnectionWrapper =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString, sDataSourceType);
sql = "SELECT * FROM " + sSourceTable + " WHERE " + loginProperty + "=@resourceId AND " +
passwordColumn + "=@password";
param1 = ConnectionWrapper.GetParameter("@resourceId", userName);
param2 = ConnectionWrapper.GetParameter("@password", password);
idr = ConnectionWrapper.ExecuteReader(sql, param1, param2);
if (idr.Read())
return idr[EntityItemProperties.Identifier.PropertyName];
else
throw new System.Exception("Unable to get unique identifier from property (" +
loginProperty + "), query(" + sql + ")");

}
catch (Exception e)
{
log.LogError(e, "Could not get unique identifier for user (" + userName + ")");
throw;
}
finally
{
if (idr != null)
idr.Close();

if (ConnectionWrapper != null)
ConnectionWrapper.Dispose();
}
}
//Returns the column name from property collection with purpose set as "password"
string GetPasswordColumnName()
{
foreach (EntityItemProperty eProp in EntityItemProperties)
{
if (!string.IsNullOrEmpty(eProp.Purpose) &&
eProp.Purpose.ToLowerInvariant() == "password")
return eProp.PropertyName;

}
return "";
}
#region IUserEntityDataSourceProvider Members

/// <summary>
/// Currently this method call is not used in any  of the  default providers.
/// It was added to provide an option in future release to perform login  for a provider
/// based on different authentication methods available in IIS instead of performing a
validation
/// based on user name & password
/// </summary>
/// <param name="httpContext"></param>
/// <returns></returns>
bool
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Skelta.Entity.UserEntityProvider.IUserEntityDataSourceProvider.ValidateIISUser(System.Web.H
ttpContext httpContext)
{
return false;
}
/// <summary>
/// Called when submit button event of login page is fired from EnterpriseConsole.
/// </summary>
/// <param name="userName">Value entered in the login page against user name field.</param>
/// <param name="password">Password entered in the login page against password
field.</param>
/// <returns>true if user credentials are valid else false.</returns>
bool Skelta.Entity.UserEntityProvider.IUserEntityDataSourceProvider.ValidateUser(string
username, string password)
{
string passwordColumn = GetPasswordColumnName();
string identifierColumn = "";
if (string.IsNullOrEmpty(loginProperty))
identifierColumn = EntityItemProperties.Identifier.PropertyName;
else
identifierColumn = loginProperty;

if (string.IsNullOrEmpty(passwordColumn))
return false;
else
{
Workflow.NET.Interfaces.IDataHandler ConnectionWrapper = null;
string sql = "";
IDataParameter param1 = null;
IDataParameter param2 = null;
IDataReader idr = null;
using (ConnectionWrapper =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString, sDataSourceType))
{
try
{
sql = "Select * From " + sSourceTable + " WHERE " + identifierColumn + "=@resourceId AND "
+ passwordColumn + "=@password";
param1 = ConnectionWrapper.GetParameter("@resourceId", username);
param2 = ConnectionWrapper.GetParameter("@password", password);
idr = ConnectionWrapper.ExecuteReader(sql, param1, param2);
bool returnFlag = false;
if (idr.Read())
returnFlag = true;
idr.Close();
return returnFlag;
}
catch (Exception e)
{
log.LogError(e, "Could not validate user (" + username + ")");
//Need not throw the exception as the UI will display error message if return value is
false.
}
}
return false;
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
#endregion
```

```
#region IEntityAdminDataSourceProvider Members
/// <summary>
/// Return admin users unique identifier value.
/// This administrator will have complete rights on the EC site.
/// This administrator will be responsible for setting security and adding users to EC
/// </summary>
/// <returns>Unique identifier value of administrator</returns>
public object[] GetAdministrator()
{
//For the  sample we'll use the column named Role with value set as Admin.
Workflow.NET.Interfaces.IDataHandler ConnectionWrapper = null;
System.Collections.ArrayList EntityItems = new System.Collections.ArrayList();
try
{
ConnectionWrapper =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString, sDataSourceType);
string stradmin = "Admin";
string sql = "SELECT DISTINCT " + EntityItemProperties.Identifier.PropertyName + " FROM " +
sSourceTable;
sql += " WHERE Role = '" + stradmin + "'";
System.Collections.ArrayList alParams = new System.Collections.ArrayList();
IDataParameter[] Parameters = (IDataParameter[])alParams.ToArray(typeof(IDataParameter));
IDataReader idr = ConnectionWrapper.ExecuteReader(sql, Parameters);
while (idr.Read())
{
EntityItems.Add(Workflow.NET.CommonFunctions.GetGuid(idr,
EntityItemProperties.Identifier.PropertyName));
}
idr.Close();
idr.Dispose();
}
catch (Exception e)
{
log.LogError(e, "Could not get the administrator user information.");
throw;
}
finally
{
if (ConnectionWrapper != null)
ConnectionWrapper.Dispose();
}
return EntityItems.ToArray();
}
#endregion
}
}
```

- **Configure Custom User Entity Data Source Provider**

Configuring custom user entity data source provider involves entries in three tables of the Repository

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Table : SKAddInProviders , Object : Skelta.Core.AddInProvider
2. Table : SKEntity, Object : Skelta.Entity.Entity
3. Table : SKEntityDataSourceProvider, Object : Skelta.Entity.DataSourceProvider

- **Add the assembly information of the user entity custom provider created, in the SKAddInProviders table**

Provider type must be **"UserDSProvider"** and name should be any unique name. The newly added provider name must be again updated in the SKEntityDataSourceProvider table.

Using API:

```
Skelta.Core.AddInProvider addInProvider = new Skelta.Core.AddInProvider(
new ApplicationObject("Sample Repository"));
addInProvider.Type = "UserDSProvider";
addInProvider.Name = "customsqlprovider";
#warning : Make sure to change the assembly  path
addInProvider.Description = "Sample Custom User Entity Data Source Provider";
addInProvider.ClassName =
 "CustomUserEntityDataSourceProvider.UserEntityDataSourceProvider";
addInProvider.AssemblyNameWithAbsoluteOrRelativePath =  @"M:\Michelle\
CustomUserEntityDataSourceProvider\
CustomUserEntityDataSourceProvider\bin\Debug\CustomUserEntityDataSourceProvider.dll";
```

addInProvider.IsGacAssembly = false;

addInProvider.IsGlobal = true;

addInProvider.IsDefault = true; // The properties IsGacAssembly, IsGlobal, IsDefault is //not currently in use. The value used in sample should be used "as is"

addInProvider.Save();

Using Script:

```
INSERT INTO SKAddInProviders VALUES (newid(),'UserDSProvider','customsqlprovider',
'Sample Custom User Entity Data Source Provider
 ','CustomUserEntityDataSourceProvider.UserEntityDataSourceProvider',
 'M:\Michelle\CustomUserEntityDataSourceProvider\CustomUserEntityDataSourceProvider\bin\
Debug\CustomUserEntityDataSourceProvider.dll',null,0,1,1,getutcdate())
```

Note: Assembly path should be changed.

Verification:

The above code creates an entry in SKAddInProviders table with type as **'UserDSProvider'**. The entry in Repository DB can be verified in database by running the query select * from SKAddInProviders Where Name='customsqlprovider'

Column Name	Value
ID	Auto Generated
Type	UserDSProvider
Name	customsqlprovider
Description	Sample Custom User Entity Data Source Provider

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Column Name	Value
ClassName	CustomUserEntityDataSourceProvider. UserEntityDataSourceProvider
Assembly	M:\Michelle\ CustomUserEntityDataSourceProvider\ CustomUserEntityDataSourceProvider\ bin\Debug \CustomUserEntityDataSourceProvider.dll
IsGacAssembly	False
Isdefault	True
IsGlobal	True
DateTime	Date time value

- Adding entry to SKEntity and SKEntityDataSourceProvider
  - Update the DefinitionXml column of SKEntity table with appropriate resource XML data for the newly created custom user entity provider that has been updated in the SKAddInProviders table.
  - Update the DefinitionXml column, Provider, Instance Name, Display Name for default entry of the List in SKEntityDataSourceProvider table with appropriate custom user entity data source provider entries.

Custom User Entity Data Source Provider Definition XML

```
<dataproviders>
<provider name="customsqlprovider" type="Custom Sql Provider">
<properties>
<property name="UserId" type="guid"  purpose="identifier" />
<property name="Name" type="string"  purpose="name"/>
<property name="Email"  type="string"  purpose="email"/>
<property name="Password" type="string"  purpose="password"/>
</properties>
<datasource>server=localhost;database=SkeltaUsers;uid=sa;pwd= XXXX</datasource>
<SourceTable>SqlUsers</SourceTable>
<loginproperty>Email</loginproperty>
</provider>
</dataproviders>
```

- "provider" node has attribute name value set as customsqlprovider as this should match the Name column entry in SKAddInProvider.
- As described in earlier sections the purpose attributes identifier, name & email are mandatory.
- Custom purpose named password is used to identify the column containing password info for authentication.
- "datasource" node contains connection string information.
- "SourceTable" node contains the table or view which contains user details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- "loginproperty" node provides information of the column which will be used as user name while logging in to Repository.

Custom Entity Definition XML

```
<entity type="user">
<properties>
<property name="UserId" type="string"  purpose="identifier" />
<property name="Name" type="string"  purpose="name"/>
<property name="Email"  type="string"  purpose="email"/>
</properties>
</entity>%%$#@@!!!*&
```

- Contains properties which are mapped to user entity definition XML properties.
- As described in earlier sections the purpose attributes identifier, name & email are mandatory.

Using API for steps 1 & 2:

```
Skelta.Core.ApplicationObject applicationObject = new Skelta.Core.ApplicationObject("Sample
Repository");
 Skelta.Entity.Entity entity = Skelta.Entity.Entity.GetEntity(applicationObject,
"UserEntity");
 string entityDefinitionXML  = "<entity type=\"user\">";
 entityDefinitionXML += "<properties>";
 entityDefinitionXML += "<property name=\"UserId\" type=\"string\"  purpose=\"identifier\"
/>";
 entityDefinitionXML += "<property name=\"Name\" type=\"string\"  purpose=\"name\"/>";
 entityDefinitionXML += "<property name=\"Email\"  type=\"string\"  purpose=\"email\"/>";
 entityDefinitionXML += "</properties>";
 entityDefinitionXML += "</entity>%%$#@@!!!*&";
 //Update default entity definition xml with matching xml for new  custom user entity data
source provider
entity.DefinitionXml = entityDefinitionXML;
entity.Save();
Skelta.Entity.DataSourceProviderCollection entityDataSourceProvider = null;
entityDataSourceProvider = new
Skelta.Entity.DataSourceProviderCollection(applicationObject,entity.Id);
 //Entity user xml need to be changed based on the custom User table
#warning : Make sure to change the connection string
 string entityDataSourceProviderDefinitionXML = "<dataproviders>";
 entityDataSourceProviderDefinitionXML += "<provider name=\"customsqlprovider\"
type=\"Custom Sql Provider\">";
 entityDataSourceProviderDefinitionXML += "<properties>";
 entityDataSourceProviderDefinitionXML += "<property name=\"UserId\" type=\"guid\"
 purpose=\"identifier\"/>";
 entityDataSourceProviderDefinitionXML += "<property name=\"Name\" type=\"string\"
 purpose=\"name\"/>";
 entityDataSourceProviderDefinitionXML += "<property name=\"Email\"  type=\"string\"
 purpose=\"email\"/>";
 entityDataSourceProviderDefinitionXML += "<property name=\"Password\" type=\"string\"
 purpose=\"password\"/>";
 entityDataSourceProviderDefinitionXML += "</properties>";
entityDataSourceProviderDefinitionXML +=
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"<datasource>server=localhost;database=SkeltaUsers;uid=sa;pwd=XXXX</datasource>";
 entityDataSourceProviderDefinitionXML += "<SourceTable>SqlUsers</SourceTable>";
 entityDataSourceProviderDefinitionXML += "<loginproperty>Email</loginproperty>";
 entityDataSourceProviderDefinitionXML += "</provider>";
 entityDataSourceProviderDefinitionXML += "</dataproviders>";
 entityDataSourceProvider .Items[0].DefinitionXml = entityDataSourceProviderDefinitionXML;
 //Custom provider name which is added in addinprovider table
 entityDataSourceProvider .Items[0].Provider = "customsqlprovider";
 entityDataSourceProvider .Items[0].InstanceName = "sqlproviderCustom";
 entityDataSourceProvider .Items[0].DisplayName = "Custom Provider";
 entityDataSourceProvider .Items[0].Entity = entity;
 // Update skelta list entry  with the custom entry in SKEntityDataSourceProvider for user
provider entityDataSourceProvider.Items[0].Save();
```

The provider name value customsqlprovider should be same as the AddInProvider Name column value.

Using Table Entry for steps 1 & 2:

The definition XML for entity should be matching the properties for entity data source provider.

Update SKEntity Set DefinitionXML = '<entity type="user"><properties><property name="UserId" type="string" purpose="identifier" /><property name="Name" type="string" purpose="name"/><property name="Email" type="string" purpose="email"/></properties></entity>%%$#@@!!!*&%%$#@@!!!*&'

Where Name = 'UserEntity'

Make sure to update the connection string information in definition XML with the appropriate value.
The **provider** column value should be same as the name column value inserted in **SKAddInProviders** table.

Note: Also, ensure that the query given below updates the row for the AVEVA Work Tasks List User Entity Data Source Provider and not the Role Entity Data Source Provider.

```
Update SKEntityDataSourceProvider Set DefinitionXML ='<dataproviders><provider
name="customsqlprovider" type="Custom Sql Provider"><properties><property name="UserId"
 type="guid"  purpose="identifier"/><property name="Name" type="string"
 purpose="name"/><property name="Email"  type="string"  purpose="email"/><property
name="Password" type="string"
 purpose="password"/></properties><datasource>server=localhost;
database=SkeltaUsers;uid=sa;pwd=XXX</datasource><SourceTable>SqlUsers</SourceTable>
<loginproperty>Email</loginproperty></provider></dataproviders>',
DisplayName='Custom Provider', Provider='customsqlprovider',
InstanceName='sqlproviderCustom' Where DisplayName ='Skelta List' and EntityId=(Select Id
from SKEntity where Name='UserEntity')
```

Note: Connection string should be changed.

Verification:

The above code creates an entry in SKEntity table. The entry in Repository DB can be verified in database by running the query select * from SKEntity Where Name= 'UserEntity'and Application='Sample Repository'

Column Name	Value
ID	Auto generated
MappingEntityId	Null
Application	Sample Repository
Name	User Entity

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Column Name	Value
Mapping Table	SKVirtualActor
DefinitionXML	<entity type="user"><properties><property name="UserId" type="string" purpose="identifier" /><property name="Name" type="string" purpose="name"/><property name="Email" type="string" purpose="email"/></properties></entity>%%$#@@!!!*&
AddInProviderId	Default AddInProviderId
LastUpdatedDateTime	Date time value

The above code creates an entry in **SKEntityDataSourceProvider** table. The entry in Repository DB can be verified in database by running the query select * from `SKEntityDataSourceProvider Where DisplayName='Custom Provider'`

Column Name	Value
ID	Auto generated
Application	Sample Repository
Entity ID	Copy the Id Column value of SKEntity of RoleEntity row here.
DisplayName	Custom Provider
DefinitionXML	<dataproviders><provider name="customsqlprovider" type="Custom Sql Provider"><properties><property name="UserId" type="guid" purpose="identifier"/><property name="Name" type="string" purpose="name"/><property name="Email" type="string" purpose="email"/><property name="Password" type="string" purpose="password"/></properties><datasource>server=localhost;database=SkeltaUsers;uid=sa;pwd=XXX</datasource><SourceTable>SqlUsers</SourceTable><loginproperty>Email</loginproperty></provider></dataproviders>
Provider	customsqlprovider
Instance Name	sqlproviderCustom
LastUpdatedDateTime	Date time value

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Verify whether the Custom User Entity Data Source Provider is configured correctly or not

Verification:

The above code updates the entries required for configuring Custom User Entity Data Source Provider. Now, we should be able to see the provider in Repository list page. The previous entry of "List" will be updated with "Custom Provider".

If you get a message "No Provider Available" in the repository list page, cross check the Logger Console for a detailed exception for the cause.

- Check the calls made to the custom user entity data source provider.

If there is no exception then we could also create the Actor object API to check the calls made to the custom user entity data source provider.

DLL REFERENCE

Skelta.HWS

Workflow.NET.NET2

The parameters passed to the Actor object are application object & real user id. Real user id can be formed by prefixing provider instance name and special characters "::" to the unique identifier value of user. We have used UserId column value for Kevin.

	UserId	Name	Role	Email	Password
	e9d21049-e3cf-4f4d-9d51-036d0145c547	Michelle	Admin	michelle@skelta....	michelle
	c6a7170a-6a35-4f79-8688-150d0ac8fe15	Kevin	Developer	kevin@skelta.com	kevin
	ce24b3d0-21dc-4821-8d55-afe4c8581716	John	Developer	john@skelta.com	john
	9d3eebb3-ebbe-4415-9d18-df4205af5c46	Elaine	Product Manager	elaine@skelta.com	elaine

Sample code:

```
Skelta.HWS.Actor actor = new Actor(new ApplicationObject("Sample Repository"),
"sqlproviderCustom::c6a7170a-6a35-4f79-8688-150d0ac8fe15");
object identifierValue = actor.Resource.Properties.Identifier.Value;
string Name = actor.Resource.Properties.Name.Value.ToString();
string Email = actor.Resource.Properties.Email.Value.ToString();
```

The above code also creates an entry to SKVirtualActor table, this can be checked using query. Select * from `SKVirtualActor Where Name='Kevin'`

Create User and Role Entity Data Source Provider

Introduction

In this section, you will learn:

1. How to use AVEVA Work Tasks API for creating a repository from scratch and then configure the custom entity data source provider
2. How to create & configure custom user entity & role entity data source provider
3. XML schema used for the definition Xml of Entity & Entity Data Source Provider

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps To Create Custom Data Source Provider

Step 1: Create Data source for the Repository
Step 2: Create a Repository and Map the data source to the repository
Step 3: Create Custom User Entity Data Source Provider for authentication and user information
Step 4: Creating Default Admin for the Repository
Step 5: Create Custom Role Entity Data Source Provider

Step 1: Create Data Source for the Repository

The following example demonstrates the creation of a data source using a connection string. The data source created will be listed in the Enterprise Console and the repository can be created using the newly created data source.

PROCEDURE

DLL REFERENCE

Skelta.FarmConfiguration.dll

Workflow.NET.NET2.dll

NAMESPACE USED

```
Skelta.FarmConfiguration;
Skelta.FarmManager;
System.Collections.Generic;
System.Reflection;
```

Try the following code in a console application by adding the above references and name spaces:

```
//Need to handle assembly resolve event for appdomain if database used is SQL Server 2005
System.AppDomain.CurrentDomain.AssemblyResolve += new
ResolveEventHandler(CurrentDomain_AssemblyResolve);
//Build the connection string for repository data source
Skelta.FarmConfiguration.HandleSkeltaFarm.ConnectionStringDetails
connectionStringDetails = new HandleSkeltaFarm.ConnectionStringDetails();
connectionStringDetails.DataSource = "SkeltaNB30";  // System name
connectionStringDetails.DatabaseName = "RepoDatabase";
connectionStringDetails.DatabaseType = "SQL Server";
connectionStringDetails.IntegratedSecurity = false; //For SQL Server Authentication it is
 False ; For Windows Authentication it is true
connectionStringDetails.UserId = "sa";
connectionStringDetails.Password = "XXX"; //Enter correct password
HandleSkeltaFarm handlerSkeltaFarm = new HandleSkeltaFarm();
handlerSkeltaFarm.PrepareConnectionString(connectionStringDetails);
//Parameters passed are data source name & description
handlerSkeltaFarm.CreateDatasource("SkeltaDatasource", "data source created for
repository");

static System.Reflection.Assembly CurrentDomain_AssemblyResolve(object sender,
ResolveEventArgs args)
{
AssemblyName requestedName = new AssemblyName(args.Name);
if (requestedName.Name.Contains("Microsoft.SqlServer.ConnectionInfo"))
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
if (requestedName.Version.ToString() == "9.0.242.0")
{
throw new Exception("Can not load 9.0.242.0 dll too");
}
else
{
AssemblyName loadThis = new AssemblyName("Microsoft.SqlServer.ConnectionInfo");
loadThis.Version = new Version("9.0.242.0");
loadThis.CultureInfo = requestedName.CultureInfo;
loadThis.SetPublicKeyToken(requestedName.GetPublicKeyToken());
return Assembly.Load(loadThis);
}
}
else
{
if (requestedName.Name.Contains("Microsoft.SqlServer.Smo"))
{
if (requestedName.Version.ToString() == "9.0.242.0")
{
throw new Exception("Can not load 9.0.242.0 dll too");
}
else
{
AssemblyName loadThis = new AssemblyName("Microsoft.SqlServer.Smo");
loadThis.Version = new Version("9.0.242.0");
loadThis.CultureInfo = requestedName.CultureInfo;
loadThis.SetPublicKeyToken(requestedName.GetPublicKeyToken());
return Assembly.Load(loadThis);
}
}
else
return null;
}
}
```

Verification:

The above code creates a database named RepoDatabase. The entry in Farm DB can be verified in database by running the query Select * from SKDataSource Where Name='SkeltaDatasource'. The same can be verified in Central Config data source list page.

Step 2: Create a Repository and Map the Data Source to the Repository

The below code creates a repository entry, during creation of repository only the default data source providers can be configured. To set the custom data source provider we need to update the repository which will be done later.

PROCEDURE

The AssemblyResolve (System.AppDomain.CurrentDomain.AssemblyResolve) event, reference and namespace used in the previous example [Step 1: Create Data Source for the Repository] are required in this sample as well.

Try the following code in a console application:

```
HandleSkeltaFarm handlerSkeltaFarm = new HandleSkeltaFarm();
 HandleSkeltaFarm.RepositoryDetails repositoryDetails = new
HandleSkeltaFarm.RepositoryDetails();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 repositoryDetails.Name = "Telematics HR Repository";
 repositoryDetails.Description = "Telematics HR Process Repository";
 repositoryDetails.DisplayName = "Telematics HR Repository";
 // Name of the data source created  in the previous step
 repositoryDetails.Datasource = "SkeltaDatasource";
 repositoryDetails.IsHidden = false;
 repositoryDetails.Sourceelements = "WORKFLOWELEMENTS\\Default";
 //Enter the email id to which exception emails must be sent.
 //These exception emails will be generated from the engine when the activity execution
fails.
 repositoryDetails.Fromemail = "admin@domain.com";
 repositoryDetails.Toemail = "admin@domain.com";
repositoryDetails.Alertuser = true;
// Select servers and repository mapping.
Dictionary<Guid, Server> serversmapped = new Dictionary<Guid, Server>();
// To select all servers/clients of current farm,
// ServerCollection servers = new ServerCollection(Farm.Local);
// To select servers only
ServerCollection servers = new ServerCollection(Farm.Local);
foreach (Server server in servers.Items)
serversmapped.Add(server.Id, server);
repositoryDetails.ServersMapped = serversmapped;
List<Guid> enterpriseConsoles = new List<Guid>();
// This code returns all EnterpriseConsole sites in the Farm.
// Ignore this part if you have not installed enterprise consoles.
WebSiteCollection websites = new WebSiteCollection("EC");
foreach (WebSite website in websites.Items)
enterpriseConsoles.Add(website.Id);
repositoryDetails.EnterpriseConsoles = enterpriseConsoles;
repositoryDetails.Provider = HandleSkeltaFarm.RepositoryProviders.SkeltaList;
handlerSkeltaFarm.RepositorySettings = repositoryDetails;
handlerSkeltaFarm.ManageRepository();
```

Verification:

The above code creates a repository named Telematics HR Repository. The entry in Farm DB can be verified in the database by running the query Select * from SKRepository Where Name=Telematics HR Repository'. The same can be verified in the Enterprise Console repository list page.

Step 3: Create Custom User Entity Data Source Provider for Authentication and User Information

Using this API a custom User Entity Data Source provider can be created.

1. Create a table named SqlUsers

   The custom entity provider fetches the user information from a table called SqlUsers.

   Script to create this table is provided below.

   ```
   GO
   /****** Object:  Table [dbo].[SqlUsers]  ******/
   SET ANSI_NULLS ON
   GO
   SET QUOTED_IDENTIFIER ON
   GO
   CREATE TABLE [dbo].[SqlUsers](
   [UserId] [uniqueidentifier] ROWGUIDCOL  NOT NULL CONSTRAINT [DF_SqlUsers_Id]  DEFAULT
   (newid()),
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 [Name] [nvarchar](100) NULL,
 [Role] [nvarchar](200) NULL,
 [Email] [nvarchar](75) NULL,
 [Password] [nchar](10) NULL,
 CONSTRAINT [PK_SqlUsers_1] PRIMARY KEY CLUSTERED
(
[UserId] ASC
) WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

2. Insert default users to SqlUsers Table

   Script to insert default users to SqlUsers Table:

```
INSERT INTO [dbo].[SqlUsers]
 ([Name]
 ,[Role]
 ,[Email]
 ,[Password])
 VALUES
 ('Michelle','Admin','michelle@skelta.com','michelle'),
('kevin','Developer','kevin@skelta.com','kevin'),
 ('john','Developer','john@skelta.com','john'),
 ('Elaine','Product Manager','elaine@skelta.com','elaine')
GO
```

	UserId	Name	Role	Email	Password
▶	d51-036d0145c547	Michelle	Admin	michelle@skelta....	michelle
	c6a7170a-6a35-4f79-8688-150d0ac8fe15	Kevin	Developer	kevin@skelta.com	kevin
	ce24b3d0-21dc-4821-8d55-afe4c8581716	John	Developer	john@skelta.com	john
	9d3eebb3-ebbe-4415-9d18-df4205af5c46	Elaine	Product Manager	elaine@skelta.com	elaine

3. Create a dll named CustomUserEntityDataSourceProvider

   PROCEDURE

   DLL REFERENCE

   Workflow.NET.NET2

   System.Transactions.dll

   System.Web.dll

   NAMESPACE USED

   System

   System.Collections.Generic

   System.Text

   System.Xml

   System.Xml.XPath

   System.Transactions

   System.Data

   Workflow.NET

   Workflow.NET.Interfaces

   Skelta.Entity

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

API INFORMATION

The User Entity Data Source Provider should

Inherit Skelta.Entity.DataSourceProviderBase class

or

Implement Skelta.Entity.IEntityDataSourceProvider, Skelta.Core.ISkeltaAddInProvider

Implement interface Skelta.Entity.UserEntityProvider.IUserEntityDataSourceProvider

Implement interface Skelta.Entity.IEntityAdminDataSourceProvide.

Skelta.Entity.DataSourceProviderBase: Base class used by the default providers, this class implements the interfaces Skelta.Entity.IEntityDataSourceProvider, Skelta.Core.ISkeltaAddInProvider. All the methods in this class are declared as virtual. The implementers should ideally implement the interfaces implemented by the base class directly. The base class is used in the sample trying to make it easier for the implementer with some of the methods already implemented with default functionality. The method which needs to be mandatorily overridden throws exception if relied on the base class. This class exposes methods for initializing the provider, fetching user information & querying users based on filters.

Skelta.Entity.UserEntityProvider.IUserEntityDataSourceProvider: Gets called during EnterpriseConsole authentication from the Log on page. This interface provides methods which gives username and password entered in the login page of EnterpriseConsole. All the user authentication code must be implemented in these methods.

Skelta.Entity.IEntityAdminDataSourceProvider: Required for setting the default administrator for a Repository. This administrator will have complete rights on the EC site. This administrator will be responsible for setting security and adding users to EC.

Note: Since AVEVA Work Tasks executes activities in Transaction scope, there is a possibility for the Data Source Provider methods getting called with in a transaction. For the sample provided, there are no specific transaction scope used. The isolation level used for Transactions in AVEVA Work Tasks is read committed, so, if any other isolation level is used in the providers it could results in exceptions in Runtime. The Best practices are:

Do not use a separate transaction for querying user information.

Suppress the transaction if the user information is taken from an external object or source. Use the Transaction Scope option available in common functions for this.

```
using (System.Transactions.TransactionScope tScope =
Workflow.NET.CommonFunctions.GetSuppressedTransactionScope())
{
//Call to external object to get user information tScope.Complete();
}
```

SAMPLE CODE:

```
namespace CustomUserEntityDataSourceProvider
{
/// <summary>
/// Custom entity data source  provider which queries sql database table/view returning
user information.
/// </summary>
public class UserEntityDataSourceProvider : DataSourceProviderBase,
Skelta.Entity.UserEntityProvider.IUserEntityDataSourceProvider,
Skelta.Entity.IEntityAdminDataSourceProvider
{
private string sConnectionString;
private string sDataSourceType;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
private string sSourceTable;
private string sCols = "";
private Log log = null;
private string loginProperty = "";

public UserEntityDataSourceProvider() { }

/// <summary>
/// Gets called  while provider is initialized.
/// </summary>
/// <param name="settings">Definition XML configured for SKEntityDataSourceProvider
table.</param>
public override void Initialize(string settings)
{
log = new Log();
log.FileName = "CustomUserEntityDataSourceProvider.cs";
XmlDocument doc = new XmlDocument();
try
{
doc.LoadXml(settings);
// Get the Connection String from the definition xml file.
XmlNode DsConnNode = doc.GetElementsByTagName("datasource").Item(0);
sConnectionString = DsConnNode.InnerText;
sDataSourceType = "sql server";
sSourceTable = doc.GetElementsByTagName("SourceTable").Item(0).InnerText;
}
catch (Exception exp)
{
log.LogError(exp, "Could not read the xml data while initializing the provider");
throw;
}
try
{
if (doc.GetElementsByTagName("loginproperty") != null)
{
XmlNode qxn = doc.GetElementsByTagName("loginproperty").Item(0);
loginProperty = qxn.InnerText;
}
}
catch (Exception ex)
{
log.LogError(ex, "Could not get loginproperty column set.");
throw;
}
}

/// <summary>
/// Called when delayed initialization on a resource property is required.
/// For example if there are 2 providers and the initial property collection for the
resource
/// has a property which is empty and if this property is requested a delayed
initialization is
/// triggered on the  provider based on the mapping  fields configured.
/// </summary>
/// <param name="filter">Filter which is internally built based on the mapping fields
configured.</param>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// <param name="dataparams">Parameters used in the filter.</param>
/// <returns>Entity item with the delayed initialization property  value.</returns>
public override EntityItem GetEntityItem(string filter, object[] dataParams)
{
if(string.IsNullOrEmpty(sCols))
BuildPropertyColumn();
List<IDataParameter> dParam = new List<IDataParameter>();
string sql = string.Empty;
using (IDataHandler dbHandler =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString,
sDataSourceType))
{
sql = "SELECT " + sCols + " FROM " + sSourceTable + " WHERE " + filter;
try
{
foreach (object[] values in dataParams)
{
dParam.Add(dbHandler.GetParameter(values[0].ToString(), values[1]));
}
return GetEntityItem(sql, dParam.ToArray());
}
catch (Exception exp)
{
log.LogError(exp, "Could not fetch entity item with filter (" + filter + ").");
throw;
}
}
}

/// <summary>
/// Called when actor object is created or any resource details are requested.
/// </summary>
/// <param name="entityItemId">Guid of the resource based on the current sample.</param>
/// <returns>User Entity Details.</returns>
public override EntityItem GetEntityItem(object entityItemId)
{
if (string.IsNullOrEmpty(sCols))
BuildPropertyColumn();
string sql = string.Empty;
IDataParameter dParam = null;
using (IDataHandler dbHandler =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString,
sDataSourceType))
{
try
{
sql = "SELECT " + sCols + " FROM " + sSourceTable + " WHERE " +
EntityItemProperties.Identifier.PropertyName + " = @resourceId";
dParam = dbHandler.GetParameter("@resourceId", entityItemId);
return GetEntityItem(sql, dParam);
}
catch (Exception exp)
{
log.LogError(exp, "Could not fetch entity item for entity item id (" + entityItemId +
").");
throw;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
}

}
EntityItem GetEntityItem(string filter, params IDataParameter[] dataparams)
{
IDataReader idr = null;
EntityItem res;
Workflow.NET.Interfaces.IDataHandler ConnectionWrapper = null;
using (ConnectionWrapper =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString,
sDataSourceType))
{
idr = ConnectionWrapper.ExecuteReader(filter, dataparams);
//Create a empty EntityItem object
res = new EntityItem();
res.Properties = EntityItemProperties.CreateDeepClone();
string fieldname;
EntityItemProperty rp;
int loopcount = 0;

while (idr.Read())
{
//For every record, update the properties values
for (int i = 0; i < idr.FieldCount; i++)
{
fieldname = idr.GetName(i);
if (res.Properties.ContainsKey(fieldname))
{
rp = ((EntityItemProperty)res.Properties[fieldname]);
if (!rp.Values.Contains(idr.GetValue(i)) && idr.GetValue(i) != System.DBNull.Value &&
idr.GetValue(i).ToString() != "") //Add all unique values of a given field in Values
collection
rp.Values.Add(idr.GetValue(i));
}
else
{  //if( loopcount==0 )
log.LogWarning("Field " + fieldname + " not found in EntityItem Properties");
}
}
loopcount++;
}
idr.Close();
idr.Dispose();
}
return res;
}

/// <summary>
/// Called  when  an filter is executed in  Runtime, for example when Human activity is
executed.
/// This method  also gets called when user look search is performed
/// </summary>
/// <param name="filter">Filter for querying the resource.</param>
/// <param name="query"></param>
/// <param name="FullyProcessedQuery"></param>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// <param name="Params"></param>
/// <returns></returns>
public override object[] QueryEntityItem(string filter, string query, string
FullyProcessedQuery, object[] Params)
{
// When QueryEntityItem is called from Runtime (Engine),
//there is an option to get the executing context details using
GetEntityExecutionContext  method.

Skelta.Entity.EntityExecutionContextInfo entityExecutionContextInfo =
Skelta.Entity.EntityExecutionContext.GetEntityExecutionContext();
//if (entityExecutionContextInfo != null)
//{
//entityExecutionContextInfo.ApplicationName;
//entityExecutionContextInfo.WorkflowName;
//entityExecutionContextInfo.ExecutionId
//}
IDataReader idr = null;
Workflow.NET.Interfaces.IDataHandler ConnectionWrapper = null;
try
{
using (ConnectionWrapper =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString,
sDataSourceType))
{
string sql = "SELECT DISTINCT " + EntityItemProperties.Identifier.PropertyName + " FROM
" + sSourceTable;
System.Collections.ArrayList EntityItems = new System.Collections.ArrayList();
if (FullyProcessedQuery != null && FullyProcessedQuery.Trim().Length > 0)
{
sql += " WHERE " + FullyProcessedQuery;
}
else
{
log.LogWarning("EntityItem filter is emtpy.");
return EntityItems.ToArray();
}

System.Collections.ArrayList alParams = new System.Collections.ArrayList();
foreach (object Param in Params)
{
object[] obj = (object[])Param;
alParams.Add(ConnectionWrapper.GetParameter((string)obj[0], obj[1]));
}
IDataParameter[] Parameters =
(IDataParameter[])alParams.ToArray(typeof(IDataParameter));
idr = ConnectionWrapper.ExecuteReader(sql, Parameters);

while (idr.Read())
{
EntityItems.Add(idr.GetValue(0));
}
idr.Close();
idr.Dispose();
return EntityItems.ToArray();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
}
catch (Exception exp)
{
if (entityExecutionContextInfo != null)
log.LogError(exp, "Could not fetch entity item while querying entity item,  filter(" +
filter + ").", entityExecutionContextInfo.ApplicationName,
entityExecutionContextInfo.WorkflowName, entityExecutionContextInfo.ExecutionId);
else
log.LogError(exp, "Could not fetch entity item while querying entity item,  filter(" +
filter + ").");
throw;
}
}
void BuildPropertyColumn()
{
if (string.IsNullOrEmpty(sCols))
{
foreach (EntityItemProperty prop in EntityItemProperties)
{
sCols += (sCols == "" ? "" : ",") + prop.PropertyName;
}
}
}

/// <summary>
/// Creates a clone  of the user entity provider.
/// </summary>
/// <returns></returns>
public override IEntityDataSourceProvider Clone()
{
UserEntityDataSourceProvider customProvider = new UserEntityDataSourceProvider();
customProvider.sCols = this.sCols;
customProvider.Name = this.Name;
customProvider.sConnectionString = this.sConnectionString;
customProvider.sDataSourceType = this.sDataSourceType;
customProvider.sSourceTable = this.sSourceTable;
EntityItemPropertiesCollection entityItems =
this.EntityItemProperties.CreateDeepClone();
foreach (EntityItemProperty eProp in entityItems)
{
customProvider.EntityItemProperties.Add(eProp.PropertyName, eProp);
}
foreach (string key in this.DataSourceMappingFields.Keys)
{
customProvider.DataSourceMappingFields.Add(key, this.DataSourceMappingFields[key]);
}
foreach (string key in this.DataSourceProperties.Keys)
{
customProvider.DataSourceProperties.Add(key, this.DataSourceProperties[key]);
}
foreach (string key in this.RuleParts.Keys)
{
customProvider.RuleParts.Add(key, this.RuleParts[key]);
}
customProvider.IsPrimaryDataSource = this.IsPrimaryDataSource;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
customProvider.DataSourceType = this.DataSourceType;
customProvider.loginProperty = this.loginProperty;
customProvider.Name = this.Name;
customProvider.CurrentEntity = this.CurrentEntity;
customProvider.DisplayName = this.DisplayName;
customProvider.RepositoryName = this.RepositoryName;
customProvider.IsAuthenticationSource = this.IsAuthenticationSource;
customProvider.ProviderPrefix = this.ProviderPrefix;
return customProvider;
}
/// <summary>
/// Converts to provider specific identifier type;
/// </summary>
/// <param name="identifierId">real entity id converted as string.</param>
/// <returns>Id which  is provider specific type converted.</returns>
public override object ConvertToIdentifierType(string identifierId)
{
switch (EntityItemProperties.Identifier.PropertyType)
{
case "string": return identifierId;
case "int": return int.Parse(identifierId);
case "guid": return new Guid(identifierId);
}
return null;
}

/// <summary>
/// Return the unique identifier for the user. The default return value is user name
itself.
/// The actual unique identifier for the user could be different from the property used
for login. For  the sample though the unique identifier is guid we have provided email
as the login property for ease of use.
/// </summary>
/// <param name="userName">Value entered in the login page against user name
field.</param>
/// <param name="password">Password entered in the login page against password
field.</param>
/// <returns>Unique identifier  value. Return value type is object as the identifier
type could be guid, int or string.</returns>
public override object GetUniqueIdentifier(string userName, string password)
{
if (string.IsNullOrEmpty(loginProperty))
return userName;
string passwordColumn = GetPasswordColumnName();
Workflow.NET.Interfaces.IDataHandler ConnectionWrapper = null;
string sql = "";
IDataParameter param1 = null;
IDataParameter param2 = null;
IDataReader idr = null;
try
{
ConnectionWrapper =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString,
sDataSourceType);
sql = "SELECT * FROM " + sSourceTable + " WHERE " + loginProperty + "=@resourceId AND "
+ passwordColumn + "=@password";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
param1 = ConnectionWrapper.GetParameter("@resourceId", userName);
param2 = ConnectionWrapper.GetParameter("@password", password);
idr = ConnectionWrapper.ExecuteReader(sql, param1, param2);
if (idr.Read())
return idr[EntityItemProperties.Identifier.PropertyName];
else
throw new System.Exception("Unable to get unique identifier from property (" +
loginProperty + "), query(" + sql + ")");

}
catch (Exception e)
{
log.LogError(e, "Could not get unique identifier for user (" + userName + ")");
throw;
}
finally
{
if (idr != null)
idr.Close();

if (ConnectionWrapper != null)
ConnectionWrapper.Dispose();
}
}
//Returns the column name from property collection with purpose set as "password"
string GetPasswordColumnName()
{
foreach (EntityItemProperty eProp in EntityItemProperties)
{
if (!string.IsNullOrEmpty(eProp.Purpose) &&
eProp.Purpose.ToLowerInvariant() == "password")
return eProp.PropertyName;

}

return "";
}

#region IUserEntityDataSourceProvider Members

/// <summary>
/// Currently this method call is not used in any  of the  default providers.
/// It was added to provide an option in future release to perform login  for a provider
/// based on different authentication methods available in IIS instead of performing a
validation
/// based on user name & password
/// </summary>
/// <param name="httpContext"></param>
/// <returns></returns>
bool
Skelta.Entity.UserEntityProvider.IUserEntityDataSourceProvider.ValidateIISUser(System.We
b.HttpContext httpContext)
{
return false;
}
/// <summary>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// Called when submit button event of login page is fired from EnterpriseConsole.
/// </summary>
/// <param name="userName">Value entered in the login page against user name
field.</param>
/// <param name="password">Password entered in the login page against password
field.</param>
/// <returns>true if user credentials are valid else false.</returns>
bool Skelta.Entity.UserEntityProvider.IUserEntityDataSourceProvider.ValidateUser(string
username, string password)
{
string passwordColumn = GetPasswordColumnName();
string identifierColumn = "";
if (string.IsNullOrEmpty(loginProperty))
identifierColumn = EntityItemProperties.Identifier.PropertyName;
else
identifierColumn = loginProperty;

if (string.IsNullOrEmpty(passwordColumn))
return false;
else
{
Workflow.NET.Interfaces.IDataHandler ConnectionWrapper = null;
string sql = "";
IDataParameter param1 = null;
IDataParameter param2 = null;
IDataReader idr = null;
using (ConnectionWrapper =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString,
sDataSourceType))
{
try
{
sql = "Select * From " + sSourceTable + " WHERE " + identifierColumn + "=@resourceId AND
" + passwordColumn + "=@password";
param1 = ConnectionWrapper.GetParameter("@resourceId", username);
param2 = ConnectionWrapper.GetParameter("@password", password);
idr = ConnectionWrapper.ExecuteReader(sql, param1, param2);
bool returnFlag = false;
if (idr.Read())
returnFlag = true;
idr.Close();
return returnFlag;
}
catch (Exception e)
{
log.LogError(e, "Could not validate user (" + username + ")");
//Need not throw the exception as the UI will display error message if return value is
false.
}
}
return false;
}
}
#endregion

#region IEntityAdminDataSourceProvider Members
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// <summary>
/// Return admin users unique identfiier value.
/// This administrator will have complete rights on the EC site.
/// This administrator will be responsible for setting security and adding users to EC
/// </summary>
/// <returns>Unique identifier value of administartor</returns>
public object[] GetAdministrator()
{
//For the  sample we'll use the column named Role with value set as Admin.
Workflow.NET.Interfaces.IDataHandler ConnectionWrapper = null;
System.Collections.ArrayList EntityItems = new System.Collections.ArrayList();
try
{
ConnectionWrapper =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(sConnectionString,
sDataSourceType);
string stradmin = "Admin";
string sql = "SELECT DISTINCT " + EntityItemProperties.Identifier.PropertyName + " FROM
" + sSourceTable;
sql += " WHERE Role = '" + stradmin + "'";
System.Collections.ArrayList alParams = new System.Collections.ArrayList();
IDataParameter[] Parameters =
(IDataParameter[])alParams.ToArray(typeof(IDataParameter));
IDataReader idr = ConnectionWrapper.ExecuteReader(sql, Parameters);
while (idr.Read())
{
EntityItems.Add(Workflow.NET.CommonFunctions.GetGuid(idr,
EntityItemProperties.Identifier.PropertyName));
}
idr.Close();
idr.Dispose();
}
catch (Exception e)
{
log.LogError(e, "Could not get the administrator user information.");
throw;
}
finally
{
if (ConnectionWrapper != null)
ConnectionWrapper.Dispose();
}
return EntityItems.ToArray();
}
#endregion
}
}
```

Configure Custom User Entity Data Source Provider

Configuring custom user entity data source provider involves entries in three tables of the Repository.

Table : SKAddInProviders , Object : Skelta.Core.AddInProvider

Table : SKEntity, Object : Skelta.Entity.Entity

Table : SKEntityDataSourceProvider, Object : Skelta.Entity.DataSourceProvider

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Add the assembly information of the user entity custom provider created, in the SKAddInProviders table

Provider type must be "UserDSProvider" and name should be any unique name. The newly added provider name must be again updated in the SKEntityDataSourceProvider table.

You can add an entry to SKAddInProviders table by using API or Database script. We have given both methods below, use any one of them and verify it using the query given in the verification topic

Using API

```
Skelta.Core.AddInProvider addInProvider = new Skelta.Core.AddInProvider(
new ApplicationObject("Sample Repository"));
addInProvider.Type = "UserDSProvider";
addInProvider.Name = "customsqlprovider";
#warning : Make sure to change the assembly  path
addInProvider.Description = "Sample Custom User Entity Data Source Provider";
addInProvider.ClassName =
 "CustomUserEntityDataSourceProvider.UserEntityDataSourceProvider";
addInProvider.AssemblyNameWithAbsoluteOrRelativePath =  @"M:\Michelle\
CustomUserEntityDataSourceProvider\
 CustomUserEntityDataSourceProvider\bin\Debug\CustomUserEntityDataSourceProvider.dll";
addInProvider.IsGacAssembly = false;
addInProvider.IsGlobal = true;
addInProvider.IsDefault = true; // The properties IsGacAssembly, IsGlobal, IsDefault  is
 //not currently in use. The value used in sample should be used "as is"
addInProvider.Save();
```

Using Script

```
INSERT INTO SKAddInProviders VALUES (newid(),'UserDSProvider','customsqlprovider',
'Sample Custom User Entity Data Source Provider','CustomUserEntityDataSourceProvider.
UserEntityDataSourceProvider','M:\Michelle\CustomUserEntityDataSourceProvider\
CustomUserEntityDataSourceProvider\bin\Debug\
CustomUserEntityDataSourceProvider.dll',null,0,1,1,getutcdate())
```

Note: Assembly path should be changed.

Verification

The above code creates an entry in SKAddInProviders table with type as 'UserDSProvider'.The entry in Repository DB can be verified in database by running the query select * from SKAddInProviders Where Name='customsqlprovider'

Column Name	Value
ID	Auto Generated
Type	UserDSProvider
Name	customsqlprovider
Description	Sample Custom User Entity Data Source Provider
ClassName	CustomUserEntityDataSourceProvider. UserEntityDataSourceProvider

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Column Name	Value
Assembly	M:\Michelle\ CustomUserEntityDataSourceProvider\ CustomUserEntityDataSourceProvider\ bin\Debug \CustomUserEntityDataSourceProvider.dll
IsGacAssembly	False
Isdefault	True
IsGlobal	True
LastUpdatedDataTime	Date time value

Adding entry to SKEntity and SKEntityDataSourceProvider

Steps:

Update the DefinitionXml column of SKEntity table with appropriate resource XML data for the newly created custom user entity provider that has been updated in the SKAddInProviders table.

Update the DefinitionXml column, Provider, Instance Name, Display Name for default entry of the List in SKEntityDataSourceProvider table with appropriate custom user entity data source provider entries.

To do the above steps, you can use API or Database script, both methods are given below, use any one of them.

The definition XML that has been used in the code,to update the DefinitionXml column of SKEntity and SKEntityDataSourceProvider table are given below for your reference.

Custom User Entity Data Source Provider Definition XML

```
<dataproviders>
<provider name="customsqlprovider" type="Custom Sql Provider">
<properties>
<property name="UserId" type="guid"  purpose="identifier" />
<property name="Name" type="string"  purpose="name"/>
<property name="Email"  type="string"  purpose="email"/>
<property name="Password" type="string"  purpose="password"/>
</properties>
<datasource>server=localhost;database=SkeltaUsers;uid=sa;pwd= XXXX</datasource>
<SourceTable>SqlUsers</SourceTable>
<loginproperty>Email</loginproperty>
</provider>
</dataproviders>
```

"provider" node has attribute name value set as customsqlprovider as this should match the Name column entry in SKAddInProvider.

As described in earlier sections, the purpose attributes identifier, name & email are mandatory.

Custom purpose named password is used to identify the column containing password info for authentication.

"datasource" node contains connection string information.

"SourceTable" node contains the table or view which contains user details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

"loginproperty" node provides information of the column which will be used as user name while logging in to the repository.

Custom Entity Definition XML

```
<entity type="user">
<properties>
<property name="UserId" type="string"  purpose="identifier" />
<property name="Name" type="string"  purpose="name"/>
<property name="Email"  type="string"  purpose="email"/>
</properties>
</entity>%%$#@@!!!*&
```

Contains properties which are mapped to user entity definition XML properties.

As described in earlier sections, the purpose attributes identifier, name and email are mandatory.

Using API for steps 1 & 2

```
Skelta.Core.ApplicationObject applicationObject = new Skelta.Core.ApplicationObject("Sample
Repository");
 Skelta.Entity.Entity entity = Skelta.Entity.Entity.GetEntity(applicationObject,
"UserEntity");
 string entityDefinitionXML  = "<entity type=\"user\">";
 entityDefinitionXML += "<properties>";
 entityDefinitionXML += "<property name=\"UserId\" type=\"string\"  purpose=\"identifier\"
/>";
 entityDefinitionXML += "<property name=\"Name\" type=\"string\"  purpose=\"name\"/>";
 entityDefinitionXML += "<property name=\"Email\"  type=\"string\"  purpose=\"email\"/>";
 entityDefinitionXML += "</properties>";
 entityDefinitionXML += "</entity>%%$#@@!!!*&";
 //Update default entity definition xml with matching xml for new  custom user entity data
source provider
entity.DefinitionXml = entityDefinitionXML;
entity.Save();

 Skelta.Entity.DataSourceProviderCollection entityDataSourceProvider = null;
entityDataSourceProvider = new
Skelta.Entity.DataSourceProviderCollection(applicationObject,entity.Id);
 //Entity user xml need to be changed based on the custom User table
#warning : Make sure to change the connection string
 string entityDataSourceProviderDefinitionXML = "<dataproviders>";
 entityDataSourceProviderDefinitionXML += "<provider name=\"customsqlprovider\"
type=\"Custom Sql Provider\">";
 entityDataSourceProviderDefinitionXML += "<properties>";
 entityDataSourceProviderDefinitionXML += "<property name=\"UserId\" type=\"guid\"
 purpose=\"identifier\"/>";
 entityDataSourceProviderDefinitionXML += "<property name=\"Name\" type=\"string\"
 purpose=\"name\"/>";
 entityDataSourceProviderDefinitionXML += "<property name=\"Email\"  type=\"string\"
 purpose=\"email\"/>";
 entityDataSourceProviderDefinitionXML += "<property name=\"Password\" type=\"string\"
 purpose=\"password\"/>";
 entityDataSourceProviderDefinitionXML += "</properties>";
 entityDataSourceProviderDefinitionXML +=
"<datasource>server=localhost;database=SkeltaUsers;uid=sa;pwd=XXXX</datasource>";
 entityDataSourceProviderDefinitionXML += "<SourceTable>SqlUsers</SourceTable>";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 entityDataSourceProviderDefinitionXML += "<loginproperty>Email</loginproperty>";
 entityDataSourceProviderDefinitionXML += "</provider>";
 entityDataSourceProviderDefinitionXML += "</dataproviders>";
 entityDataSourceProvider .Items[0].DefinitionXml = entityDataSourceProviderDefinitionXML;
 //Custom provider name which is added in addinprovider table
 entityDataSourceProvider .Items[0].Provider = "customsqlprovider";
 entityDataSourceProvider .Items[0].InstanceName = "sqlproviderCustom";
 entityDataSourceProvider .Items[0].DisplayName = "Custom Provider";
 entityDataSourceProvider .Items[0].Entity = entity;
 // Update skelta list entry  with the custom entry in SKEntityDataSourceProvider for user
provider
entityDataSourceProvider.Items[0].Save();
```

The provider name value customsqlprovider should be same as the AddInProvider Name column value.

Using Table Entry for steps 1 & 2:

The definition XML for entity should be matching the properties for entity data source provider.

```
Update SKEntity Set DefinitionXML = '<entity type="user"><properties><property
name="UserId" type="string"  purpose="identifier" /><property name="Name" type="string"
 purpose="name"/><property name="Email"  type="string"
 purpose="email"/></properties></entity>%%$#@@!!!*&%%$#@@!!!*&'
Where Name = 'UserEntity'
```

Make sure to update the connection string information in definition XML with the appropriate value.
The provider column value should be the same as the name column value inserted in SKAddInProviders table.

Note: Also, ensure that the query given below updates the row for the List User Entity Data Source Provider and not the Role Entity Data Source Provider.

```
Update SKEntityDataSourceProvider Set DefinitionXML ='<dataproviders><provider
name="customsqlprovider" type="Custom Sql Provider"><properties><property name="UserId"
type="guid"  purpose="identifier"/><property name="Name" type="string"
 purpose="name"/><property name="Email"  type="string"  purpose="email"/><property
name="Password" type="string"
 purpose="password"/></properties><datasource>server=localhost;
database=SkeltaUsers;uid=sa;pwd=XXX</datasource><SourceTable>SqlUsers</SourceTable>
<loginproperty>Email</loginproperty></provider></dataproviders>',
DisplayName='Custom Provider', Provider='customsqlprovider',
InstanceName='sqlproviderCustom' Where DisplayName ='Skelta List' and EntityId=(Select Id
from SKEntity where Name='UserEntity')
```

Note: Connection string should be changed.

Verification

The above code creates an entry in SKEntity table. The entry in Repository DB can be verified in database by running the query select * from SKEntity Where Name='UserEntity'and Application='Sample Repository'

Column Name	Value
ID	Auto generated
MappingEntityId	Null
Application	Sample Repository
Name	User Entity

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Column Name	Value
Mapping Table	SKVirtualActor
DefinitionXML	<entity type="user"><properties><property name="UserId" type="string" purpose="identifier" /><property name="Name" type="string" purpose="name"/><property name="Email" type="string" purpose="email"/></properties></entity>%%$#@@!!!*&
AddInProviderId	Default AddInProviderId
LastUpdatedDateTime	Date time value

The above code creates an entry in SKEntityDataSourceProvider table. The entry in Repository DB can be verified in database by running the query select * from SKEntityDataSourceProvider Where DisplayName='Custom Provider'

Column Name	Value
ID	Auto generated
Application	Sample Repository
Entity ID	Copy the Id Column value of SKEntity of RoleEntity row here.
DisplayName	Custom Provider
DefinitionXML	<dataproviders> <provider name="customsqlprovider" type="Custom Sql Provider"><properties> <property name="UserId" type="guid" purpose="identifier"/> <property name="Name" type="string" purpose="name"/> <property name="Email" type="string" purpose="email"/> <property name="Password" type="string" purpose="password"/> </properties> <datasource>server=localhost;database=SkeltaUsers;uid=sa;pwd=XXX</datasource><SourceTable>SqlUsers</SourceTable><loginproperty>Email</loginproperty><

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Column Name	Value
	/provider></dataproviders>
Provider	customsqlprovider
Instance Name	sqlproviderCustom
LastUpdatedDateTime	Date time value

Verify whether the Custom User Entity Data Source Provider is configured correctly or not

Verification:

The above code updates the entries required for configuring Custom User Entity Data Source Provider. Now, we should be able to see the provider in Repository list page. The previous entry of "List" will be updated with "Custom Provider".

If you get a message "No Provider Available" in the repository list page, cross check the Logger Console for a detailed exception for the cause.

Check the calls made to the custom user entity data source provider.

If there is no exception then we could also create the Actor object API to check the calls made to the custom user entity data source provider.

DLL REFERENCE

Skelta.HWS

Workflow.NET.NET2

The parameters passed to the Actor object are application object & real user id. Real user id can be formed by prefixing provider instance name and special characters "::" to the unique identifier value of user. We have used UserId column value for Kevin.

	UserId	Name	Role	Email	Password
	e9d21049-e3cf-4f4d-9d51-036d0145c547	Michelle	Admin	michelle@skelta....	michelle
	c6a7170a-6a35-4f79-8688-150d0ac8fe15	Kevin	Developer	kevin@skelta.com	kevin
	ce24b3d0-21dc-4821-8d55-afe4c8581716	John	Developer	john@skelta.com	john
	9d3eebb3-ebbe-4415-9d18-df4205af5c46	Elaine	Product Manager	elaine@skelta.com	elaine

Sample code

```
Skelta.HWS.Actor actor = new Actor(new ApplicationObject("Sample Repository"),
"sqlproviderCustom::c6a7170a-6a35-4f79-8688-150d0ac8fe15");
object identifierValue = actor.Resource.Properties.Identifier.Value;
string Name = actor.Resource.Properties.Name.Value.ToString();
string Email = actor.Resource.Properties.Email.Value.ToString();
```

The above code also creates an entry to SKVirtualActor table, this can be checked using query. select * from SKVirtualActor Where Name='Kevin' .

Step 4: Create Default Admin for the Repository

The following example demonstrates the creation of default admin for the repository using API.

Create default admin for the repository:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

PROCEDURE

DLL REFERENCE

```
Skelta.CentralConfigCodeBehind.dll
Skelta.Forms.Core.dll
```

NAMESPACE USED:

```
Skelta.Forms.Core
Skelta.Forms.Core.Controls
Skelta.Repository.List
string repositoryName = "Sample Repository";
Skelta.Core.ApplicationObject applicationObject = new
Skelta.Core.ApplicationObject(repositoryName);
 //Create ListDefintion object
 Skelta.Repository.List.ListDefinition ld = new
Skelta.Repository.List.ListDefinition(applicationObject, "Security Group Mapping");
 //Create ListDatahandler object
 Skelta.Repository.List.ListDataHandler ldh = new
Skelta.Repository.List.ListDataHandler(repositoryName, "Security Group");
 Guid SGID = ldh.GetListItemId("Administrator"); //Get Administrator SecurityGroup Id
 bool IsalreadyMapped = false; //flag variable to check whether user already configured or
not.
 Skelta.Repository.Security.SecurityLevel _SecurityLevel =
Skelta.Repository.Security.SecurityLevel.Repository;
 string _ApplicationName = repositoryName;
//Using entity infra to get the admin user from entity data source provider.
Skelta.Entity.Entity userEntity = Skelta.Entity.Entity.GetEntity(applicationObject,
"UserEntity");
object[] adminRealId =
((Skelta.Entity.IEntityAdminDataSourceProvider)userEntity.DataBridge.DataBridgeProviders["s
qlproviderCustom"]).GetAdministrator();
//Change the provider user details
Skelta.HWS.Actor actr = new Skelta.HWS.Actor(applicationObject, "sqlproviderCustom::" +
adminRealId[0]);
string UserId = actr.Id.ToString();
string UserName = actr.Resource.Properties.Name.Value.ToString();
//Create SecurityRightsHandler Object
Skelta.Repository.Security.SecurityRightsHandler srh = new
Skelta.Repository.Security.SecurityRightsHandler(repositoryName);
Workflow.NET.Interfaces.IDataHandler dbhandler =
Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(ld.Configuration);
IDataParameter[] parametres = new IDataParameter[3];
IDataReader rdr = null;
using (dbhandler)
{
parametres[0] = dbhandler.GetParameter("@ApplicationName", _ApplicationName);
 parametres[1] = dbhandler.GetParameter("@SGId", SGID);
 parametres[2] = dbhandler.GetParameter("@UserId", new Guid(UserId));
 string strQuery = "Select * from " + ld.TableName + " Where Application=@ApplicationName
AND SecurityGroupId=@SGId AND UserId= @UserId";
 rdr = dbhandler.ExecuteReader(strQuery, parametres);
 if (rdr.Read())
 {
 if (rdr[0] != System.DBNull.Value)
IsalreadyMapped = true;
}
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (rdr != null)
rdr.Close();
if (!IsalreadyMapped) //If user not configured as admin
{
using (TransactionScope TScope = Workflow.NET.CommonFunctions.GetNewTransactionScope())
{
Skelta.Repository.List.ListItem li = new Skelta.Repository.List.ListItem(ld);
ListMainForm listTableForm =
((ListMainForm)li.ListForm.Records[0].FindControlByID("_sys_sgm_form"));
((ListTextDataItem)listTableForm.FindControlByID("_sys_sgm_sgtitle")).Value =
_SecurityLevel.ToString(); //"Administrator";
 ((ListGuidDataItem)listTableForm.FindControlByID("_sys_sgm_sgid")).Value = SGID;
 ((ListGuidDataItem)listTableForm.FindControlByID("_sys_sgm_userid")).Value = UserId;
 ((ListGuidDataItem)listTableForm.FindControlByID("_sys_sgm_roleid")).Value = Guid.Empty;
 ((ListIntDataItem)listTableForm.FindControlByID("_sys_sgm_priority")).Value = 0;
 ((ListGuidDataItem)listTableForm.FindControlByID("_sys_sgm_mappedlistId")).Value =
Guid.Empty;
 ((ListGuidDataItem)listTableForm.FindControlByID("_sys_sgm_mappedlistitemid")).Value =
Guid.Empty;
 ((ListGuidDataItem)listTableForm.FindControlByID("_sys_sgm_effective_security_id")).Value
= Guid.Empty;
 ((ListGuidDataItem)listTableForm.FindControlByID("_sys_sgm_security_customization_id")).Va
lue = Guid.Empty;
li.Save(ListItemVersionStatus.Published);
TScope.Complete();
}
Skelta.CentralConfigCodeBehind.CommonFunctions commonFunctions = new
Skelta.CentralConfigCodeBehind.CommonFunctions();
commonFunctions.UpdateRespositoryPropetyCollection(repositoryName);
dbhandler = DataHandlerFactory.GetDataHandler(ld.Configuration);
using (dbhandler)
{
IDataParameter[] param = new IDataParameter[2];
param[0] = dbhandler.GetParameter("@UserId", new Guid(UserId));
param[1] = dbhandler.GetParameter("@Application", (DbType)System.Data.SqlDbType.VarChar,
50, ParameterDirection.Input, repositoryName);
dbhandler.ExecuteProcedure("USP_Update_UserId", param);
}
 }
```

Verification:

The above code created an Admin user with complete rights for the repository. The above code will also do an entry in SKVirtualActor table. The code would execute entity data source provider methods to get the required user information. Use this query to check the user entry in virtual actor table. select * from SKVirtualActor Where Name='Michelle'. The user named Michelle can now login to the EnterpriseConsole site using the email Id (since loginproperty for provider is set as Email) and check the menu.

- **Additional API to validate the API Entries**:

API to validate if entry already exists for Data Source, Repository and AddInProvider is listed below.

```
/// <summary>
/// To check if the data source to be created already exists
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// </summary>
/// <param name="DataSourceName">Name of the data source to be checked</param>
/// <returns></returns>
public static bool IsDataSourceExists(string DataSourceName)
{
Log SkeltaLog = new Log();
try
{
DataSourceCollection dsc = new DataSourceCollection();
if (dsc.Contains(DataSourceName))
{
return true;
}
else
{
SkeltaLog.LogInformation("Skelta on " + System.Net.Dns.GetHostName() + " does not contain "
+ DataSourceName + ".  Unable to find datasource.");
return false;
}
}
catch (Exception exception)
{
SkeltaLog.LogError(exception, "Error while getting repository details");
return false;
}
}
/// <summary>
/// To check if the repository exists.  Refer Skelta Log for Exceptions if any
/// </summary>
/// <param name="Repository"></param>
/// <returns>True if exists else false.  false even on exceptions</returns>
public static bool IsRepositoryExist(string Repository)
{
Log SkeltaLog = new Log();
try
{
RepositoryCollection rc = new RepositoryCollection();
if (rc.Contains(Repository))
{
return true;
}
else
{
return false;
}
}
catch (Exception exception)
{
SkeltaLog.LogError(exception, "Error while getting repository details");
return false;
}
}
/// <summary>
/// To check if the Provider with the given name for the repository exists
/// </summary>
/// <param name="RepositoryName">Name of the repository for which the configuration needs
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
to be added</param>
/// <param name="AddInProviderName">Name of the Add In Provider to be checked</param>
/// <returns>True if exists else false</returns>
public static bool IsProviderExists(string RepositoryName, string AddInProviderName)
{
Log SkeltaLog = new Log();
try
{
ApplicationObject applicationObject = new ApplicationObject(RepositoryName);
AddInProviderCollection skAddInProviderCollection = new
AddInProviderCollection(applicationObject);
return skAddInProviderCollection.Contains(AddInProviderName);
}
catch (Exception exception)
{
SkeltaLog.LogError(exception, "Error while getting provider details");
throw;
}
}
```

Step 5: Create Custom Role Entity Data Source Provider

The custom role entity provider fetches the role information from the same table "SqlUsers". The column Role in this table contains role information for a user.

The below configuration is mandatory for the Role Provider.

1. The EntityIdentifierType method should always return string as the user entity provider value type is always a string (Ex: "<Provider Instance Name>::<User Unique Identifier>").
2. The instance for User entity data source provider & Role entity data source provider should be same. The role provider does not exist by itself and is always mapped to a user provider. The users returned by the role needs to be mapped to a user provider to be used within AVEVA Work Tasks. Due to this behavior it's mandatory to have the providers having same instance name. For this sample the instance name used is "sqlproviderCustom".
3. The identifier type for Role provider is always string as the Role name itself is considered as unique identifier.

PROCEDURE

DLL REFERENCE

```
Workflow.NET.NET2 .dll
 Skelta.Repository.Web.dll
System.Transactions.dll
```

NAMESPACE USED

```
System
System.Collections.Generic
System.Text
System.Xml
System.Xml.XPath
System.Transactions
System.Data
Workflow.NET
Workflow.NET.Interfaces
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Skelta.Entity
System.Collections.Specialized
Workflow.NET.Storage
Skelta.Entity.RoleEntityProvider
System.IO
```

API Configuration

```
The Role Entity Data Source Provider should:
Inherit Skelta.Entity.DataSourceProviderBase class
```

Or

```
Implement Skelta.Entity.IEntityDataSourceProvider, Skelta.Core.ISkeltaAddInProvider
Implement interface Skelta.Entity.RoleEntityProvider.IRoleEntityDataSourceProvider
```

The only method which is accessed from IEntityDataSourceProvider for Role Provider is GetEntityItem (object entityId). Since there is no specific UI for role provider the other interface methods will be used for future release.

Skelta.Entity.RoleEntityProvider.IRoleEntityDataSourceProvider: The methods in this interface provide information related to Roles & Users related to each role.

Sample code

```
/// <summary>
/// Role entityt data source  provider.
/// </summary>
class RoleEntityDataSourceProvider : DataSourceProviderBase,
Skelta.Entity.RoleEntityProvider.IRoleEntityDataSourceProvider
{
  bool _IsConnectionValid = false;
  string _ConnectionString;
  string _ApplicationName;
  private Log logger = new Log();
#region IRoleEntityDataSourceProvider Members
  /// <summary>
  /// Repository  name.
  /// </summary>
  public string ApplicationName
{
  get
{
  return this._ApplicationName;
}
  set
{
  this._ApplicationName = value;
}
}
  /// <summary>
  /// Connection string for role provider
  /// </summary>
  public string ConnectionString
{
  get
{
  return this._ConnectionString;
}
  set
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
  this._ConnectionString = value;
}
}
  /// <summary>
  /// The identifier type value  should always be string. It's always string as the type
for user entity provider will always be string.
  /// </summary>
  public string EntityIdentifierType
{
 get
{
 return "string";
}
}
  /// <summary>
  /// Return all the roles for the  provider, Get called from Role look up provider (Queue,
Security)
  /// </summary>
  /// <returns>Collection of roles as string array</returns>
  public string[] GetAllRoles()
{
  // When search is performed  on role lookup , the context  information  of search can be
got using ther object
Skelta.Repository.Web.Lookup.RoleExecutionContextInfo searchContextInfo
= Skelta.Repository.Web.Lookup.RoleExecutionContext.GetRoleExecutionContext();
  string[] StrRolesArray = { };
StringCollection StrRoleCollection = new StringCollection();
System.Data.IDataReader idrRoles;
  try
{
  if (_IsConnectionValid)
{
  // Modify the query as per your table details
  string StrSQl = "";
  if (searchContextInfo == null)
StrSQl = "SELECT DISTINCT Role FROM  SqlUsers WHERE Role is not NULL ORDER BY Role";
  else
StrSQl = "SELECT DISTINCT Role FROM  SqlUsers WHERE Role is not NULL AND Role like '%" +
searchContextInfo.RoleSearchString + "%' ORDER BY Role";
IDataHandler dbHandler = DataHandlerFactory.GetDataHandler(this.ConnectionString, "sql
server");
  using (dbHandler)
{
idrRoles = dbHandler.ExecuteReader(StrSQl);
  while (idrRoles.Read())
{
StrRoleCollection.Add(idrRoles.GetString(0));
}
idrRoles.Close();
StrRolesArray = new string[StrRoleCollection.Count];
StrRoleCollection.CopyTo(StrRolesArray, 0);
}
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
  catch (Exception ex)
{
logger.LogError(ex, "@@Error while getting the SkeltaRoleProvider GetAllRoles");
  throw;
}
  return StrRolesArray;
}
  /// <summary>
  /// Get the roles associated to a user. Called from Queue or security for getting the
users associated to a role  after role selection
  /// </summary>
  /// <param name="resource">resource </param>
  /// <returns>Arrays of Roles</returns>
  public string[] GetRolesForUser(Resource resource)
{
StringCollection StrRoleForUserCollection = new StringCollection();
System.Data.IDataReader idrRolesForUser;
  string[] StrRolesForUserArray = { };
  string StrSql = "";
StrSql = "  SELECT Role";
StrSql += " FROM SqlUsers ";
StrSql += " WHERE  UserId =@UserId";
  try
{
  if (_IsConnectionValid)
{
IDataHandler dbHandler = DataHandlerFactory.GetDataHandler(this.ConnectionString, "sql
server");
  using (dbHandler)
{
System.Data.IDataParameter iparmUserId = dbHandler.GetParameter("@UserId",
resource.Properties.Identifier.Value);
idrRolesForUser = dbHandler.ExecuteReader(StrSql, iparmUserId);
  while (idrRolesForUser.Read())
{
StrRoleForUserCollection.Add(idrRolesForUser.GetString(0));
}
idrRolesForUser.Close();
}
StrRolesForUserArray = new string[StrRoleForUserCollection.Count];
StrRoleForUserCollection.CopyTo(StrRolesForUserArray, 0);
}
}
  catch (Exception ex)
{
logger.LogError(ex, "@@SkeltaRoleProvider@GetRolesForUser");
  throw;
}
  return StrRolesForUserArray;
}
  /// <summary>
  /// Collection  of  unique users id's which  belongs to the Role. The Keys for the
colleciton hold the unique identifier of the user & the value  could be name or any  custom
value.
  /// </summary>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
  /// <param name="roleName"></param>
  /// <returns></returns>
  public Dictionary<object, string> GetUsersInRole(string roleName)
{
  string StrSql = string.Empty;
Dictionary<object, string> StrUsersInRoleArray = new Dictionary<object, string>();
System.Data.IDataReader idrUsersInRole;
  try
{
  if (_IsConnectionValid)
{
StrSql = "  SELECT UserId , Name";
StrSql += " FROM SqlUsers ";
StrSql += " WHERE Role=@Role";
IDataHandler dbHandler = DataHandlerFactory.GetDataHandler(this.ConnectionString, "sql
server");
  using (dbHandler)
{
System.Data.IDataParameter iparmUserId = dbHandler.GetParameter("@Role", roleName);
idrUsersInRole = dbHandler.ExecuteReader(StrSql, iparmUserId);
  while (idrUsersInRole.Read())
{
StrUsersInRoleArray.Add(idrUsersInRole.GetGuid(0), idrUsersInRole.GetString(1));
}
idrUsersInRole.Close();
}
}
}
  catch (Exception ex)
{
logger.LogError(ex, "@@SkeltaRoleProvider@GetUsersInRole");
  throw;
}
  return StrUsersInRoleArray;
}
  /// <summary>
  /// Checks if the user belongs to a specific role.
  /// </summary>
  /// <param name="resource">User object.</param>
  /// <param name="roleName">Role name.</param>
  /// <returns></returns>
  public bool IsUserInRole(Resource resource, string roleName)
{
  string StrSql = string.Empty;
  bool IsUserInRole = false;
System.Data.IDataReader idrUserInRole;
  try
{
  if (_IsConnectionValid)
{
StrSql = " SELECT * FROM SqlUsers";
StrSql += " WHERE  UserId = @UserId AND Role = @Role";
IDataHandler dbHandler = DataHandlerFactory.GetDataHandler(this.ConnectionString, "sql
server");
  using (dbHandler)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
System.Data.IDataParameter iparmRolesAppid = dbHandler.GetParameter("@UserId",
resource.Properties.Identifier.Value);
System.Data.IDataParameter iparmUserId = dbHandler.GetParameter("@Role", roleName);
idrUserInRole = dbHandler.ExecuteReader(StrSql, iparmRolesAppid, iparmUserId);
  while (idrUserInRole.Read())
{
IsUserInRole = true;
  break;
}
idrUserInRole.Close();
}
}
}
  catch (Exception ex)
{
logger.LogError(ex, "@@SkeltaRoleProvider@IsUserInRole");
IsUserInRole = true;
}
  return IsUserInRole;
}
  void IRoleEntityDataSourceProvider.Initialize(string definition)
{
  if (!_IsConnectionValid)
Initialize(definition);
}
  /// <summary>
  /// Checks if a role exists in the provider.
  /// </summary>
  /// <param name="roleName">Role name.</param>
  /// <returns></returns>
  public bool RoleExists(string roleName)
{
  bool CheckRoleStatus = false;
  try
{
  if (_IsConnectionValid)
{
  string StrRoleName = roleName;
  string StrSQl = "SELECT distinct role FROM SqlUsers WHERE Role =@Role";
IDataHandler dbHandler = DataHandlerFactory.GetDataHandler(this.ConnectionString, "sql
server");
System.Data.IDataReader idrRoles;
  if (!string.IsNullOrEmpty(StrRoleName))
{
  using (dbHandler)
{
System.Data.IDataParameter iparmRole = dbHandler.GetParameter("@Role", roleName);
idrRoles = dbHandler.ExecuteReader(StrSQl, iparmRole);
  while (idrRoles.Read())
{
CheckRoleStatus = true;
}
idrRoles.Close();
}
}
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
  catch
{
CheckRoleStatus = true;
}
  return CheckRoleStatus;
}
#endregion
#region "DataSourceProviderBase"
  /// <summary>
  /// Loads the configuration info
  /// </summary>
  /// <param name="definitionXml"></param>
  public override void Initialize(string definitionXml)
{
  try
{
XmlDocument xDoc;
xDoc = new XmlDocument();
  try
{
xDoc.Load(new StringReader(definitionXml));
}
  catch (System.Exception ex)
{
logger.LogError(ex, "Could not open the document ()");
  throw ex;
}
  //Get connection string
  try
{
XmlNode xnc = xDoc.GetElementsByTagName("datasource").Item(0);
_ConnectionString = xnc.InnerText;
_IsConnectionValid = true;
}
  catch (System.Exception ex)
{
logger.LogError(ex, "Cannot get required information.");
  throw ex;
}
}
  catch (Exception ex)
{
logger.LogError(ex, "@@SkeltaMembershipDataProvider@Initialize");
  throw ex;
}
}
  /// <summary>
  /// GetEntityItem is the only interface method of IEntityDataSourceProvider which will be
used. This method call is initiated  after Role is selected from UI and the virtual id for
the role gets created.
  /// This method does not get called while listing different roles for the provider.
  /// </summary>
  /// <param name="entityId">Id of the EntityItem</param>
  /// <returns></returns>
  public override EntityItem GetEntityItem(object entityId)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
  string Identifier = EntityItemProperties.Identifier.PropertyName;
  //It's role provider, so return single property value
EntityItemPropertiesCollection tempResProperties = EntityItemProperties.CreateDeepClone();
EntityItem entityItem = new EntityItem();
  //We need to consider role itself as unique identifier.
tempResProperties[Identifier].Values.Add(entityId);
  if (!string.IsNullOrEmpty(ProviderPrefix))
entityId = entityId.ToString().Replace(ProviderPrefix, "");
tempResProperties["RoleName"].Values.Add(entityId);
entityItem.Properties = tempResProperties;
  return entityItem;
  //It's mandatory for the role provider to perisist the RoleName in db, if  a property
named RoleName is not
  //specfified or coded to be perissted in DB then the below code can be used to add or set
the same
  //if (!tempResProperties.ContainsKey("RoleName"))
  //{
  //  EntityItemProperty roleProperty = new EntityItemProperty();
  //  roleProperty.PropertyName = "RoleName";
  //  roleProperty.PropertyDisplayName = "Role";
  //  roleProperty.PersistColumn = "RoleName";
  //  roleProperty.Persist = true;
  //  tempResProperties.Add(roleProperty.PropertyName, roleProperty);
  //}
  //else
  //{
  //  tempResProperties["RoleName"].PersistColumn = "RoleName";
  //  tempResProperties["RoleName"].Persist = true;
  //}
}
  public override IEntityDataSourceProvider Clone()
{
  //Create a shallow clone of the instance
RoleEntityDataSourceProvider roleProvider = new RoleEntityDataSourceProvider();
roleProvider = this;
  return roleProvider;
}
#endregion
}
```

- Configure Custom Role Entity Data Source Provider

Configuring custom role entity data source provider involves entries in two tables of the Repository

- Table : SKAddInProviders , Object : Skelta.Core.AddInProvider
- Table : SKEntityDataSourceProvider, Object : Skelta.Entity.DataSourceProvider

I. Adding Assembly information in SKAddInProvider table

```
API for Adding Role provider in SKAddInProvider table: Create an entry
in SKAddInProviders for the role provider.
ApplicationObject applicationObject = new ApplicationObject("Sample Repository");
 Skelta.Core.AddInProvider addInProvider = new
Skelta.Core.AddInProvider(applicationObject);
 addInProvider.Type = "Role";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 addInProvider.Name = "customsqlroleprovider";
#warning : Make sure to change the assembly  path
 addInProvider.Description = "Sample Custom Role Entity Data Source Provider";
 addInProvider.ClassName =
"CustomUserEntityDataSourceProvider.RoleEntityDataSourceProvider";
 addInProvider.AssemblyNameWithAbsoluteOrRelativePath = @"M:\Michelle\
CustomUserEntityDataSourceProvider\CustomUserEntityDataSourceProvider\bin\Debug\
CustomUserEntityDataSourceProvider.dll";
 addInProvider.IsGacAssembly = false;
 addInProvider.IsGlobal = true;
 addInProvider.IsDefault = true; // The properties IsGacAssembly, IsGlobal, IsDefault  is
not currently in use. The value used in sample should be used "as is"
 addInProvider.Save();
```

Database script for adding Role provider in SKAddInProviders table

```
INSERT INTO SKAddInProviders VALUES (newid(),'Role','customsqlroleprovider','Sample Custom
Role Entity Data Source
Provider','CustomUserEntityDataSourceProvider.RoleEntityDataSourceProvider','M:\Michelle\
CustomUserEntityDataSourceProvider\CustomUserEntityDataSourceProvider\bin\Debug\
CustomUserEntityDataSourceProvider.dll',null,0,1,1,getutcdate())
```

Note: Assembly path should be changed.

Verification:

The above code creates an entry in SKAddInProviders table with type as Role. The entry in Repository DB can be verified in database by running the query select * from SKAddInProviders Where Name='customsqlroleprovider '

Column Name	Value
ID	Auto Generated
Type	Role
Name	customsqlroleprovider
Description	Sample Custom Role Entity Data Source
ClassName	CustomUserEntityDataSourceProvider. RoleEntityDataSourceProvider
Assembly	M:\Michelle\ CustomUserEntityDataSourceProvider\ CustomUserEntityDataSourceProvider\ bin\Debug \CustomUserEntityDataSourceProvider.dll
Settings	Null
IsGacAssembly	False
Isdefault	True
IsGlobal	True

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Column Name	Value
LastUpdatedDateTime	Date time value

II. Creating an Entry in SKEntityDataSourceProvider table

API for updating the List Role Provider with Custom Role Provider.

Update the role entry created for the list with custom role provider in SKEntityDataSourceProvider table.

```
Skelta.Entity.Entity entity = Skelta.Entity.Entity.GetEntity(applicationObject,
"RoleEntity");
Skelta.Entity.DataSourceProvider roleEntitySourceProvider = null;
roleEntitySourceProvider = new Skelta.Entity.DataSourceProvider(applicationObject,
"skeltalist");
 //Role entity definition
 string  roleProviderDefinitionXML = "";
 roleProviderDefinitionXML += "<dataproviders>";
 roleProviderDefinitionXML += "<provider name=\"customsqlprovider\" type=\"Custom Role
Provider\">";
 roleProviderDefinitionXML += "<properties>";
 roleProviderDefinitionXML += "<property name=\"Role\" type=\"string\"
purpose=\"identifier\"/>";
 roleProviderDefinitionXML += "<property name=\"RoleName\" type=\"string\" persist=\"true\"
persistcolumn=\"rolename\"/>";
 roleProviderDefinitionXML += "</properties>";
roleProviderDefinitionXML += "<datasource>server=localhost;database=SkeltaUsers;User
Id=sa;pwd=XXXX;</datasource>";
roleProviderDefinitionXML += "</provider>";
 roleProviderDefinitionXML += "</dataproviders>";
roleEntitySourceProvider.DefinitionXml = roleProviderDefinitionXML;
//Change the role provider name.
roleEntitySourceProvider.Provider = "customsqlroleprovider";
roleEntitySourceProvider.InstanceName = "sqlproviderCustom"; // Instance Name should be
same as the instance name set for the user provider.
roleEntitySourceProvider.DisplayName = "Custom Role Provider";
// adding an entry in SKEntityDataSourceProvider fro Role provider
roleEntitySourceProvider.Entity = entity;
roleEntitySourceProvider.Save();
```

Database script for updating the Skelta list Role Provider with Custom Role Provider.

Make sure to update the connection string information in the definition XML with appropriate value.
The provider column value should be same as the name column value inserted in SKAddInProviders table.

Note: Also, ensure that the query given below updates the row for the List Role Entity Data Source Provider and not the User Entity Data Source Provider.

//For Windows Authentication

```
Update SKEntityDataSourceProvider
Set
DefinitionXML =' <dataproviders><provider name="customsqlprovider" type="Custom Role
Provider"><properties><property name="Role" type="string" purpose="identifier"/><property
name="RoleName" type="string" persist="true" persistcolumn="rolename"/></properties>
<datasource>server=localhost; database=SkeltaUsers;Integrated
Security=true;</datasource></provider></dataproviders>',
DisplayName='Custom Role Provider',Provider='customsqlroleprovider',
InstanceName='sqlproviderCustom' Where InstanceName='skeltalist' and EntityId=(Select Id
from SKEntity where Name='RoleEntity')
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

//SQL SERVER Authentication

```
Update SKEntityDataSourceProvider
Set
DefinitionXML =' <dataproviders><provider name="customsqlprovider" type="Custom Role
Provider"><properties><property name="Role" type="string" purpose="identifier"/><property
name="RoleName" type="string" persist="true" persistcolumn="rolename"/></properties>
<datasource>server=localhost; database=SkeltaUsers;User
 Id=sa;pwd=XXXX;</datasource></provider></dataproviders>',
DisplayName='Custom Role Provider',Provider='customsqlroleprovider',
InstanceName='sqlproviderCustom' Where InstanceName='skeltalist' and EntityId=(Select Id
from SKEntity where Name='RoleEntity')
```

Note: Connection string should be changed.

The provider name value customsqlroleprovider should be same as the AddInProvider Name column value.

Verification:

The above code creates an entry in SKEntityDataSourceProvider table. The entry in Repository DB can be verified in database by running the query select * from SKEntityDataSourceProvider Where DisplayName='Custom Role Provider

Column Name	Value
ID	Auto generated
Application	Sample Repository
Entity ID	Copy the Id Column value of SKEntity of RoleEntity row here.
DisplayName	Custom Role Provider
DefinitionXML	<dataproviders><provider name="customsqlprovider" type="Custom Role Provider"><properties><property name="Role" type="string" purpose="identifier"/><property name="RoleName" type="string" persist="true" persistcolumn="rolename"/></properties> <datasource>server=localhost; database=SkeltaUsers;User Id=sa;pwd=XXXX;</datasource></provider></dataproviders>
Provider	customsqlroleprovider
Instance Name	sqlproviderCustom
LastUpdatedDateTime	Date Time Value

Verification:

Now we should be able to access role & respective user information from the role provider. Sample code to check role information is listed below.

```
AddInProviderCollection addInProviderCollection = new
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
AddInProviderCollection(applicationObject);
Dictionary<string, ISkeltaAddInProvider> addinProviders =
addInProviderCollection.GetProvidersForType(true, "Role");
ISkeltaRoleProvider iSkeltaRoleProvider;
//Addin provider added with  name as RoleProvider is the entity  provider and should be
 used to access the role entity data source provider.
 iSkeltaRoleProvider = (ISkeltaRoleProvider)addinProviders["RoleProvider"];
 iSkeltaRoleProvider.ApplicationName = "Sample Repository";
 string[] rolesCollection = iSkeltaRoleProvider.GetAllRoles();
 Dictionary<object,string> usersInRole =
iSkeltaRoleProvider.GetUsersInRole(rolesCollection[0]);
```

The above code displays all the roles configured.

```
Code below creates an entry in SkVirtualRole table.
//For each role, a virtual role id gets created which is used internally for reference
 with in AVEVA Work Tasks.
Skelta.Core.VirtualRole virtualRole = new VirtualRole(applicationObject);
 Guid virtualRoleId = virtualRole.GetVirtualRoleId(rolesCollection[0]);
The script below can be used to check the entries in respective tables.
select * from SKAddInProviders Where Name='customsqlprovider'
select * from SKEntityDataSourceProvider Where DisplayName='Custom Role Provider'
select * from SKVirtualRole where ResourceHandler='Sample Repository'
```

Custom Role Provider Fetching Roles from SQL and Users from Active Directory

Setup: Open a new .NET Class Library project, provide the namespace and class name for the project accordingly and add reference to "Workflow.NET.NET2.dll" from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder.

PROCEDURE

DLL REFERENCE

```
Workflow.NET.NET2
System.DirectoryServices
```

NAMESPACE USED

```
System
System.IO
System.Collections.Generic
System.Text
System.Collections
System.Collections.Specialized
System.Data
System.Xml
System.DirectoryServices
System.Security.Cryptography
System.Text.RegularExpressions
```

Sample Code

To support various type of Active Directory, you need to use appropriate APIs. Example, for RODC use ReadonlyServer authenticationType while creating instance of DirectoryEntry.

```
using Skelta.Core;
using Skelta.Entity;
using Skelta.Entity.RoleEntityProvider;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Collections.Specialized;
using System.DirectoryServices;
using System.IO;
using System.Xml;
using Workflow.NET;
using Workflow.NET.Interfaces;
using Workflow.NET.Storage;
namespace CustomUserEntityProvider
{
public class Roles : DataSourceProviderBase, IRoleEntityDataSourceProvider,
ISkeltaAddInProvider
{
bool _IsConnectionValid = false;
string _ConnectionString;
string _SQLConnectionString;
string _LDAPConnectionString;
string _ApplicationName;
private string _UserID, _Password, sDataSourceType;
private bool _IsPasswordEncrypted = false;
private Workflow.NET.Log log = new Workflow.NET.Log();
private string _LDAPServer, _LDAPSearchBase, _LDAPBaseFilter;
private SearchScope _SearchScope;
string _ProviderPrefix = "";
private Config config = new Config();
private ReferralChasingOption _ReferralChasing;
private int _SizeLimit;
private TimeSpan _TimeOut;
private Log logger = new Log();
#region IRoleEntityDataSourceProvider Members
/// <summary>
/// Name of the Repository
/// </summary>
public string ApplicationName
{
get
{
return this._ApplicationName;
}
set
{
this._ApplicationName = value;
}
}
/// <summary>
/// Connection string
/// </summary>
public string ConnectionString
{
get
{
return this._ConnectionString;
}
set
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
this._ConnectionString = value;
}
}
/// <summary>
/// SQL Connection string
/// </summary>
public string SQLConnectionString
{
get
{
return this._SQLConnectionString;
}
set
{
this._SQLConnectionString = value;
}
}
/// <summary>
/// LDAP Connection string
/// </summary>
public string LDAPConnectionString
{
get
{
return this._LDAPConnectionString;
}
set
{
this._LDAPConnectionString = value;
}
}
/// <summary>
/// The identifier type value should always be string.
/// It's always string as the type for user entity provider will always be /// string.
/// </summary>
public string EntityIdentifierType
{
get
{
return "string";
}
}
/// <summary>
/// Getting the information like SizeLimit,Timeout,Referral Chasing from the /// definition
xml
/// </summary>
internal class XmlPropertyHandler
{
private XmlDocument xmlDoc;
/// <summary>
/// Loads the definition xml value to the xml document
/// </summary>
/// <param name="xmlString">Definition xml is passed as string</param>
public XmlPropertyHandler(string xmlString)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
xmlDoc = new XmlDocument();
xmlDoc.Load(new StringReader(xmlString));
}
/// <summary>
/// Getting the value of Searchscope[Subtree,base etc] from the xml document
/// </summary>
/// <returns></returns>
internal SearchScope GetSearchScope()
{
XmlNodeList xnl = xmlDoc.GetElementsByTagName("SearchScope");
if (xnl == null || xnl.Count == 0)
return SearchScope.Subtree; //Default Search Scope;
string SearchScopeString = xnl.Item(0).InnerText.ToUpperInvariant().Trim();
if (SearchScopeString == "BASE")
{
return SearchScope.Base;
}
else
{
if (SearchScopeString == "ONELEVEL")
return SearchScope.OneLevel;
else
return SearchScope.Subtree;
}
}
/// <summary>
/// Checks whether password is encrypted or not
/// </summary>
/// <returns></returns>
internal bool IsPasswordEncrypted()
{
XmlNodeList xnl = xmlDoc.GetElementsByTagName("AuthenticationPassword");
if (xnl == null || xnl.Count == 0)
return false; //Default Settings;
if (xnl.Item(0).Attributes["IsEncrypted"] == null)
{
return false;
}
string IsPasswordEncryptedString =
xnl.Item(0).Attributes["IsEncrypted"].Value.ToUpperInvariant().Trim();
if (IsPasswordEncryptedString == "TRUE")
{
return true;
}
else
{
return false;
}
}
//Getting the sizelimit from the xml document
internal int GetSizeLimit()
{
XmlNodeList xnl = xmlDoc.GetElementsByTagName("SizeLimit");
if (xnl == null || xnl.Count == 0)
return 0;
string PageSizeString = xnl.Item(0).InnerText.ToUpperInvariant().Trim();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
int PageSize = 0;
try
{
PageSize = int.Parse(PageSizeString);
}
catch { }
return PageSize;
}
//Getting the referral chasing value from the xml document
internal ReferralChasingOption GetReferralChasing()
{
XmlNodeList xnl = xmlDoc.GetElementsByTagName("ReferralChasing");
if (xnl == null || xnl.Count == 0)
return ReferralChasingOption.External; //Default Referral Chasing Value;
string RCString = xnl.Item(0).InnerText.ToUpperInvariant().Trim();
if (RCString == "NONE")
{
return ReferralChasingOption.None;
}
else
{
if (RCString == "ALL")
return ReferralChasingOption.All;
else
{
if (RCString == "SUBORDINATE")
return ReferralChasingOption.Subordinate;
else
return ReferralChasingOption.External;
}
}
}
//Getting Timeout value from the xml document
internal TimeSpan GetTimeOut()
{
XmlNodeList xnl = xmlDoc.GetElementsByTagName("TimeOut");
if (xnl == null || xnl.Count == 0)
return TimeSpan.FromSeconds(-1.0);
string TimeOutString = xnl.Item(0).InnerText.ToUpperInvariant().Trim();
double Seconds;
try
{
Seconds = double.Parse(TimeOutString);
}
catch
{
Seconds = -1.0;
}
return TimeSpan.FromSeconds(Seconds);
}
//Getting LDAPServer value from the xml document
internal string GetLDAPServer()
{
XmlNodeList xnl = xmlDoc.GetElementsByTagName("LDAPServer");
if (xnl == null || xnl.Count == 0)
return "LDAP://";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
string s = xnl.Item(0).InnerText.Trim();
if (!s.EndsWith("/"))
{
s += "/";
}
return s;
}
//Getting LDAPSearchBase value from the xml document
internal string GetLDAPSearchBase()
{
XmlNodeList xnl = xmlDoc.GetElementsByTagName("LDAPSearchBase");
if (xnl == null || xnl.Count == 0)
return "";
return xnl.Item(0).InnerText.Trim();
}
//Getting Domain name from the xml document
internal string GetDomainName()
{
XmlNodeList xnl = xmlDoc.GetElementsByTagName("DomainName");
if (xnl == null || xnl.Count == 0)
return "";
return xnl.Item(0).InnerText.Trim();
}
//Getting Base filter value from the xml document
internal string GetBaseFilter()
{
XmlNodeList xnl = xmlDoc.GetElementsByTagName("LDAPBaseFilter");
if (xnl == null || xnl.Count == 0)
return "";
return xnl.Item(0).InnerText.Trim();
}
}
/// <summary>
/// Getting all the roles from sql table
/// </summary>
/// <returns> string i.e. all the roles from the table</returns>
public string[] GetAllRoles()
{
// When search is performed on role lookup , the context informaiton of search can be got
using ther object
Skelta.Repository.Web.Lookup.RoleExecutionContextInfo searchContextInfo =
Skelta.Repository.Web.Lookup.RoleExecutionContext.GetRoleExecutionContext();
string[] StrRolesArray = { };
StringCollection StrRoleCollection = new StringCollection();
System.Data.IDataReader idrRoles;
try
{
if (_IsConnectionValid)
{
// Modify the query as per your table details
string StrSQl = "";
if (searchContextInfo == null)
StrSQl = "SELECT DISTINCT Role FROM users WHERE Role is not NULL ORDER BY Role";
else
StrSQl = "SELECT DISTINCT Role FROM users WHERE Role is not NULL AND Role like '%" +
searchContextInfo.RoleSearchString + "%' ORDER BY Role";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using (IDataHandler dbHandler = DataHandlerFactory.GetDataHandler(this.SQLConnectionString,
"sql server"))
{
idrRoles = dbHandler.ExecuteReader(StrSQl);
while (idrRoles.Read())
{
int il1 = StrRoleCollection.Add(idrRoles.GetString(0));
}
idrRoles.Close();
StrRolesArray = new string[StrRoleCollection.Count];
StrRoleCollection.CopyTo(StrRolesArray, 0);
}
}
}
catch (Exception ex)
{
logger.LogError(ex, "@@Error while getting the SkeltaRoleProvider GetAllRoles");
}
return StrRolesArray;
}
//Getting DirectoryEntry object
private DirectoryEntry GetDirectoryEntry(string Path)
{
// To access an RODC environment appropriate flag need to be set or use alternate active
directory API's. As an example
AuthenticationTypes authenticationTypesForSearch = AuthenticationTypes.Secure |
AuthenticationTypes.ReadonlyServer | AuthenticationTypes.ServerBind;
DirectoryEntry de = new DirectoryEntry(Path,null,null, authenticationTypesForSearch);
return de;
}
/// <summary>
/// Collection of unique users id's which belongs to the Role.
/// The Keys for the colleciton hold the unique identifier of the user & the value could be
name or any custom value.
/// </summary>
/// <param name="resource"></param>
/// <returns></returns>
public Dictionary<object, string> GetUsersInRole(string roleName)
{
Dictionary<object, string> StrUsersInRoleArray = new Dictionary<object, string>();
try
{
using (DirectoryEntry entry = GetDirectoryEntry(_LDAPConnectionString))
{
using (DirectorySearcher deSearch = new DirectorySearcher())
{
deSearch.SearchRoot = entry;
deSearch.Filter = "(&(objectClass=group) (cn=" + roleName + "))";
SearchResultCollection results = deSearch.FindAll();
if (results.Count > 0)
{
try
{
using (DirectoryEntry group = GetDirectoryEntry(results[0].Path))
{
object members = group.Invoke("Members", null);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
foreach (object member in (IEnumerable)members)
{
try
{
using (DirectoryEntry x = new DirectoryEntry(member))
{
string user = x.Properties["sAMAccountName"].Value.ToString();
StrUsersInRoleArray.Add(user, user);
}
}
catch (Exception e)
{
throw e;
}
}
}
}
catch (Exception e)
{
throw e;
}
}
}
}
}
catch (Exception ex)
{
throw new Workflow.NET.ActiveDirectoryException(Skelta.Core.DS.SeverityLevel.Information,
"Error while retrieving users from role (" + roleName + ")" + ex.Message, new
Skelta.Core.ApplicationObject(RepositoryName));
}
return StrUsersInRoleArray;
}
/// <summary>
/// Get the roles associated to a user.
/// Called from Queue or security for getting the users associated to a role after role
selection
/// </summary>
/// <param name="resource"></param>
/// <returns></returns>
public string[] GetRolesForUser(Workflow.NET.Resource resource)
{
StringCollection StrRoleForUserCollection = new StringCollection();
string[] StrRolesForUserArray = { };
try
{
using (DirectoryEntry entry = GetDirectoryEntry(_LDAPConnectionString))
{
using (DirectorySearcher mySearcher = new DirectorySearcher(entry))
{
mySearcher.Filter = "(&(objectClass=user) (cn=" + ((resource.Properties.Name.Value != null
&& resource.Properties.Name.Value.ToString() != "") ?
resource.Properties.Name.Value.ToString().ToLowerInvariant() :
resource.Properties.Identifier.Value.ToString().ToLowerInvariant()) + "))";
mySearcher.PropertiesToLoad.Add("memberOf");
int propertyCount;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
SearchResult myresult = mySearcher.FindOne();
propertyCount = myresult.Properties["memberOf"].Count;
string dn;
int equalsIndex = 0;
for (int i = 0; i <= propertyCount - 1; i++)
{
dn = (string)myresult.Properties["memberOf"][i];
string strRoles = dn.Trim("CN=".ToCharArray()).Split(',')[0];
if (equalsIndex == -1)
{
return null;
}
StrRoleForUserCollection.Add(strRoles);
}
}
}
StrRolesForUserArray = new string[StrRoleForUserCollection.Count];
StrRoleForUserCollection.CopyTo(StrRolesForUserArray, 0);
}
catch (Exception ex)
{
if (ex.GetType() == typeof(System.NullReferenceException))
{
throw new Workflow.NET.ActiveDirectoryException(Skelta.Core.DS.SeverityLevel.Information,
"User does not have a role assigned", new Skelta.Core.ApplicationObject(RepositoryName));
}
else
{
throw new Workflow.NET.ActiveDirectoryException(Skelta.Core.DS.SeverityLevel.Information,
"Error while retrieving roles for the user" + ex.Message, new
Skelta.Core.ApplicationObject(RepositoryName));
}
}
return StrRolesForUserArray;
}
/// <summary>
/// Checks if the user belongs to a specific role.
/// </summary>
/// <param name="resource">User object</param>
/// <param name="roleName">Role name</param>
/// <returns></returns>
public bool IsUserInRole(Workflow.NET.Resource resource, string roleName)
{
bool isGroupMember = false;
try
{
using (DirectoryEntry entry = new DirectoryEntry(_LDAPConnectionString))
{
DirectorySearcher deSearch = new DirectorySearcher();
deSearch.SearchRoot = entry;
deSearch.Filter = "(&(objectClass=group) (cn=" + roleName + "))";
SearchResultCollection results = deSearch.FindAll();
if (results.Count > 0)
{
DirectoryEntry group = null;
try
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
group = GetDirectoryEntry(results[0].Path);
object members = group.Invoke("Members", null);
foreach (object member in (IEnumerable)members)
{
DirectoryEntry x = null;
try
{
x = new DirectoryEntry(member);
if
(x.Properties[EntityItemProperties.Identifier.PropertyName].Value.ToString().ToLowerInvaria
nt() == resource.Properties.Identifier.Value.ToString().ToLowerInvariant())
{
isGroupMember = true;
break;
}
}
catch (Exception e)
{
throw e;
}
finally
{
if (x != null)
{
x.Close();
x.Dispose();
}
}
}
}
catch (Exception e)
{
throw e;
}
}
}
}
catch (Exception ex)
{
log.LogInformation("Error while checking user from role (" + roleName + ") for domain(" +
_ConnectionString + ")" + ex.Message);
}
return isGroupMember;
}
/// <summary>
/// Verify whether role exists in the sql provider
/// </summary>
/// <param name="roleName">role name </param>
/// <returns></returns>
public bool RoleExists(string roleName)
{
bool CheckRoleStatus = false;
if (_IsConnectionValid)
{
string StrRoleName = roleName;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
string StrSQl = "SELECT distinct role FROM users WHERE role =@role";
IDataHandler dbHandler = DataHandlerFactory.GetDataHandler(this._SQLConnectionString, "sql
server");
System.Data.IDataReader idrRoles;
if (!string.IsNullOrEmpty(StrRoleName))
{
using (dbHandler)
{
System.Data.IDataParameter iparmRole = dbHandler.GetParameter("@role", roleName);
idrRoles = dbHandler.ExecuteReader(StrSQl, iparmRole);
while (idrRoles.Read())
{
CheckRoleStatus = true;
}
idrRoles.Close();
}
}
}
return CheckRoleStatus;
}
#endregion
#region ISkeltaAddInProvider Members
Guid _Id = Guid.Empty;
string _Settings = "";
Guid ISkeltaAddInProvider.Id
{
get { return _Id; }
}
//Initializing the provider
void ISkeltaAddInProvider.InitializeProvider(string settings, Guid id)
{
_Id = id;
_Settings = settings;
}
string ISkeltaAddInProvider.Settings
{
get { return _Settings; }
}
#endregion
#region "DataSourceProviderBase"
/// <summary>
/// Initializing the preferences [ReferralChasing,SearchScope,SizeLimit, LDAPBaseFilter] of
Active directory
/// </summary>
/// <param name="definitionXml"></param>
public override void Initialize(string definitionXml)
{
try
{
XmlDocument xDoc;
xDoc = new XmlDocument();
try
{
xDoc.Load(new StringReader(definitionXml));
}
catch (System.Exception ex)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
logger.LogError(ex, "Could not open the document ()");
throw ex;
}
//Get connection string
try
{
XmlNode xnc = xDoc.GetElementsByTagName("datasource").Item(0);
_SQLConnectionString = xnc.InnerText;
sDataSourceType = "sql server";
_IsConnectionValid = true;
if (string.IsNullOrEmpty(_LDAPServer))
{
XmlPropertyHandler xmlHandler = new XmlPropertyHandler(definitionXml);
_SearchScope = xmlHandler.GetSearchScope();
_ReferralChasing = xmlHandler.GetReferralChasing();
_SizeLimit = xmlHandler.GetSizeLimit();
_TimeOut = xmlHandler.GetTimeOut();
//_UserID = xmlHandler.GetUserID();
// _Password = xmlHandler.GetPassword();
_LDAPBaseFilter = xmlHandler.GetBaseFilter();
_LDAPSearchBase = xmlHandler.GetLDAPSearchBase();
_LDAPServer = xmlHandler.GetLDAPServer();
_ProviderPrefix = xmlHandler.GetDomainName();
if (!string.IsNullOrEmpty(_LDAPSearchBase)
&& string.IsNullOrEmpty(_ProviderPrefix))
_ProviderPrefix = _LDAPSearchBase.Split(',')[0].Split('=')[1] + "\\";
_IsPasswordEncrypted = xmlHandler.IsPasswordEncrypted();
_LDAPConnectionString = _LDAPServer + _LDAPSearchBase;
}
}
catch (System.Exception ex)
{
logger.LogError(ex, "Cannot get required information.");
throw ex;
}
}
catch (Exception ex)
{
logger.LogError(ex, "@@SkeltaMembershipDataProvider@Initialize");
throw ex;
}
}
internal delegate string GetDistiguisedNameFromCommonNameHandler(string CN);
internal class QueryToSearchFilterConversion
{
Hashtable Expressions = new Hashtable();
Hashtable Brackets = new Hashtable();
ArrayList BracketsArray = new ArrayList();
public string Output = "";
private Hashtable ParamHashTable = new Hashtable();
}
public override IEntityDataSourceProvider Clone()
{
//Clone the instance
Roles roleProvider = new Roles();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
roleProvider = this;
return roleProvider;
}
/// <summary>
/// GetEntityItem is the only interface method of IEntityDataSourceProvider which will be
used. This method call is initiated after Role is selected from UI and the virtual id for
the role gets created.
/// This method does not get called while listing different roles for the provider.
/// </summary>
/// <param name="entityItemId"></param>
/// <returns></returns>
public override EntityItem GetEntityItem(object entityItemId)
{
string Identifier = EntityItemProperties.Identifier.PropertyName;
//It's role provider, so return single property value
EntityItemPropertiesCollection tempResProperties = EntityItemProperties.CreateDeepClone();
EntityItem entityItem = new EntityItem();
//We need to consider role itself as unique identifier.
tempResProperties[Identifier].Values.Add(entityItemId);
if (!string.IsNullOrEmpty(ProviderPrefix))
entityItemId = entityItemId.ToString().Replace(ProviderPrefix, "");
//It's mandatory for the role provider to perisist the RoleName in db, if a property named
RoleName is not
//specfified or coded to be perissted in DB then the below code can be used to add or set
the same
if (!tempResProperties.ContainsKey("RoleName"))
{
EntityItemProperty roleProperty = new EntityItemProperty();
roleProperty.PropertyName = "RoleName";
roleProperty.PropertyDisplayName = "Role";
roleProperty.PersistColumn = "RoleName";
roleProperty.Persist = true;
tempResProperties.Add(roleProperty.PropertyName, roleProperty);
}
else
{
tempResProperties["RoleName"].PersistColumn = "RoleName";
tempResProperties["RoleName"].Persist = true;
}
tempResProperties["RoleName"].Values.Add(entityItemId);
entityItem.Properties = tempResProperties;
return entityItem;
}
#endregion
}
}
```

Table Configuration:

Roles Database:

The roles database used in this example, has a single table called Users, which uses a column "Role" to assign/ identify roles for the users.

Configuring custom roles involves entries in three tables of Skelta Repository database, namely:

1. SKAddInProviders
2. SKEntity

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. SKEntityDataSourceProvider

The order and the detailed steps of configuration is listed below:

In the SKAddInProvider insert a new record with the following field values:

Column Name	Value
ID	Auto Generated
Type	Roles
Name	Name for the Custom roles
Description	Any description / Null
ClassName	Name of the class implementing
Assembly	Path of the DLL built.
IsGacAssembly	False
Isdefault	False
LastUpdatedDateTime	Date time value

Example

```
Insert into SKAddInProviders
(Type,Name,Description,ClassName,Assembly,IsGacAssembly,Isdefault,LastUpdatedDateTime)VALUE
S('Role','NewRoles','try mapping','NewRoles.Roles','bin\NewRoles.dll','False','False','5/
12/2008 7:01:01 AM')
```

1. Open the SKEntity table. No entries are required. Copy the Entity table "Id" column value of the RoleEntity row , for the entry to be made in SKEntityDataSourceProvider table.
2. Open the SKEntityDataSourceProvider table, you will have a row with the default user provider "activedirectory". Now insert a new entry for our custom roles as defined.

Column	Value
ID	Auto generated
Application	Name of the Repository
Entity ID	Copy the Id Column value of SKEntity of RoleEntity row here.
DisplayName	Forms

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Column	Value
DefinitionXML	The definition xml should have details related to both sql provider and Active directory. for example, in the following definition xml the data source <dataproviders><provider name="sqlprovider" type="sql server"><properties><property name="role" type="string" displayname="Role" purpose="identifier"/></properties><datasource>Data Source=SKELTADTP38;Initial Catalog=DS3;Integrated Security=True</datasource><Preferences><SearchScope>SubTree</SearchScope><ReferralChasing>External</ReferralChasing><TimeOut>60</TimeOut><SizeLimit>0</SizeLimit><AuthenticationUser></AuthenticationUser><AuthenticationPassword></AuthenticationPassword></Preferences><LDAPServer>LDAP://</LDAPServer><LDAPSearchBase>DC=skelta,DC=dom</LDAPSearchBase><LDAPBaseFilter>(objectCategory=Person)</LDAPBaseFilter><DomainName>SKELTA\</DomainName></provider></dataproviders>
Provider	column must have the same name given in the "Name" column of SKAddInProviders table.
Instance Name	activedirectory

Example:

```
insert into
SKEntityDataSourceProvider(Application,EntityID,DisplayName,DefinitionXML,Provider,Instance
Name)
VALUES('ADSQLRep','69b60890-c24d-4ae5-8f03-cdd709a50d17','Forms','<dataproviders><provider
name="sqlprovider" type="sql server"><properties><property name="role" type="string"
displayname="Role" purpose="identifier"/></properties><datasource>Data
Source=SKELTADTP38;Initial Catalog=DS3;Integrated
Security=True</datasource><Preferences><SearchScope>SubTree</SearchScope><ReferralChasing>E
xternal</ReferralChasing><TimeOut>60</TimeOut><SizeLimit>0</SizeLimit><AuthenticationUser><
/AuthenticationUser><AuthenticationPassword
></AuthenticationPassword></Preferences><LDAPServer>LDAP://</LDAPServer><LDAPSearchBase>DC=
skelta,DC=dom</LDAPSearchBase><LDAPBaseFilter>(objectCategory=Person)</LDAPBaseFilter><Doma
inName>SKELTA\</DomainName></provider></dataproviders>','NewRoles','activedirectory').
```

- In the **SKEntityDatasourceProvider** table, every repository will have two entries. One with the display name "user provider name" for example, "activedirectory" and the other with the "user provider role" for example, "Active Directory role". Delete the row corresponding to the display name " Active Directory role". This is done to avoid the roles to be listed in the Enterprise Console from the active directory.

The Classes and Interfaces that need to be used to implement Custom Roles in AVEVA Work Tasks are:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. The Custom Roles Class has to be derived over "DataSourceProviderBase" part of Skelta.Entity namespace.
2. The Interface to be implemented by the Custom roles are:
   a. **IRoleEntityDataSourceProvider** – [Skelta.Entity.RoleEntityProvider]
   b. **ISkeltaAddInProvider** – [Skelta.Core]

Key Methods and Properties:

I**RoleEntityDataSourceProvider** Members

1. **public string IRoleEntityDataSourceProvider.EntityIdentifierType() -** This property returns the "identifier" type for the resources, which is "string" by default.
2. **void IRoleEntityDataSourceProvider.Initialize()** - This method is used to initialize the XML definition of the roles entity.

ISkeltaAddInProvider Members

1. **Guid ISkeltaAddInProvider.Id -** This property returns the AddInProviderId of the default role provider.
2. **void ISkeltaAddInProvider.InitializeProvider(string settings, Guid id) -** This method is used to initialize the provider.

DataSourceProviderBase Class Members

1. **public override void Initialize(string definitionXml)** - This method reads the XML definition.
2. **public override EntityItem GetEntityItem(object entityId)** - This returns the entity type, which is role and returns single property value.

After defining the methods and the properties as per the custom roles definition being used, build the project that generates <custom roles name>. DLL. This DLL needs to be now copied to the AVEVA Work Tasks bin folder.

Troubleshooting

You will get the "Object reference not set to an instance of an object" error when you try to open the Enterprise Console. One of the reason for this could be the access to the assembly path [that you have mentioned in the SKAddInProviders tables] is denied. To avoid this:

1. Copy the dll (NewRoles.dll) which you have created and put it in the folder where you have full access.
2. In SKAddInProviders table's Assembly column, update the assembly path.

Example:

1. If your dll is in C:\, then in the Assembly column of the SKAddInProviders table, update the url as C:\NewRoles.dll.
2. If your dll is in D:\Sample, then in the Assembly column of the SKAddInProviders table, update the url as D:\Sample\NewRoles.dll.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Associate Workflow to Scheduler Event

Overview

In business environment, based on a schedule, users might require triggering workflow daily at a particular time or particular day or particular month or last day of the month to launch a certain task to perform repeatedly. This can be achieved by associating Scheduler event provided by AVEVA Work Tasks in the Association page.

Example: A business process involved in the leave balance update. Assume that, this process update happens periodically on 25th of every month at 10 AM. This can be scheduled using the schedule workflow setup.

To schedule a workflow

1. Select **Workflows** from the **Enterprise Console** pull-down menu.
2. Select the workflow to be scheduled, and then click the **Associate** icon on the ribbon bar, or Create Associations in the context menu. The Association page appears.
3. Click **Scheduler** on the ribbon bar to schedule the workflow. The Scheduler association page appears.
4. Set **Start Date**, **End Date**, **Schedule Task**, and **Start Time**. You can schedule the task on a daily basis by using the Schedule Task Daily or schedule the task for reoccurrence using **Repeat Task** check box.
5. Select the option **Last day of the month(s)** to schedule the task on the last day of the selected month(s).
6. Provide **Version** of the workflow in **Workflow Details**. See Setting up a Schedule for more details.
7. Click **Attach Event** to schedule the workflow.

Set up a Schedule

The Scheduler has different options such as Start Date, End Date, Schedule Task, Start Time, Select Week Day, Monthly, Select Month, Last day of month(s), Repeat Task, and Workflow Details.

Note: No validation occurs when saving the association if the start time is earlier than the current time, requiring the user to manually enter a future date before saving.

Start Date

This option is for setting up the start date of period in which the scheduled workflow should be run.

Example: If you set Start Date as '01-04-2009' then the scheduled workflow will start getting triggered from '01-04-2009'.

End Date

This option is for setting up the end date of period in which the scheduled workflow should be stop running.

Example: If you set the End Date as '31-03-2010' then the scheduled workflow will stop running on '31-03-2010'.

Schedule Task

This option is for setting up a schedule for Daily/Weekly/Monthly basis. Workflow execution will be based on this setup also option to set the start time using the Start Time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Daily

Set the schedule on a daily basis. On selecting Daily from the Schedule Task drop-down, the Schedule Task Daily option will be displayed. Here, specify the number of days in the Every text box to execute the task on a specified day.

Example: Every day, every 2 days or once every 5th day.

Weekly

This option is for scheduling the task on weekly basis. On select of Weekly from Schedule Task drop-down, the Schedule Task Weekly will be displayed. Here specify the week in the Every text box to execute the task on specified week. You can schedule the task on particular day of the specified week. You can select the day by clicking Select WeekDay button.

Example: Every 1st week, every 3rd week, every Wednesday of 2nd week or All days of 4th week.

Select WeekDay

On click of Select WeekDay the Select Week Day window is displayed. Here you can select day by checking the specific day or days. You can select all days by checking All option.

Monthly

This option is for scheduling the task on monthly basis. On select of **Monthly** from **Schedule Task** drop-down, the **Schedule Task Monthly** will be displayed. Here specify the month in the Every text box to execute the task on specified month. You can schedule the task on particular day of the specified month. You can select the month by clicking **Select Month**.

Select Month

On click of Select Month the Select Months window is displayed. Here you can select month by checking the specific month or months. You can select all month by checking All option.

Last day of month(s)

This option is for scheduling the task on the last day of the selected months.

Example: If you select February month, the task will be scheduled on 28th of February or 29th of February (if leap year). Similarly, if you select January month, the task will be scheduled on 31st of January.

Repeat Task

Repeat Task option is for setting up reoccurrences. Schedule can set for repeat on minute basis or hourly basis. Repeat can be such as every 1 minute/every 1 hour.

Workflow Details

Workflow details selection is for selecting a particular version of the workflow. If not selected the latest version will be considered.

Edit Scheduler Association

You can change the scheduled execution time of the workflow or any event using **Edit Scheduler Association** option.

To edit scheduler association

1. Select **Workflows** from the **Enterprise Console** pull-down menu.
2. Right-click the workflow whose schedule is to be changed, and then select **Edit Scheduler Association**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

option in the context menu. The Edit Scheduler Association page appears.

3. Set **Start Date**, **End Date**, **Schedule Task**, and **Start Time**. You can schedule the task on a daily basis by using the Schedule Task Daily or schedule the task for re-occurrence using **Repeat Task** check box.

Note: It is not recommended to give 1 minute time interval for repeat task.

4. Select the option **Last day of the month(s)** to schedule the task on the last day of the selected month(s).
5. Provide **Version** of the workflow in **Workflow Details**. See Setting up a Schedule for more details.
6. Click **Update Event** to reschedule the workflow.

Example of Scheduling Tasks Using Scheduler Event

Consider a scenario where you have already created a workflow and want to execute the workflow every day on an hourly basis during 15th June 2009 to 30th June 2009.

To achieve the same, follow the given steps:

1. Log on to the **Enterprise Console**.
2. From the **Enterprise Console** pull-down menu, select **Workflows**.
3. Design and publish a workflow.
4. Select this workflow and then select **Associate** from the ribbon bar, or select **Create Association** from the context menu.
5. Click **Scheduler** to schedule workflow to run.
6. Select **Start Date**.
7. Select **End Date**.
8. Select **Daily** from **Schedule Task** drop-down list.
9. In the **Schedule Task Daily** section, enter '1' in the **Every** text box.
10. Check **Repeat Task**. In this section, enter '1' in the **Every** text box and Hours from drop-down list.
11. In the **Workflow Details** section, select Latest from the Version drop-down list.
12. Click **Attach Event**. A confirmation window appears.
13. Click **OK**.
14. To check whether workflow is triggered or not on specified duration, select **Inbox** from the **Enterprise Console** Menu button.

Manage Work Items

In AVEVA Work Tasks, you can manage work items by forwarding them to other users, or by aborting them.

Forward Work Items to Other Users

You can forward work items to other users.

Only user who have rights to forward a work item (**Can Forward Work Item** right should be checked) to other users will be able to view this menu item.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To forward work items to other users

1. Log on to the **Enterprise Console**.
2. From the **Enterprise Console** pull-down menu, select **Tools** > **Manage Work items**. The **Manage Work items** page appears. In this page all work items will be displayed in the grid. It also displays the information about work items which are assigned to different users within the application.
3. Select the required work item(s) to forward to different user, and click **Forward**. Alternatively, select **Forward** from the context menu. The **Manage Work item(s)** window appears.
4. Select the required user from the **User Lookup** window. The selected user will be populated.

   Note: Select only one user at a time to forward work item(s).
5. Click **Forward**. The selected work item gets assigned to the selected user and same will be updated in the grid.

Abort Work Items

User can also abort (kill) the work items in the Manage Work items. To kill the work items, the logged-in user should have the permission to kill the work items. For this, the Can Kill Work Item right should be checked.

To abort (kill) the work item, follow the given steps

1. Log on to the **Enterprise Console**.
2. From the **Enterprise Console** pull-down menu, select **Tools > Manage Work items**. The **Manage Work items** page appears.

   In this page all work items will be displayed in the grid. It also displays the information about work items which are assigned to different users within the application.
3. Select the required work item(s) to abort and then click **Abort** (top right corner), or select **Abort** from the context menu.

   Note: You can select multiple work items at a time by selecting check box in the grid.
4. On click of **Abort**, a confirmation message appears.
5. Click **Yes**.

Add New Tab and Custom Logic to Process Execution View

AVEVA Work Tasks allows users to create new tab and custom logic in the Process Execution View window.

To add a new tab and custom logic to the 'Process Execution View' pop-up window, follow the steps given below:

1. Open the following aspx page:
   BPMUITemplates\Default\ProcessExecutionView\ExecutionView.aspx
2. Make the modification which are marked in bold:

```
<radTS:RadTabStrip ID="RadTabStrip1" runat="server" Skin="SkeltaTabstrip"
CssClass="TabRadTabstrip"
OnClientTabSelected="ProcessTabClick" SelectedIndex="0">
<tabs>
<radTS:Tab ID="Tab1" runat="server" Text="Analysts View"  Value="tab3"  >
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
   </radTS:Tab>
   <radTS:Tab ID="Tab2" runat="server" Text="Progress view" Value="tab2">
   </radTS:Tab>
   <radTS:Tab ID="Tab3" runat="server" Text="Execution view" Value="tab1">
   </radTS:Tab>
   //Add the below line to create a new tab called Custom View in Execution view.
   <radTS:Tab ID="Tab4" runat="server" Text="Custom view" Value="tab4">
   </radTS:Tab>
   </tabs>
   </radTS:RadTabStrip>
   </div>

   <script>
    function ProcessTabClick(sender, eventArgs)
   {
   LoadReport(eventArgs.Tab.Value.substr(3,1));
   }

   function LoadReport(name)
   {
   _skLoadedReport = name;

   var uri="";
   switch(name)
   {
   //Case 4 is added to illustrate the example of adding a custom page to the  newly
   created tab
    //For the sake of example the custom page is also given as the analyst view. Change
   this file and point to the file in which custom logic is implemented
     case "4":
   uri="<%=HttpUtility.UrlEncode(_BaseControl.GetSecuredUriParameter(Workflow.NET.Report
   .WorkflowExecutionReportTypes.AnalystsView))%>";break;
   case "3":
   uri="<%=HttpUtility.UrlEncode(_BaseControl.GetSecuredUriParameter(Workflow.NET.Report
   .WorkflowExecutionReportTypes.AnalystsView))%>";break;
   case "2":
   uri="<%=HttpUtility.UrlEncode(_BaseControl.GetSecuredUriParameter(Workflow.NET.Report
   .WorkflowExecutionReportTypes.ProgressView))%>";break;
   case "1":
   uri="<%=HttpUtility.UrlEncode(_BaseControl.GetSecuredUriParameter(Workflow.NET.Report
   .WorkflowExecutionReportTypes.ExecutionView))%>";break;
}
```

3. You can see a **Custom View** tab in the Process Execution Window.
4. In the same way, you can add a new tab in the 'BPMUITemplates\Default\ProcessExecutionView\ ExecutionDetailsWindow.aspx' page (refer to the code given below) and custom logic in new url.

```
   <div id='tabstrip1' style="width: 100%;background-color:white;">
    <radTS:RadTabStrip ID="RadTabStrip1" runat="server"
    Skin="SkeltaTabstrip"  OnClientTabSelected="ProcessTabClick" SelectedIndex="0">
    <Tabs>
   <radTS:Tab runat="server" Text="Step Details" Value="step">
   </radTS:Tab>
   <radTS:Tab runat="server" Text="Variables" Value="variable">
   </radTS:Tab>
   <radTS:Tab runat="server" Text="Contents" Value="contents">
   </radTS:Tab>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<radTS:Tab runat="server" Text="Content" Value="content">
</radTS:Tab>
//Add the below line to create a new tab called Custom View in Execution Details
view.
//change should be made in the scripts to accept the new "value" created. For the
sake of example, now the value is given same as that of "variable"
 <radTS:Tab ID="Tab1" runat="server" Text="Custom View" Value="variable">
</radTS:Tab>

</Tabs>
</radTS:RadTabStrip>
</div>
```

After adding the above mentioned code to the aspx file, a **Custom View** tab pears in the Execution details window.

Customize the 'Step Details' Window

AVEVA Work Tasks allows you to customize the 'Step Details' window.

The following customization can be done in the 'Step Details' window:

1. Hide Step Details, Variables, Contents and Content tabs.
2. Localize the Description content.
3. Customize the look and feel of this window.

Steps to hide the Step Details, Variables, Content and Contents tabs:

1. Open the following aspx page from AVEVA Work Tasks installed folder. *\Web\BPMUITemplates\Default\ProcessExecutionView\ExecutionDetailsWindow.aspx*
2. Locate the following code and set the property **visible="false"** for the required tab to be hidden. For example, to hide **Step Details** tab, set **Visible** property as **false** for this tab.

```
<div id='tabstrip1' style="width: 100%;background-color:white;">
        <radTS:RadTabStrip ID="RadTabStrip1" runat="server"  Visible="false"
    Skin="SkeltaTabstrip"   OnClientTabSelected="ProcessTabClick" SelectedIndex="0">
            <Tabs>
                <radTS:Tab runat="server" visible="false" Text="Step Details" Value="step">
                </radTS:Tab>
                <radTS:Tab runat="server" Text="Variables" Value="variable">
                </radTS:Tab>
                <radTS:Tab runat="server" Text="Contents" Value="contents">
                </radTS:Tab>
                <radTS:Tab runat="server" Text="Content" Value="content">
                </radTS:Tab>

            </Tabs>
        </radTS:RadTabStrip>
    </div>
```

3. Control the page details. If you do not want to show tab details, then control it by editing the following methods.
   a. **ProcessTabClick** - This method will open the details of the clicked node.
   b. **ShowDetails** - This method will open the 'Step Details' window.
   c. **ShowVariables** - This method will open the 'Variables', 'Contents', 'Content' windows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
function ProcessTabClick(sender, eventArgs)
{
    if( eventArgs.Tab.Value=="step" ) ShowDetails();
    if( eventArgs.Tab.Value=="variable") ShowVariables("variables");
    if( eventArgs.Tab.Value=="contents") ShowVariables("contents");
    if( eventArgs.Tab.Value=="content") ShowVariables("content");
}

function ShowDetails()
{
    var l=window.location.toString();
     l=l.replace("executiondetailswindow.aspx","executiondetails.aspx");
     document.getElementById('framecontent').src = l;
}

function ShowVariables(displayMode)
{
    var l=window.location.toString();
     l=l.replace("executiondetailswindow.aspx","ShowVariablesAndContents.aspx");
     l+="&dm="+displayMode;
     document.getElementById('framecontent').src = l;

}
```

To Localize the description content, follow the steps given below:

1. Locate the *WorkflowElements\Default\en-US\Actions\XML* folder from AVEVA Work Tasks installed folder.
2. All the step details associated with any action will be from the respective Action definition from the Actions.xml file.

Note: Before modifying, take a backup of **Actions.xml** files. While modifying the **Actions.xml**, one should not modify the placeholders declared.

In order to modify the step description, you need to change the following description in the Actions.xml

```
- <action name="Start" hidden="true" helptemplate="StartActivity1.html" helptemplatewinheight="450" helptemplatewinwidth="600">
    <description>Workflow process starting point, Holds configuration for Global Variables,Local Variables,Content Expressions
      and Workflow synchronization mode.</description>
    <image resourcename="start-action.png" />
    <handler classname="ActionStartHandler" />
  - <visual-attributes>
      <shape>circle</shape>
      <backcolor>lightgreen</backcolor>
      <imagevisible>hidden</imagevisible>
    </visual-attributes>
  - <report rendertype="xsl" xsl="ActionDescription.xsl">
      <handler class="Workflow.NET.Report.ReportHandler" assembly="Bin\Workflow.NET.ReportHandler.NET2.dll" />
    - <messages>
      - <message id="ActionStarted">
        - <![CDATA[
                                        Start Action was initiated at <script
            language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.

          ]]>
        </message>
      </messages>
    </report>
  - <properties>
    - <property name="Variables" type="grid" helpstring="Variables declared at Workflow Level. Global variables can be used to
        get/set data across actions at workflow level. Local variables can be declared by prefixing '@' symbol to the variable
        name. Local variables can be used to get/set data at action level." helptemplate="Start-Variables.html"
        helptemplatewinheight="450" helptemplatewinwidth="600">
```

To customize the look and feel of the window, follow the steps given below:

Open the *BPMUITemplates\Default\Common\StyleSheet\Global.css* from AVEVA Work Tasks installed folder to customize the look and feel of the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For more details on the css class and the images, please go through the following files:

1. BPMUITemplates\Default\ProcessExecutionView\ShowVariablesAndContents.aspx
2. BPMUITemplates\Default\ProcessExecutionView\ ExecutionDetailsWindow.aspx

Customization

The following features can be customized in AVEVA Work Tasks:

- User interface
- Font
- Work item control

Customize User Interfaces

AVEVA Work Tasks now provides a new style sheet named **Global.css**, consisting of around 100 styles which are commonly used by most of the controls.

Other user interface customization options include customizing the **Farm Configuration Wizard**, and the **Enterprise Console Menu**.

Style Sheets

Overview

Earlier, most of the controls followed styles sheets of its own and sometimes even jumbled. Some controls had even multiple style sheets under their respective folders. Adding to that, some pages had even inline styles. So, changing the style was a Herculean tasks. Also, the changes might not reflect some times, because of the multiple existence of same class.

Now, customizing the UI styles as per your choice is no more a hectic task. Not more than 100 styles to be changed to give a look as per your desire.

AVEVA Work Tasks introduced a new style sheet named Global.css consisting around 100 styles is added and is commonly used by most of the controls. Around 40 style sheet files are removed which were used by individual controls. Same is applicable to Themes folders.

New common style sheet for all controls:

1. BPMUITemplates\Default\Common\StyleSheet\Global.css
2. The following style sheets are removed:
3. BPMUITemplates\Default\BAM\Stylesheet\global.css
4. BPMUITemplates\Default\Common\StyleSheet\common.css
5. BPMUITemplates\Default\Common\StyleSheet\DecisonsStyle.css
6. BPMUITemplates\Default\Common\StyleSheet\global1.css

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. BPMUITemplates\Default\Common\StyleSheet\NewStyleforWorkitemControl.css
8. BPMUITemplates\Default\Common\StyleSheet\report.css
9. BPMUITemplates\Default\Common\StyleSheet\SchemaDefinition.css
10. BPMUITemplates\Default\Common\StyleSheet\workflowdotnet.css
11. BPMUITemplates\Default\en-US\StyleSheet\calender.css
12. BPMUITemplates\Default\en-US\StyleSheet\common.css
13. BPMUITemplates\Default\en-US\StyleSheet\global.css
14. BPMUITemplates\Default\en-US\StyleSheet\workflowdotnet.css
15. BPMUITemplates\Default\Expression\Stylesheet\pd.css
16. BPMUITemplates\Default\Expression\Stylesheet\properties.css
17. BPMUITemplates\Default\Expression\Stylesheet\StyleDefault.css
18. BPMUITemplates\Default\HWS\Stylesheet\global.css
19. BPMUITemplates\Default\ProcessExecutionView\Stylesheet\common.css
20. BPMUITemplates\Default\ProcessExecutionView\Stylesheet\global.css
21. BPMUITemplates\Default\ProcessExecutionView\Stylesheet\properties.css
22. BPMUITemplates\Default\ProcessExecutionView\Stylesheet\varandcontents.css
23. BPMUITemplates\Default\ProcessExecutionView\Stylesheet\wer.css
24. BPMUITemplates\Default\ProcessExecutionView\Stylesheet\workflowdotnet.css
25. BPMUITemplates\Default\Repository\ListControl\Styles\Style.css
26. BPMUITemplates\Default\Repository\Site\styles\ApplicationStyles.css
27. BPMUITemplates\Default\Repository\Site\styles\template-wizard.css
28. BPMUITemplates\Default\Repository\Site\styles\aquila.css
29. BPMUITemplates\Default\SOA\SOA.css
30. BPMUITemplates\Default\StyleSheet\calender.css
31. BPMUITemplates\Default\StyleSheet\common.css
32. BPMUITemplates\Default\StyleSheet\global.css
33. BPMUITemplates\Default\StyleSheet\workflowdotnet.css
34. BPMUITemplates\Default\StyleSheet\global1.css

Exceptional case:

Though most of the customization happens by changing the Global.css, some controls like Process Designer control, Forms control still refers to some of the style sheets under respective folders. You need to change the styles in the style sheets listed below for complete customization.

1. BPMUITemplates\Default\ProcessDesigner\Menu\mainmenu.css
2. BPMUITemplates\Default\ProcessDesigner\Menu\menu.css
3. BPMUITemplates\Default\ProcessDesigner\Stylesheet\global.css
4. BPMUITemplates\Default\ProcessDesigner\Stylesheet\pd.css
5. BPMUITemplates\Default\ProcessDesigner\Stylesheet\properties.css
6. BPMUITemplates\Default\ProcessDesigner\Stylesheet\SchemaDefinition.css

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. BPMUITemplates\Default\ProcessDesigner\toolbar\Graphics\Graphics.css
8. BPMUITemplates\Default\SkeltaForms\StyleSheet\Designer.css
9. BPMUITemplates\Default\SkeltaForms\StyleSheet\menustyles.css
10. BPMUITemplates\Default\SkeltaForms\StyleSheet\SkeltaForm.css

Customize Farm Configuration Wizard User Interface

The background image and other images of Farm Configuration Wizard can be changed.

To do so, add the following keys in HKEY_LOCAL_MACHINE\SOFTWARE\SkeltaSilentInstallation\

- ConfigPopUpWizardImage – For setting popup background images in the Wizard.
- ConfigWizardSmallImage – For setting the background image of main screens but note that some of the screens have a smaller top bar like the Services screen.
- ConfigWizardImage – Background image of main screens like the Welcome screen.

Customize Enterprise Console Menu

Enterprise Console menu appears on the left hand side of the enterprise console website and is used to navigate through various AVEVA Work Tasks controls.

If the Repository is mapped to themes, then take the style sheet based on themes path. The themes path is as follows:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\Default\CommonUXControls\StyleSheet\UXControls.css

If the Repository is not mapped to themes, then take the style sheet from the following path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\CommonUXControls\StyleSheet\UXControls.css

- CSS Class Used for Main Menu Style is #Menu0.
- CSS Class Used for Title is #Menu0 .menuItem .title.
- CSS Class Used for Mouse hover on the Main Menu item is #Menu0 .menuItem:hover.
- CSS Class Used for Selected item on Main menu is #Menu0 .menuItem.selected

Style Change for Submenu

- CSS Class Used for submenu style is .SubMenu.
- CSS Class Used for submenu Title is .SubMenu .menuItem .title
- CSS Class Used for submenu Description is .SubMenu .menuItem .desc
- CSS Class used for submenu MenuItem Indicator is .SubMenu .menuItem .subMenuIndicator
- CSS Class used for submenu Menu Item hover and selected is .SubMenu .menuItem:hover, .SubMenu .menuItem.selected

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Customize Fonts

The primary font used in the Enterprise Console and Central Configuration sites is the Roboto font. This font is available in *BPMUITemplates/Default/Common/Fonts*.

Perform the following steps if you have customized the default stylesheet CSS file or added a custom stylesheet CSS file.

1. Include the font face for the Roboto font in the CSS files with corresponding relative path to the downloaded Roboto fonts folder.

   Ensure that the relative path refers to the folder in which the stylesheet CSS file is available. By default, the Roboto fonts folder is available in *BPMUITemplates/Default/Common/Fonts.*

   The following sample illustrates how to include the Roboto font in the customized stylesheet file.

   ```
   @font-face {
   font-family: "Roboto";
   src: url("../../../../Common/fonts/Roboto/Roboto-Regular.ttf") format("truetype");
   }
   @font-face {
   font-family: "Roboto";
   font-weight: bold;
   font-style: italic;
   src: url("../../../../Common/fonts/Roboto/Roboto-BoldItalic.ttf") format("truetype");
   }
   @font-face {
   font-family: "Roboto";
   font-style: italic;
   src: url("../../../../Common/fonts/Roboto/Roboto-italic.ttf") format("truetype");
   }
   @font-face {
   font-family: "Roboto";
   font-weight: bold;
   src: url("../../../../Common/fonts/Roboto/Roboto-Bold.ttf") format("truetype");
   }
   ```

2. Use the Roboto font in the stylesheet classes.

   For example: `font-family: "Roboto", "Segoe UI"`

Customize Work Item Control

To customize the Work Item Control, you need to navigate to the installed Path i.e.

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\web\BPMUITemplates\Default\WorkItemList\UserControls\WorkItemControl.ascx.

There are two sections in the Work Item Control:

1. Grid Left Menu. See Customizing Left Menu Grid for more details.
2. Grid Details View. See Customizing Grid Details View for more details.

To customize Control, do the following:

1. Go to *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Common\StyleSheet*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Open the Global.css file.
3. You can customize the UI by making change in the specified styles which are present in Global.css.

Customize Left Menu Grid

The Left Menu Grid basically lists the name of the workflow which has been triggered.

To customize the Left Menu Grid of the Work item Control, follow the steps given below:

1. Open *WorkItemControl.ascx* page from the *BPMUITemplates\Default\Workitemlist\UserControls* folder.
2. You can find the Style Sheet in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\BPMUITemplates\Default\Common\StyleSheet\Global.css*
3. Search for the Web Grid, where id="grdLeftMenu". [This is the Grid Left Menu ID].

To change background Color

1. You can change the background image of the Grid Left Menu by making changes in the **FrameStyle** tag of the Grid Left Menu.
2. Make the following changes in the attribute **BackgroundImage** of the **FrameStyle** tag by changing the existing image:

```
<FrameStyle  BorderWidth="0px" BorderStyle="Solid" BackgroundImage="<%$
ReplaceTemplateExpn:default/WorkItemList/Images/title-bgNew.png%>"  ></FrameStyle>
```

To change the selected row style

1. You can change the selected rowstyle of the Grid Left Menu by making change in the SelectedRowStyle tag of the Grid Left Menu.

```
<SelectedRowStyle  CssClass="GridSelectedRowStyle">
```

2. Search for **GridSelectedRowStyle** in Global.css file and then make the following changes:

```
.GridSelectedRowStyle {
border-width:1px;
font-family:"Segoe UI", "Trebuchet MS";
font-size:8pt;
border-style:solid;
background-color:Red;
border-color:#ACA899;
color:#363636;
}
```

Customize Grid Details View

To customize the Grid Details View, search for Web Grid where id="WebGrid1". It uses two stylesheets which are in the following path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Common\StyleSheet\Global.css

The following can be customized in the Grid Details View:

- Changing Header Style and Background Image
- Changing the color of the selected row style

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Changing the background color of the Frame
- Changing footer style of the Grid Details View

To change Header Style

To change the Header style of the Grid Details View, follow the steps given below:

- Open the Global.css file. You can find it at the following location:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Common\StyleSheet\Global.css

- To change the background image of the header, change the attribute BackgroundImage in the Css Class GridHeaderStyle tag.

```
<HeaderStyle Wrap="false" Overflow="hidden" OverflowX="hidden" OverflowY="hidden"
CssClass="GridHeaderStyle"></HeaderStyle>
.GridHeaderStyle
{
font-size:8pt;
font-family:"Segoe UI", "Trebuchet MS";
border-color:#e1e1e1;
border-style:solid;
color:#363636;
background-image:url(../images/grid-title-bg.png);
height:20px;
background-color:#e3efff;
border-width:1px;
}
```

- Header color can be altered by making change in the GridHeaderStyle class in the Global.css file.

```
GridHeaderStyle
{
font-size:8pt;
font-family:"Segoe UI", "Trebuchet MS";
border-color:#e1e1e1;
border-style:solid;
color:Red;
background-image:url(../images/grid-title-bg.png);
height:20px;
background-color:#e3efff;
border-width:1px;
}
```

Changing the color of the selected row style

- To alter the color of the selected row style, you need to make change in the SelectedRowStyle tag.
- Change the color attributes in the class GridSelectedRowStyle in the Global.css file.

```
<SelectedRowStyle CssClass="GridSelectedRowStyle"></SelectedRowStyle>
.GridSelectedRowStyle {
border-width:1px;
font-family:"Segoe UI", "Trebuchet MS";
font-size:8pt;
border-style:solid;
background-color:#e1eeff;
border-color:#ACA899;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
color:#363636;
}
```

Changing the background color of the Frame

- To change the background color of the frame, change the attribute of colors in the GridFrameStyle class in Global.Css file.

```
<FrameStyle GradientType="Vertical" CssClass="GridFrameStyle"></FrameStyle>
.GridFrameStyle{
font-size:8pt;
font-family:"Segoe UI", "Trebuchet MS";
color:#363636;
border-style:solid;
border-width:1px;
border-color:#9ac6ff;
background-color:#FFFFFF;
}
```

Changing the footer style of the Grid Details View

- We can change the color of the Status bar by changing the CSSClass GridStatusBarStyle.

```
.GridStatusBarStyle
{
font-size:8pt;
font-family:"Segoe UI", "Trebuchet MS";
color:#363636;
background-color:#e4e4e4;
border-width:1px;
border-style:solid;
border-color:#ACA899;
}
```

- Change the color attributes in the GridStatusBarStyle CSSClass in the Global.Css file..

```
<StatusBarStyle CssClass="GridStatusBarStyle">
<Padding Top="2px" Left="2px" Bottom="2px" Right="2px"></Padding>
</StatusBarStyle>
```

Change Height and Width of the Grid Details View

To modify the height and width of the Grid Details View:

Change **height** and width attribute of the **table** tag. For example, change height as 50%. This will reduce the height of the grid.

```
<td vAlign="top" id="GridDetailsTD">
<table height="50%" width="100%" class="TableFont" border="0"  cellpadding="0"
cellspacing="0">
```

Change Font Style Based on the Status of the Work Item

Font Style of the item can be changed in the Work Item list based on the different status or states. You need to make changes in the global.css style sheet.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- When the task is in **Normal** state that is if there is nothing to act on that activity then all those activities fonts are kept under .WLNormal tag. For example, if it is Information activity, change background color as red.
- Do the following change in the .WLNormal tag

```
.WLNormal
{
 background-color:Red;
 font-size:7pt;
 font-family:Segoe UI,Verdana;
 font-weight:normal;
      }
```

- You can change the background color, when the task is in **Awaiting** state. For example, when the task is in **Awaiting** state, change the background color as green.
- Do the following change in the .WLAwaiting tag:

```
.WLAwaiting
{
 font-size:7pt;
 font-family:Segoe UI,Verdana;
 font-weight:300;
 background-color:Green;
}
```

- You can change the background color, when the task is in Planned state. For example, when the task is in Planned state from queue, change the background color as aqua.
- Do the following change in the .WLPlannedQueueItem tag.

```
.WLPlannedQueueItem
{
 font-size:7pt;
 font-family:Segoe UI,Verdana;
 font-weight:normal;
 background-color:Aqua;
}
```

- You can change the background color, when the task is in **Hold** state. For example, when the task is in Hold state, change the background color to Blue.
- Do the following change in the .WLHold tag.

```
.WLHold
{
 font-size:7pt;
 font-family:Verdana;
 font-weight:normal;
 background-color:Blue;
}
```

Add or Remove Images

You can add or remove images in the toolbar. These images are present in:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\web\BPMUITemplates\Default\WorkItemList\toolbar\Graphics\img path.

To add a new image to the toolbar:

- Add the following line of code in LoadToolBar() method and based on the user requirement, call the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

javascript method to call the Custom page. For example, add a new Refresh image in the toolbar.

```
RadToolbarButton refreshButton = new RadToolbarButton();
refreshButton.CommandName = "Refresh";
refreshButton.ButtonImage = "refresh.png";
refreshButton.Attributes.Add("onclick", "javascript:Callrefresh('" + applicationName
+ "','" + Actor.Id + "');");
refreshButton.ToolTip = _WorkItemControl.GlobalResourceSet.GetString("Refresh");
refreshButton.ID = "tbrefreshbutton";
toolbar.Items.Add(refreshButton);
```

Remove an Image from the Toolbar

To remove an image from the toolbar:

- For example, if you want to remove the Show Queue image from the toolbar, remove the following line of code from the LoadToolBar() method:

```
RadToolbarButton showQueueButton = new RadToolbarButton();
showQueueButton.CommandName = "Show Queue";
showQueueButton.ButtonImage = "show-queue.png";
showQueueButton.Attributes.Add("onclick", "javascript:CallQueue('" + applicationName
+ "','" + Actor.Id + "');");
showQueueButton.ToolTip = _WorkItemControl.GlobalResourceSet.GetString("Show queue");
showQueueButton.ID = "tbshowqueuebutton";
toolbar.Items.Add(showQueueButton);
```

Customize User Name

To customize the user name appearing on the top-right corner, follow the steps given below:

1. If you want to change the caption work item list, search for the following line of code in the NewKeysBPM.txt locale file. This file is present in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Locales\en-US\NewKeysBPM.txt*.

```
<td align="right" width="80%"
class="LoginName"><%=_WorkItemControl.GlobalResourceSet.GetString("Workitem list
of")%> <%=CurrentLoggedUserName%>  </td>
```

2. Change the text Work Item list of to Welcome

```
<td align="right" width="80%"
class="LoginName"><%=_WorkItemControl.GlobalResourceSet.GetString("Welcome")%>
<%=CurrentLoggedUserName%>  </td>
```

Customize Document View

To enable or disable Take Ownership option from the Work Item detail view

1. To disable Take Ownership drop-down while viewing an item from the work item detail view, navigate to the path *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\web\BPMUITemplates\Default\WorkItemList\UserControls\ActivityUIDropDownControl.ascx*.
2. Search for asp:dropdownlist where ID="SelectCommonActionType" and then change Visible attribute to false

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

in the drop-down and also you can hide the Go button visibility.

```
<asp:DropDownList runat=server CssClass="select" ID="SelectCommonActionType"
Visible="false">
 </asp:DropDownList>
<input type="submit" Class="button2" name="ButtonGo" value=" <%=strbtnGo%> "
onclick="return Go_Submit()" style="visibility:hidden;"/>
```

To change the background color of the Comments section

1. To change the background color of any activity, navigate to the path *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Activities\Activity*.
   For example, Approval activity, select the ApprovalUI.htm file from the below path *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Activities\Activity\ApprovalUI.htm*
2. Change the bgcolor attribute of the TD tag to change the Comments section to the specified color.

   <TD bgcolor="#ececec" VALIGN="TOP"> // Change bgcolor="Red"

To change the Custom Default View

To change the custom default view, navigate to the CustomDocViewDefault.html page.

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Activities\Activity\CustomDocViewDefault.html and make change based on the requirements.

Themes

AVEVA Work Tasks supports **themes** that you can use to change the look and feel of the user interface. AVEVA Work Tasks provides **Default** and **Lite** themes. While creating a repository, ensure that the theme name specified in the **Theme Name** text box, in the **Manage Repository** page of the **Central Configuration**, is present in the following install path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\web\BPMUITemplates\Default\Themes

Apply Theme to the Core Product

You can apply themes to the core product by enabling the **Enable Theme** option available in the **Manage Repository** page of the **Central Configuration**.

To apply theme at repository level

1. Log on to the **Central Configuration**.
2. Click **Repository Settings**. The **Repository List** page appears.
3. Click **Add**. The **Manage Repository** page appears.
4. Enter the **Repository Name** and **Description**.
5. Check the **Enable Theme** check box. By default, the theme name is **Default**. If you want to apply **Lite** theme, then you can specify theme name as **Lite** in the **Theme Name** text box.
6. Click **OK**. The theme gets applied to the created repository.
7. To verify, log on to the **Enterprise Console**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:

- You can also change themes by clicking on the existing repository and then enable the themes option in the **Manage Repository** page.
- Ensure that the theme name entered or created is not present in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\XMLSchemas\MappedThemes.xml* file.
- Ensure that the theme name specified in the **Theme Name** text box is present in the following install path: *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\web\BPMUITemplates\Default\Themes*
- If a repository is created with old themes supported in 2017 releases, like blue, the theme is always redirected to the **Default** theme.

Themes in Work Item Control

Work Item Control has the following two sections:

- **Grid Left Menu**: Lists the name of the workflow which is been triggered.
- **Grid Details View**: Shows the details of that item.

Work Item Control uses the following two style sheets:

- If theme is not applied, then all the css paths are considered from the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Default\Common\StyleSheet\global.css*.
- If theme is applied, then selected then the css path from the theme folder directory structure are considered.

For example, if you select **Default** theme, then Work Item Control takes the css path from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\Default\Common\StyleSheet\global.css*.

If you select the theme as Lite, then Work Item Control takes the css from the path *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\ BPMUITemplates\Default\Themes\Lite\Common\StyleSheet\ global.css*.

Apply Themes in Grid Left Menu

Grid Left Menu

For Left Grid Style, the following CSS classes are used:

- LeftGridFrameStyle
- LeftGridRowStyle
- LeftGridSelectedRowStyle

Apply Themes in Grid Details View

Grid details view in Default and Lite Theme

For Right Grid Style, the following CSS classes are used:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- GridHeaderStyle
- GridFrameStyle
- GridSelectedRowStyle

Themes in Calendar Control

Calendar Control consists of the following three main grids:

The first grid lists all the calendars that the user has created. There is a *main Calendar and Child Calendar*. The *Child Calendar inherits* from the *main Calendar*.

Header Style

Header style of the main grid of Calendar control uses the stylesheet class **GridHeaderStyle**.

If theme is not applied, then it takes all the css path from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Common\StyleSheet\global.css*.

If the theme is applied, then it takes the css path from the theme folder directory structure.

For example, if you select the Default theme, then the Calendar Control takes the css path from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\Default\Common\StyleSheet\ global.css*.

Default Theme Image

```
<HeaderStyle CssClass="GridHeaderStyle" >
</HeaderStyle>
```

Frame Style

You can set the background image of the frame style of the main page grid by making change in the background image of the frame style.

Sample Code

```
<FrameStyle GradientType="Vertical" CssClass="GridFrameStyle" >
</FrameStyle>
```

Themes in Queue Control

Queue Control consists of four WebGrids. These WebGrids are present in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\HWS\QueueControl.ascx* page in the HWS folder.

The Queue Control refers to the following stylesheet:

If theme is not applied, then it take all the css path from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Common\StyleSheet\global.css*.

If theme is selected, then it takes the css path from the theme folder directory structure.

For example, if you select the **Default** theme, then Queue Control takes the css path from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\Default\Common\StyleSheet\ global.css*.

In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Queue Management**. The **Queue Management** window appears. The window displays the list of queues in a web grid with **Id = _WebGridQueEdit**.

The following parts of the queue that can be customized:

1. Header Style
2. Frame Style

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. RadTabStrip
4. Worklist layout
5. Legend Table

Header Style

Header style of the main grid of the Calendar Control uses the stylesheet class GridHeaderStyle.

Sample Code

```
<HeaderStyle CssClass="GridHeaderStyle">
</HeaderStyle>
```

Lite Theme Image

Frame Style

You can set the background of the frame style of the main page grid by making change in the background color attribute of the GridFrameStyle.

Sample Code

```
<FrameStyle GradientType="Vertical" CssClass="GridFrameStyle">
</FrameStyle>
```

RadTabStrip

The rad tab strip with ID RadTabStrip1 consists of the following four tabs:

1. Participants
2. Settings
3. Dispatch Pattern
4. Escalation

Worklist Layout

The Worklist layout WebGrid is available in the **Queue Preference** panel of the **Settings** tab. This page is available in the following path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\WorkItemList\UserControls\WorkListLayout.ascx

Themes in Process Designer Control

The menu bar in the Process Designer can be personalized using the following procedure:

Menu Bar

The **MenuPD** refers to the following path:

- **No theme is applied**: *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\telerik\RadControls\Menu\Skins\SkeltaMenuPD*
- **Default Theme**: *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\ Default\Themes\default\telerik\RadControls\Menu\Skins\SkeltaMenuPD*

Tool Bar

The **Toolbar** refers to the following path:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **No theme is applied**: *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\telerik\RadControls\Toolbar\Skins\SkeltaToolbar*
- **Default Theme**: *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\default\telerik\RadControls*

Footer

The Footer refers to the following path:

- **No theme is applied**: *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\ProcessDesigner\Graphics\PD-footer-bg.png*
- **Default Theme**: *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\ Default\Themes\default\ProcessDesigner\Graphics\PD-footer-bg.png*

The style sheet, Pd.css is available in the following paths:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\Default\ProcessDesigner\Stylesheet\pd.css

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\Default\Common\StyleSheet\global.css

Activities Title

The style sheet classes used are sktoolboxwindow and sk_toolbox_category.

Splitter

The style sheet classes used are pdtoolboxanchor.

Change Style of New Menu in Enterprise Console

If the Repository is mapped to themes, then take the style sheet based on themes path. The themes path is as follows:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\Default\CommonUXControls\StyleSheet\UXControls.css

- CSS Class Used for Main Menu Style is #Menu0.
- CSS Class Used for Title is #Menu0 .menuItem .title.
- CSS Class Used for Mouse hover on the Main Menu item is #Menu0 .menuItem:hover.
- CSS Class Used for Selected item on Main menu is #Menu0 .menuItem.selected

Style Change for Submenu

- CSS Class Used for submenu style is .SubMenu.
- CSS Class Used for submenu Title is .SubMenu .menuItem .title
- CSS Class Used for submenu Description is .SubMenu .menuItem .desc
- CSS Class used for submenu MenuItem Indicator is .SubMenu .menuItem .subMenuIndicator
- CSS Class used for submenu Menu Item hover and selected is .SubMenu .menuItem:hover, .SubMenu .menuItem.selected

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Change Style of Action Bar in Lists

If the Repository is mapped to themes, then take the style sheet based on themes path. The themes path is as follows:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\Default\CommonUXControls\StyleSheet\UXControls.css

- CSS Class Used for ActionBar style is .actionBar.
- CSS Class Used for ActionBar Menu Item are
  - actionBar .menuItem .content .name, actionBar .menuItem:hover .icon,
  - actionBar .menuItem:hover .name, actionBar .menuItem .content .icon
- CSS Class Used for ActionBar SubMenu Item is actionBar .subMenu, .actionBar .subSubMenu

Change Style of Window

If the Repository is mapped to themes, then take the style sheet based on themes path. The themes path is as follows:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\Default\telerik\RadControls\Window\Skins\SkeltaWindow\Window.css

- CSS Class Used for Window Title is RadWTitleRow

Change Style of Buttons

If the Repository is mapped to themes, then take the style sheet based on themes path. The themes path is as follows:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\Default\Common\StyleSheet\Global.css

CSS Class Used for button classes are:

```
inputbutton
button,[buttonsecondary-skbtn="submit"]
button,[buttoninputgo-skbtn="submit"]
inputsecondarybutton
input[type="button"]:disabled
input[type="button"]:disabled:hover
input[type="submit"]:disabled
input[type="submit"]:disabled:hover
input[type="button"]:hover
input[type="submit"]:hover
```

Content Handler

Content Handler is used to access application specific data inside a workflow. Content Handlers can be used at the repository level and also at the workflow level. Use this content variable to access application values in the custom table.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Use Content Handler

Content Handler is used to access application specific data inside a workflow. Content Handlers can be used at the repository level and also at the workflow level. Using this content variable we can access application values in the custom table.

We will see the steps to implement a content handler.

- Create a table in the database with the following script.

```
USE [DBName]
GO
/****** Object:  Table [dbo].[LeaveApplications]  Script Date: 03/29/2010 20:03:58
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[LeaveApplications](
 [LeaveApplicationID] [int] NOT NULL,
 [UserID] [int] NULL,
 [FromDate] [datetime] NULL,
 [ToDate] [datetime] NULL,
 [NoOfDays] [int] NULL,
 [LeaveType] [int] NULL,
 [Reason] [int] NULL,
 [ApplicationDate] [int] NULL,
 CONSTRAINT [PK_LeaveApplications] PRIMARY KEY CLUSTERED
(
 [LeaveApplicationID] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS  = ON, ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Add few rows of data for the created table.

- Steps to create **Repository level content Handler**

An entry in the SKAddInProviders table is required for the content handler. Execute the following query for the repository database.

There will be already an entry for Provider Name IContentHandler by default which we can update and use for our specific example. Updating the entry with condition, provider name = IcontentHandler is sufficient in this example. Change the database name, tablename and security settings as required in the below query.

```
update  SKAddInProviders
set [Settings]='<content><datasource type="sql
server">server=Localhost;database=TestRepo;Integrated
Security=true</datasource><tables><table name="LeaveApplications"><fields><field
name="LeaveApplicationID" type="int" canUpdate="false" identifier="true"
contentName="leaveapplicationid"/><field name="UserID" type="int" canUpdate="false"
contentName="userid" /><field name="FromDate" type="datetime" canUpdate="false"
contentName="fromdate"/><field name="ToDate" type="datetime" canUpdate="false"
contentName="todate"/><field name="NoOfDays" type="int" canUpdate="false"
contentName="noofdays"/><field name="LeaveType" type="int" canUpdate="false"
contentName="leavetype"/><field name="Reason" type="int" canUpdate="false"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
contentName="reason"/><field name="ApplicationDate" type="int" canUpdate="false"
contentName="applicationdate"/></fields><filter><![CDATA[LeaveApplicationID=<%#Variable.Lea
veApplicationId%>]]></filter></table></tables></content>',
IsGlobal='True'
where name='IContentHandler'
```

- Steps to create Repository level content Handler when the user's database is a part of the Repository.

An entry in the SKAddInProviders table is required for the content handler. Execute the following query for the repository database.There will be already an entry for Provider Name IContentHandler by default which we can update and use for our specific example. Updating the entry with condition, provider name = IcontentHandler is sufficient in this example. Since the database is a part of the Repository, the user need not use the previous code sample, but use the sample given below. In this sample you can see that instead of the connection string, the UseRepositoryConnectionString="yes" code is used.

```
update SKAddInProviders
set [Settings]='<content UseRepositoryConnectionString="yes"><tables><table
name="LeaveApplications"><fields><field name="LeaveApplicationID" type="int"
canUpdate="false" identifier="true" contentName="leaveapplicationid"/><field name="UserID"
type="int" canUpdate="false" contentName="userid" /><field name="FromDate" type="datetime"
canUpdate="false" contentName="fromdate"/><field name="ToDate" type="datetime"
canUpdate="false" contentName="todate"/><field name="NoOfDays" type="int" canUpdate="false"
contentName="noofdays"/><field name="LeaveType" type="int" canUpdate="false"
contentName="leavetype"/><field name="Reason" type="int" canUpdate="false"
contentName="reason"/><field name="ApplicationDate" type="int" canUpdate="false"
contentName="applicationdate"/></fields><filter><![CDATA[LeaveApplicationID=<%#Variable.Lea
veApplicationId%>]]></filter></table></tables></content>',
IsGlobal='True'
where Name='IContentHandler'
```

The defined entry will be inserted in the SKAddInProviders table after the successful execution of the above query.

- Isglobal property decides whether the content handler will be available for the whole repository or not. If set to true, the handler will be available for the whole repository. If the Content handler has to be made available only for a specific workflow, then set IsGlobal Property as False and make an entry in the SKAddInProviderSettings
- Steps to create WorkFlow level content Handler

This step has to be done only if the content handler is for a specific workflow. Delete the 'settings' column entry from the Content Handler and make it "Null"

Set the IsGlobal property as false in SKAddInProviders.

Make the following entry in the SKAddInProviderSettings table. Change the reponame, workflow name and the AddInProviderID(Id of the AddInProvider in the SKAddInProviders table) from the below query as required.

```
insert into SKAddInProviderSettings
(Application,Workflow,AddInProviderID,Settings,IsDefault)
values
('RepoName','WorkFlowName','a8edae68-420f-450d-8df2-6a5b31389207','<?xml version="1.0"
encoding="utf-8" ?><content><datasource type="sql
server">server=localhost;database=TestRepo;Integrated
Security=true</datasource><tables><table name="LeaveApplications"><fields><field
name="LeaveApplicationID" type="int" canUpdate="false" identifier="true"
contentName="leaveapplicationid"/><field name="UserID" type="int" canUpdate="false"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
contentName="userid" /><field name="FromDate" type="datetime" canUpdate="false"
contentName="fromdate"/><field name="ToDate" type="datetime" canUpdate="false"
contentName="todate"/><field name="NoOfDays" type="int" canUpdate="false"
contentName="noofdays"/><field name="LeaveType" type="int" canUpdate="false"
contentName="leavetype"/><field name="Reason" type="int" canUpdate="false"
contentName="reason"/><field name="ApplicationDate" type="int" canUpdate="false"
contentName="applicationdate"/></fields><filter><![CDATA[LeaveApplicationID=<%#Variable.Lea
veApplicationId%>]]></filter></table></tables></content>','True')
```

- Create a new workflow in the repository with a variable as the same name as of the FillterCondition set above (LeaveApplicationID in this example).

The whole XML file which can be used to import the below WF is given here. Import and make the necessary changes. Create another content variable which takes value from the client object which is calling the workflow.

//test is used here as we are using the following code to execute the workflow.

objClient.Execute(provider + "::" + virtualUserId, "<test>" + 10001 + "</test>");

10001 is one of the LeaveApplicationId in the table.

Use an update variable to set the value of the content variable to the Variable created.

Create a new Approval Activity on the "updated" link of the update activity with subject field as the from date in the Content Variable.

Publish and run the workflow using Client Object.Execute method. Pass the LeaveApplicationId along with it as shown below

objClient.Execute(provider + "::" + virtualUserId, "<test>" + 10001 + "</test>");

- Go to the Inbox and you should be able to find an Approval activity with "from date" of the leave application as subject.

The Content Handler has been successfully implemented for the default dll.

XML to Import WF which Implements Custom Content Handler

```
- <workflow name="" version="2">
- <diagramdatacollection>
- <diagramdata key="DesignerInstanceId">
- <![CDATA[
34eb47cb40164d33adcad5f358b6ded8
]]>
</diagramdata>
</diagramdatacollection>
- <action name="Approval1" type="Approval" displayname="Approval" description=""
looponoutput="">
- <properties>
- <property name="Action Display Name">
- <![CDATA[
]]>
</property>
- <property name="Delivery Channels">
- <propertychannel selectedchannelscount="1">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
- <channels>
- <channel name="WebWorkList">
<config url="" />
- <forms>
- <form name="Approval Form" type="NativeWebForm">
<actors grantaccess="True" />
<roles grantaccess="True" />
</form>
</forms>
</channel>
<removedchannels list="" />
</channels>
</propertychannel>
</property>
- <property name="Work Item Fields">
<Workitemfields />
</property>
- <property name="To">
- <resource>
- <globalfilter>
- <![CDATA[
]]>
</globalfilter>
- <dispglobalfilter>
- <![CDATA[
]]>
</dispglobalfilter>
- <filter>
- <![CDATA[
[UserDisplayName] = ["admin"]
]]>
</filter>
- <dispfilter>
- <![CDATA[
[Display Name] = ["admin"]
]]>
</dispfilter>
- <multipleproviderfilter>
- <![CDATA[
[UserDisplayName] = ["admin"]
]]>
</multipleproviderfilter>
- <multipleproviderdispfilter>
- <![CDATA[
[Display Name] = ["admin"]
]]>
</multipleproviderdispfilter>
</resource>
</property>
- <property name="Assign Queue">
- <queuevalue>
- <value>
- <![CDATA[
]]>
</value>
- <valueID>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
- <![CDATA[
]]>
</valueID>
</queuevalue>
</property>
- <property name="Min. Acknowledgements [% or Number]">
- <![CDATA[
1
]]>
</property>
- <property name="Max. Acknowledgements [% or Number]">
- <![CDATA[
100%
]]>
</property>
- <property name="Enable TakeOwnership">
- <![CDATA[
Yes
]]>
</property>
- <property name="Forwarding Allowed?">
- <![CDATA[
No
]]>
</property>
- <property name="Ignore Unavailable Resources?">
- <![CDATA[
No
]]>
</property>
- <property name="Alternate Resource Allowed?">
- <![CDATA[
Yes
]]>
</property>
- <property name="Subject">
- <workflowmemovalue>
- <height>
- <![CDATA[
350
]]>
</height>
- <width>
- <![CDATA[
650
]]>
</width>
- <value>
- <![CDATA[
<%#Content.IContentHandler.fromdate%>
]]>
</value>
- <valueHTML>
- <![CDATA[
&lt;%#Content.IContentHandler.fromdate%&gt;
]]>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</valueHTML>
- <classname>
- <![CDATA[
cc0ce05b681841758dc082a82f633b9a
]]>
</classname>
</workflowmemovalue>
</property>
- <property name="Notes">
- <workflowmemovalue>
- <height>
- <![CDATA[
350
]]>
</height>
- <width>
- <![CDATA[
650
]]>
</width>
- <value>
- <![CDATA[
]]>
</value>
- <valueHTML>
- <![CDATA[
]]>
</valueHTML>
- <classname>
- <![CDATA[
a097a01f59f4495087759bf1b2308e08
]]>
</classname>
</workflowmemovalue>
</property>
- <property name="From Email Address">
- <![CDATA[
admin@domain.com
]]>
</property>
- <property name="Hide Responses?">
- <![CDATA[
0
]]>
</property>
- <property name="Show Custom Document View Window">
- <![CDATA[
Yes
]]>
</property>
- <property name="Priority">
- <![CDATA[
Medium
]]>
</property>
- <property name="Send Notification Email">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
- <![CDATA[
No
]]>
</property>
- <property name="Notify if alternate considered">
- <![CDATA[
No
]]>
</property>
- <property name="Redirect URL">
- <workflowmemovalue>
- <height>
- <![CDATA[
350
]]>
</height>
- <width>
- <![CDATA[
650
]]>
</width>
- <value>
- <![CDATA[
]]>
</value>
- <valueHTML>
- <![CDATA[
]]>
</valueHTML>
- <classname>
- <![CDATA[
2c234c1402c24bdb8accd02ef50ff5ae
]]>
</classname>
</workflowmemovalue>
</property>
- <property name="Custom Document View URL">
- <workflowmemovalue>
- <height>
- <![CDATA[
350
]]>
</height>
- <width>
- <![CDATA[
650
]]>
</width>
- <value>
- <![CDATA[
]]>
</value>
- <valueHTML>
- <![CDATA[
]]>
</valueHTML>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
- <classname>
- <![CDATA[
dfb9a8e55be84cd49f5536364accced4
]]>
</classname>
</workflowmemovalue>
</property>
- <property name="Use html template from property">
- <![CDATA[
No
]]>
</property>
- <property name="HTML template">
- <workflowmemovalue>
- <height>
- <![CDATA[
350
]]>
</height>
- <width>
- <![CDATA[
650
]]>
</width>
- <value>
- <![CDATA[
]]>
</value>
- <valueHTML>
- <![CDATA[
]]>
</valueHTML>
- <classname>
- <![CDATA[
b93a2d9730ca4f77af6d7b051b634afb
]]>
</classname>
</workflowmemovalue>
</property>
- <property name="Affirmative Answer for Approval [% or Number]">
- <![CDATA[
100%
]]>
</property>
- <property name="Action Limit[% or Number]">
- <![CDATA[
100%
]]>
</property>
- <property name="Wait For All The Participants To Act">
- <![CDATA[
No
]]>
</property>
- <property name="Set Calendar">
- <![CDATA[
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
]]>
</property>
- <property name="Minimum Time for Action">
- <![CDATA[
0
]]>
</property>
- <property name="Timeout warning for Acknowledgement">
- <![CDATA[
0
]]>
</property>
- <property name="Timeout for Acknowledgement">
- <![CDATA[
0
]]>
</property>
- <property name="Timeout warning for Action">
- <![CDATA[
0
]]>
</property>
- <property name="Timeout for Action">
- <![CDATA[
0
]]>
</property>
- <property name="Multiple TimeOut Warning">
- <![CDATA[
]]>
</property>
- <property name="Enable Redirection?">
- <![CDATA[
No
]]>
</property>
</properties>
- <visual-attributes>
<height>65</height>
<isloop>false</isloop>
<top>75</top>
<width>65</width>
<left>275</left>
</visual-attributes>
</action>
- <action name="Start" type="Start" displayname="" description="" looponoutput="">
- <properties>
- <property name="Variables">
- <![CDATA[
0001#00380002$0018LeaveApplicationID$0006number
]]>
</property>
- <property name="Contents">
- <![CDATA[
0001#00540004$0018LeaveApplicationID$0006number$0006//test$0000
]]>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</property>
- <property name="XmlVariables">
- <XmlVariables>
- <XmlVariable>
- <Name>
- <![CDATA[
SKEventData
]]>
</Name>
- <Description>
- <![CDATA[
]]>
</Description>
- <StorageType>
- <![CDATA[
XmlDocument
]]>
</StorageType>
- <Scope>
- <![CDATA[
global
]]>
</Scope>
- <Schema>
- <![CDATA[
<?xml version="1.0" encoding="utf-16"?>
<xsd:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
attributeFormDefault="unqualified" elementFormDefault="qualified">
<xs:element name="SKEventData">
<xs:complexType>
<xs:sequence>
<xs:element name="EventProperties" />
<xs:element name="EventType" />
<xs:element name="EventProvider" />
<xs:element name="EventDateTimeUTC" type="xs:string" />
<xs:element name="EventDateTime" type="xs:string" />
<xs:element name="UserId" />
<xs:element name="UserName" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xsd:schema>
]]>
</Schema>
- <Xsl>
- <![CDATA[
]]>
</Xsl>
- <Storage>
- <InternalStorage>
- <![CDATA[
variables
]]>
</InternalStorage>
</Storage>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</XmlVariable>
</XmlVariables>
</property>
- <property name="ExecutionView">
- <![CDATA[
]]>
</property>
- <property name="SOAObjects">
<SOAObjects />
</property>
</properties>
- <visual-attributes>
<height>65</height>
<isloop>false</isloop>
<top>75</top>
<width>65</width>
<left>100</left>
</visual-attributes>
</action>
- <action name="Update Variable1" type="UpdateVariableValue"
displayname="UpdateVariableValue" description="" looponoutput="">
- <properties>
- <property name="Set Variable">
- <contentupdationfields>
- <contentupdationfield name="Variable.LeaveApplicationID" operator="=" mode="ObjTree"
methodName="c2c73d7d351242d29e8792000fdc5159">
- <![CDATA[
Content.LeaveApplicationID
]]>
</contentupdationfield>
- <classname>
- <![CDATA[
2f5c0ce2df16485cbd93022b5e6fd2bb
]]>
</classname>
- <jstring>
- <![CDATA[
[{"RHSMode":"ObjTree",
"LHSColumn":"Variable.LeaveApplicationID",
"RHSColumn":"Content.LeaveApplicationID",
"OPRColumn":"",
"ColumnExpression":{"DisplayExpressionString":"",
"ActualExprtessionString":"",
"MethodParameterList":"",
"HtmlVerexpString":"",
"ReferenceList":"",
"ReferencedFunctionList":"",
"NodeInfo":""}}]
]]>
</jstring>
</contentupdationfields>
</property>
- <property name="StopOnExceptions">
- <![CDATA[
Yes
]]>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</property>
</properties>
- <visual-attributes>
<height>65</height>
<isloop>false</isloop>
<top>225</top>
<width>65</width>
<left>125</left>
</visual-attributes>
</action>
- <link from="Update Variable1" to="Approval1" condition="Updated" description=""
name="Link0">
- <visual-attributes>
<sourceport>r</sourceport>
<sourceportside>r</sourceportside>
<sourceportindex>0</sourceportindex>
<sourceportfinalside>r</sourceportfinalside>
<sourceportfinalindex>0</sourceportfinalindex>
<sourceportfinalx>190</sourceportfinalx>
<sourceportfinaly>258</sourceportfinaly>
<targetport>b</targetport>
<targetportside>dynamic</targetportside>
<targetportindex>0</targetportindex>
<targetportfinalside>b</targetportfinalside>
<targetportfinalindex>0</targetportfinalindex>
<targetportfinalx>308</targetportfinalx>
<targetportfinaly>140</targetportfinaly>
<serializedpath>Link0;false;3;308;258;Link0,right,0,213,258,190,258,false,false,NaN,NaN,!,,
Link0_0,;Link0,,0,213,258,213,258,false,false,NaN,NaN,!,,Link0_1,;Link0,right,0,308,258,213
,258,false,true,260,258,!,,Link0_2,;Link0,up,0,308,163,308,258,false,true,308,210,!,,Link0_
3,;Link0,,0,308,163,308,163,false,false,NaN,NaN,!,,Link0_4,;Link0,up,0,308,140,308,163,true
,false,NaN,NaN,!,,Link0_5,</serializedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
- <link from="Start" to="Update Variable1" condition="" description="" name="Link1">
- <visual-attributes>
<sourceport>b</sourceport>
<sourceportside>dynamic</sourceportside>
<sourceportindex>0</sourceportindex>
<sourceportfinalside>b</sourceportfinalside>
<sourceportfinalindex>0</sourceportfinalindex>
<sourceportfinalx>133</sourceportfinalx>
<sourceportfinaly>140</sourceportfinaly>
<targetport>t</targetport>
<targetportside>dynamic</targetportside>
<targetportindex>0</targetportindex>
<targetportfinalside>t</targetportfinalside>
<targetportfinalindex>0</targetportfinalindex>
<targetportfinalx>158</targetportfinalx>
<targetportfinaly>225</targetportfinaly>
<serializedpath>Link1;false;2;133;195;Link1,down,0,133,154,133,140,false,false,NaN,NaN,!,,L
ink1_0,;Link1,,0,133,154,133,154,false,false,NaN,NaN,!,,Link1_1,;Link1,down,0,133,210,133,1
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
54,false,true,133,182,!,,Link1_2,;Link1,right,0,158,210,133,210,false,false,0,0,!,,Link1_3,
;Link1,down,0,158,222,158,210,false,false,0,0,!,,Link1_4,;Link1,,0,158,222,158,222,false,fa
lse,NaN,NaN,!,,Link1_5,;Link1,down,0,158,225,158,222,true,false,NaN,NaN,!,,Link1_6,</serial
izedpath>
<thickness>1</thickness>
<color>#000000</color>
<linetype />
<islocked>false</islocked>
</visual-attributes>
</link>
</workflow>
```

Use Custom Content Handler

AVEVA Work Tasks allows us to create custom content handlers which can be used for fetching data to workflow in custom logic.

To implements Custom Content Handler

1. Create a Class Library Project. Code for the sample custom Class Library
2. Compile and put the dll in the AVEVA Work Tasks Bin folder.
3. The entry for default content handler will be already there in the SKAddInProviders table. Rename the 'Name' field of the specific entry from 'IContentHandler' to some other name.
4. Make the following entry in the SKAddInProviders table in the repository database.

   ```
   INSERT INTO [SKAddInProviders]
   (Type,[Name],[ClassName],[Assembly],[Settings],[IsGacAssembly])
   VALUES('ContentHandler','IContentHandler','CustomContentHandler.DatabaseContentHandle
   r',bin\CustomContentHandler.dll','',0)
   ```
5. Follow the steps in Using Default Content Handler which will successfully create the Custom Content handler.

Code Sample - Custom Content Handler Class File

```
using System;
using System.Web.Caching;
using System.Collections;
using System.Xml;
using System.Xml.XPath;
using System.IO;
using System.Data;
using System.Text.RegularExpressions;
using Workflow.NET.Interfaces;
using Workflow.NET;
using Workflow.NET.Engine.Interfaces;
using Workflow.NET.Storage;
using Skelta.Core;
using Workflow.NET.Engine;
namespace CustomContentHandler
{
 public class DatabaseContentHandler : IContentHandler, ISkeltaAddInProvider
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
CacheWrapper cache;
string sConnectionString;
string sConnectionAssembly;
string sConnectionClass;
string sConnectionParam;
DataSet data;
string dataSourceType = null;
string connectionString = null;
SortedList ContentCollection;
AppContentsCollection filteredContent;
Context ctx;
Log logger;
string name;
public virtual string Name
{
get
{
  return this.name;
}
  set
{
  this.name = value;
}
}
  public DatabaseContentHandler()
{
logger = new Log();
}
  IContentHandler IContentHandler.Clone()
{
  DatabaseContentHandler contentHandler = new DatabaseContentHandler();
contentHandler.connectionString = this.connectionString;
contentHandler.dataSourceType = this.dataSourceType;
contentHandler.data = this.data;
contentHandler.ctx = this.ctx;
contentHandler.connectedPreviously = this.connectedPreviously;
contentHandler.filteredContent = this.filteredContent;
contentHandler.ContentCollection = this.ContentCollection;
contentHandler.logger = this.logger;
contentHandler.name = this.name;
  return contentHandler;
}
  public virtual void InitializeContent(Context ctx, string Param)
{
  try
{
filteredContent = new AppContentsCollection();
  this.ctx = ctx;
string applicationName = "";
string[] temp = Param.ToString().Split(new string[] { "#SKC#" }, StringSplitOptions.None);
                if(temp.Length > 0)
                {
                    Param = temp[0];
                    applicationName = temp[1];
                }
ContentCollection = new SortedList();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
data = new DataSet("content");
  XPathDocument doc;
  //Get the data source type and connection string
  if (!File.Exists(Param))
{
doc = new XPathDocument(new StringReader(Param));
}
  else
{
doc = new XPathDocument(Param);
}
  XPathNavigator nav = doc.CreateNavigator();
  XPathNodeIterator iterator = nav.Select("//datasource");
  if (iterator.MoveNext())
{
dataSourceType = iterator.Current.GetAttribute("type", "");
connectionString = iterator.Current.GetAttribute("connectionString", "");
sConnectionClass = iterator.Current.GetAttribute("class", "");
sConnectionAssembly = iterator.Current.GetAttribute("assembly", "");
sConnectionParam = iterator.Current.GetAttribute("param", "");
connectionString = iterator.Current.Value;
sConnectionString = connectionString;
  if (ctx != null)
{
cache =
  new CacheWrapper();
  bool showLog = (ctx.ProcessDefinition.ActionsDefinition.WorkflowConfigFile.DebugLevel
== "111") ? true : false;
  string application = ctx.ProcessDefinition.ApplicationName;
  string cacheKeyName = (sConnectionParam == null || sConnectionParam == "") ? application
: sConnectionParam;
  if ((sConnectionClass != null && sConnectionClass != "") &&
sConnectionAssembly != null && sConnectionAssembly != "")
{
  if (cache.Cache["ContentHandlerConnString" + cacheKeyName] != null)
{
connectionString = cache.Cache["ContentHandlerConnString" + cacheKeyName].ToString();
  if (showLog)
logger.LogInformation("Getting connection string from Cache (Content Handler):" +
connectionString);
}
  else
{
  //Set the dependency files to invalidate the cache.
System.Web.Caching.CacheDependency fileDependency;
  if (Param != null && Param != "" && Param.ToUpperInvariant().IndexOf(".XML") >= 0)
fileDependency = new CacheDependency(new string[] { Config.GetConfigFilePath()
+ "WorkflowNET.Config.Xml", Param });
  else
fileDependency = new CacheDependency(Config.GetConfigFilePath()
+ "WorkflowNET.Config.Xml");
connectionString = CommonFunctions.GetConnectionString("Content Handler", sConnectionClass,
sConnectionAssembly, sConnectionParam, sConnectionString, application);
cache.Cache.Add("ContentHandlerConnString" + cacheKeyName, connectionString,
fileDependency, DateTime.MaxValue,
  TimeSpan.Zero, System.Web.Caching.CacheItemPriority.Normal, null);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
  if (showLog)
logger.LogInformation("Getting connection string from DLL(Content Handler):" +
connectionString);
}
}
}
}
  else
{
  throw new Exception("Datasource not configured in the file " + Param);
}
  //Read the tables
iterator = nav.Select("/content/tables/table");
  while (iterator.MoveNext())
{
  XPathNavigator table = iterator.Current;
  try
{
ParseTable(table);
}
  catch (Exception e)
{
logger.LogError(e, "Could not parse the xml for table " + table + ".");
  throw e;
}
}
}
  catch (Exception e)
{
logger.LogError(e, "Could not initialize the application content handler");
  throw e;
}
}
  public virtual void Clear()
{
filteredContent.Clear();
}
  void ParseTable(XPathNavigator Table)
{
  string tableName = Table.GetAttribute("name", "");
  FilterDataTable dt = new FilterDataTable(tableName);
data.Tables.Add(dt);
  //Console.WriteLine(tableName);
  //Prase the fields
  XPathNodeIterator fields = Table.Select("//tables/table[@name='" + tableName
+ "']/fields/*");
  while (fields.MoveNext())
{
  string fldName = fields.Current.GetAttribute("name", "");
  //Console.WriteLine(fldName);
  string fldType = fields.Current.GetAttribute("type", "");
  string fldSize = fields.Current.GetAttribute("size", "");
  if (fields.Current.GetAttribute("identifier", "") != null &&
fields.Current.GetAttribute("identifier", "") == "true")
{
dt.Identifier = fldName;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
  string contentName = fields.Current.GetAttribute("contentName", "");
  string canUpdate = fields.Current.GetAttribute("canUpdate", "");
  ContentField field = new ContentField(fldName, fldType);
field.CanUpdate = canUpdate.ToLowerInvariant() == "true" ? true : false;
dt.Columns.Add(field);
ContentCollection.Add(contentName, field);
  //Can populate the content collection in this method if required.
}
((FilterDataTable)dt).FilterCondition =
Table.SelectChildren("filter", "").Current.Value.Trim();
}
  bool connectedPreviously = false;
  IDataHandler GetConnection(string DatasourceType, string ConnectionString)
{
  IDataHandler dataHandler = null;
  if (connectedPreviously)
{
dataHandler = DataHandlerFactory.GetDataHandler(
ConnectionString, DatasourceType);
}
  else
{
dataHandler = DataHandlerFactory.GetDataHandler(
ConnectionString, DatasourceType, 3);
}
connectedPreviously = true;
  return dataHandler;
}
  public virtual AppContent GetContent(string ContentName)
{
  if (filteredContent[ContentName] != null && filteredContent[ContentName].Value != null)
{
  return filteredContent[ContentName];
}
  else
{
  AppContent content = GetContent(new string[] { ContentName })[ContentName];
  if (content != null && content.Value != null)
{
  return content;
}
  else
{
  return null;
}
}
}
  public virtual AppContentsCollection GetContent(params string[] ContentNames)
{
  AppContentsCollection tempContentCollection = new AppContentsCollection();
  IDataReader reader = null;
  IDataHandler dataHandler = null;
  try
{
  bool firstIteration = true;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
  string sql = null;
  string tableName = null;
  foreach (string contentName in ContentNames)
{
  //Get the field for the content.
  ContentField field = (ContentField)ContentCollection[contentName];
  string fieldName = field.ColumnName;
tableName = field.Table.TableName;
  if (firstIteration)
{
  string identifier = ((FilterDataTable)data.Tables[tableName]).Identifier;
sql = "select " + identifier + " as identifier, " + fieldName;
firstIteration = false;
}
  else
{
sql += ", " + fieldName;
}
}
sql += " from " + tableName;
  while (true)
{
  try
{
dataHandler = GetConnection(dataSourceType, connectionString);
  string tempSql = sql;
  string filter = ((FilterDataTable)data.Tables[tableName]).FilterCondition;
  if (filter != null && filter != "")
{
  Regex regex = new Regex("<%#([a-zA-Z0-9]*).([a-zA-Z0-9]*)%>");
  MatchCollection matches = regex.Matches(filter);
  ArrayList parameters = new ArrayList(matches.Count);
  int i = 1;
  if (matches != null)
{
  foreach (Match match in matches)
{
  string strToBeReplaced = match.Value;
  string type = match.Groups[1].Value;
  string key = match.Groups[2].Value;
  object val = null;
  if (type.ToLowerInvariant() == "content")
{
  if (ctx.Contents[key].Value != null)
{
val = ctx.Contents[key].Value;
}
}
  else if (type.ToLowerInvariant() == "variable")
{
  if (ctx.Variables.Contains(key) && ctx.Variables[key].Value != null)
{
val = ctx.Variables[key].Value;
}
}
  else
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
val = "";
}
filter = filter.Replace(strToBeReplaced, "@param" + i);
  IDataParameter param = dataHandler.GetParameter("@param" + i, val);
parameters.Add(param);
i++;
}
}
tempSql += " where " + filter;
reader = dataHandler.ExecuteReader(tempSql,
(IDataParameter[])parameters.ToArray(typeof(IDataParameter)));
}
  else
{
reader = dataHandler.ExecuteReader(sql, null);
}
  if (reader.Read())
{
  object identifierValue = reader["identifier"];
  //Read the values
  foreach (string contentName in ContentNames)
{
  AppContent content = new AppContent();
  ContentField field = (ContentField)ContentCollection[contentName];
content.Name = contentName;
content.Identifier = identifierValue;
content.Value = reader[field.ColumnName];
content.Type = field.Type;
content.CanUpdate = field.CanUpdate;
filteredContent[content.Name] = content;
tempContentCollection[content.Name] = content;
}
reader.Close();
}
  break;
}
  catch (Exception e)
{
  if (dataHandler != null && dataHandler.ConnectionState == ConnectionState.Open)
{
logger.LogError(e, "Could not get the content from " + name);
  throw e;
}
  else
{
logger.LogWarning("Database connection lost abruptly, will try to connect again");
}
}
  finally
{
  try
{
dataHandler.Dispose();
}
  catch { }
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
}
}
  catch (Exception e)
{
logger.LogError(e, "Could not get the content " + String.Join(",", ContentNames, 0,
ContentNames.Length) + ".");
}
  return tempContentCollection;
}
  public virtual AppContent GetContent(object Identifier, string ContentName)
{
  AppContent content = GetContent(Identifier, new string[] { ContentName })[ContentName];
  if (content != null && content.Value != null)
{
  return content;
}
  else
{
  return null;
}
}
  public virtual AppContentsCollection GetContent(object Identifier, params string[]
ContentNames)
{
  AppContentsCollection contentValues = new AppContentsCollection();
  try
{
  string sql = "select ";
  bool firstIteration = true;
  string tableName = "";
  string identifierFieldName = "";
  foreach (string contentName in ContentNames)
{
  ContentField field = (ContentField)ContentCollection[contentName];
  string fieldName = field.ColumnName;
  if (firstIteration)
{
tableName = field.Table.TableName;
  FilterDataTable table = (FilterDataTable)data.Tables[tableName];
identifierFieldName = table.Identifier;
firstIteration = false;
sql += identifierFieldName + " as identifier," + fieldName;
}
  else
{
sql += "," + fieldName;
}
}
sql += " from " + tableName + " where " + identifierFieldName + "=@paramId";
  IDataHandler dataHandler = null;
  while (true)
{
  try
{
dataHandler = GetConnection(this.dataSourceType, this.connectionString);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
  IDataParameter paramId = dataHandler.GetParameter("@paramId", Identifier);
  IDataReader reader = dataHandler.ExecuteReader(sql, paramId);
  if (reader.Read())
{
  object identifierValue = reader["identifier"];
  foreach (string contentName in ContentNames)
{
  AppContent content = new AppContent();
content.Name = contentName;
content.Identifier = identifierValue;
content.Value = reader[((ContentField)ContentCollection[contentName]).ColumnName];
contentValues.Add(contentName, content);
}
}
reader.Close();
  break;
}
  catch (Exception e)
{
  if (dataHandler != null && dataHandler.ConnectionState == ConnectionState.Open)
{
logger.LogError(e, "Could not get the content from " + name);
  throw e;
}
  else
{
logger.LogWarning("Database connection lost abruptly, will try to connect again");
}
}
  finally
{
  try
{
dataHandler.Dispose();
}
  catch { }
}
}
}
  catch (Exception e)
{
logger.LogError(e, "Could not get the content " + String.Join(",", ContentNames, 0,
ContentNames.Length) + " with identifier " + Identifier + ".");
  throw e;
}
  return contentValues;
}
  public virtual object[] FilterContent(string FullyProcessedQuery, params object[]
Parameters)
{
  IDataHandler dataHandler = null;
  while (true)
{
  try
{
  ArrayList identifiers = new ArrayList();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
dataHandler = GetConnection(this.dataSourceType, this.connectionString);
  ArrayList contentParameters = new ArrayList();
  if (Parameters != null)
{
  foreach (object param in Parameters)
{
  object[] paramArray = (object[])param;
contentParameters.Add(dataHandler.GetParameter((string)paramArray[0], paramArray[1]));
}
}
  IDataReader reader = dataHandler.ExecuteReader(FullyProcessedQuery,
(IDataParameter[])contentParameters.ToArray(typeof(IDataParameter)));
  while (reader.Read())
{
identifiers.Add(reader[0]);
}
reader.Close();
  return identifiers.ToArray();
}
  catch (Exception e)
{
  if (dataHandler != null && dataHandler.ConnectionState == ConnectionState.Open)
{
logger.LogError(e, "Could not filter the content using the query \"" + FullyProcessedQuery
+ "\".");
  throw e;
}
  else
{
logger.LogWarning("Database connection lost abruptly. Will try to connect to the database
again.");
}
}
  finally
{
  try
{
dataHandler.Dispose();
}
  catch { }
}
}
}
  public virtual AppContentsCollection GetContentDetails()
{
  AppContentsCollection contents = new AppContentsCollection();
  foreach (string contentName in ContentCollection.Keys)
{
  ContentField field = (ContentField)ContentCollection[contentName];
  AppContent content = new AppContent();
content.Name = contentName;
content.Type = field.Type;
contents.Add(contentName, content);
}
  return contents;
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
  public virtual bool UpdateContent(AppContent Content)
{
  return UpdateContent(new AppContent[] { Content });
}
  public virtual bool UpdateContent(params AppContent[] ContentList)
{
  IDataHandler dataHandler = null;
  int rowsUpdated = 0;
  while (true)
{
  try
{
  using (System.Transactions.TransactionScope TScope =
Skelta.Core.DS.SingleTableObject.GetTransactionScope())
{
dataHandler = GetConnection(this.dataSourceType, this.connectionString);
  using (dataHandler)
{
  bool firstIteration = true;
  string sql = null;
  string tableName = null;
  object identifierValue = null;
  ArrayList parameters = new ArrayList();
  int i = 1;
  foreach (AppContent content in ContentList)
{
  if (!((ContentField)ContentCollection[content.Name]).CanUpdate)
{
  throw new Exception("Can not update the content that is not updatable.");
}
  //Get the field for the content.
  ContentField field = (ContentField)ContentCollection[content.Name];
  string fieldName = field.ColumnName;
  if (firstIteration)
{
tableName = field.Table.TableName;
identifierValue = content.Identifier;
sql = "update " + tableName + " set " + fieldName + "=@param" + i;
firstIteration = false;
}
  else
{
sql += ", " + fieldName + "=@param" + i;
}
parameters.Add(dataHandler.GetParameter("@param" + i, content.Value));
i++;
filteredContent[content.Name] = content;
}
  string identifier = ((FilterDataTable)data.Tables[tableName]).Identifier;
sql += " where " + identifier + "=@paramId";
parameters.Add(dataHandler.GetParameter("@paramId", identifierValue));
rowsUpdated = dataHandler.ExecuteUpdate(sql,
(IDataParameter[])parameters.ToArray(typeof(IDataParameter)));
}
  if (rowsUpdated > 0)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
  return true;
}
  else
{
  return false;
}
TScope.Complete();
}
}
  catch (Exception e)
{
  if (dataHandler != null && dataHandler.ConnectionState == ConnectionState.Open)
{
logger.LogError(e, "Could not update the content values.");
filteredContent.Clear();
  throw e;
}
  else
{
logger.LogWarning("Database connection lost abruptly. Will try to connect to the database
again.");
}
}
}
}
  public virtual void Dispose()
{
  //ctx.Close();
}
  class FilterDataTable : DataTable
{
  internal FilterDataTable(string TableName)
: base(TableName)
{
}
  internal string FilterCondition = null;
  internal string Identifier = null;
}
  public class ContentField : DataColumn
{
  public string Type = "string";
  internal ContentField(string FieldName, string DbType)
{
  this.Type = DbType;
  Type dbType = typeof(string);
  switch (DbType)
{
  case "int":
dbType = typeof(int);
  break;
  case "float":
dbType = typeof(double);
  break;
  case "varchar":
  break;
  case "datetime":
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
dbType = typeof(DateTime);
  break;
  default:
  break;
}
  this.ColumnName = FieldName;
  this.DataType = dbType;
}
  public bool CanUpdate;
}
#region ISkeltaAddInProvider Members
  string _Settings = "";
  Guid _Id = Guid.Empty;
  void ISkeltaAddInProvider.InitializeProvider(string settings, Guid id)
{
_Settings = settings;
_Id = id;
}
  string ISkeltaAddInProvider.Settings
{
  get { return _Settings; }
}
  Guid ISkeltaAddInProvider.Id
{
  get { return _Id; }
}
#endregion
}
}
```

Persist Logged Exceptions to Database

When the workflow is triggered or when the user submits the activity, or when the workflow engine sends an alert about an activity, the action may fail and thus also the workflow. It becomes necessary for the users to understand the main cause for the failure of the workflow so that they can modify/handle it and thus improve the process life cycle.

In the previous version of the AVEVA Work Tasks product, exceptions were stored in the log file, logged to the Event Viewer and sending an email based on the configuration. However, to troubleshoot any issue, it was more difficult for users to track the exact exception message, based on the Execution ID (say).

To help the users to trace the exact exception message, all the exceptions and warnings can be persisted in the database. So, the users can now see all the exceptions occurred for that workflow by selecting the "View Exceptions" from the workflow item context menu. This pops up a window which lists all the exceptions occurred for that particular workflow.

Enable Exception Persistence

To enable the persistence of exceptions and warnings into the database so that all these can be viewed from **View Exceptions**, do the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Go to **Farm Configuration** wizard.
2. Click **Service Settings**. The **Service Settings** dialog box appears.
3. Click **Global Settings**, and then select Log/Transaction tab.
4. Select the correct values to enable the persistence of exceptions and warnings.

Different Views for Viewing the Exceptions

There are three view options for viewing exceptions:

- Workflow level
- Instance level exception
- Workflow action step level

Workflow Level

There are three different ways to know all the exceptions occurred.

Workflow level

Right-click the workflow item, and then select the **View Exceptions** option. The **Exception Details** page appears. This page lists all the exceptions that has occurred for the workflow and all the exceptions that has occurred for different workflow instances.

Note:
- For the Quickflows, the **Exception Details** page will not show the values for **ExecutionDetailsID** field.
- For the Quickflows containing Child Quickflows, the **Exception Details** page will not show the values for **ExecutionID** field.

Instance Level Exception

You can view the instance level exceptions from the BAM reports.

1. Go to **Enterprise Console** menu, and then click **Reports**.
   The **Business Activity Monitoring** page appears.
2. Click **Analyze**, click **Transaction**, and then click **Failed**.
   Details of **All Workflows Executed** with Errors appear on the right pane.
3. Double-click an item to open.
4. Click the **View Exceptions** tab to see all the exceptions that has occurred for the particular workflow instance.

Workflow Action Step Level

You can view the workflow action step level exceptions from the BAM reports.

1. Go to **Enterprise Console** menu, and then click Reports.
   The **Business Activity Monitoring** page appears.
2. Click **Analyze**, click **Transaction**, and then click **Failed**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Details of **All Workflows Executed with Errors** appear on the right pane.

3. Double-click an item to open.

   You can see the workflow processes from **Analyst View**, **Progress View**, or **Execution View** in the dialog box that appears.

4. Click any tab, and then double-click the failed activity.

   You can see the details of the particular activity in the dialog box that appears.

5. Click the **View Exceptions** tab to see all the exceptions that has occurred for the particular workflow instance.

AVEVA Work Tasks Pro Application

AVEVA Work Tasks Pro is a premium mobile application for AVEVA Work Tasks. It is designed for business professionals who use a mobile phone or a tablet device to access the app while on the move. AVEVA Work Tasks Pro can view and act on tasks, fill the form that initiates a process, and submit forms offline, among other capabilities.

Custom Conflict Management

For the Fill Forms submission, there is no conflict management by default. However, there is a basic conflict management for work items such as **Approval Activity** and **Invoke Form Activity**. The user can customize this by creating a conflict management component and registering the same in the **SKAddinProviders** table under the repository database.

Perform the following steps to create a custom conflict management component.

1. Create a .NET Class library project.
2. Refer the **SE.Bpm.ConflictHandler.dll**.
3. Create a class which implements the **IConflictHandler** interface. This interface has the following methods:
   - **ValidateWorkItem** – When this function is implemented, it validates the work item being submitted.
   - **ValidateFillForm** – When this function is implemented, it validates the Fill Form being submitted.
   - **ResolveWorkItem** – When this function is implemented, it resolves the work item, if there is a conflict.
   - **ResolveFillForm** – When this function is implemented, it resolves the Fill Form, if there is conflict.
   - **CanResolve** – When this property is implemented, it has a value indicating when the conflict can be resolved.
   - **CanContinue** – When this property is implemented, it has a value indicating whether the process (submission of work items or Fill Forms) can continue even if there is a conflict.
4. Build the component.
5. Place the component in a path, which is accessible to the Workflow Engine Service and Enterprise Console web application.
6. Provide the class name and the dll path in the **SKAddinProviders** table in the repository database.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Providers for Controls

AVEVA Work Tasks supports providers of different types that deliver the basic functionality for the Controls. The following list describes the providers:

- **Channel Provider**: Used to configure channels.
- **Form Provider**: Used to configure forms.
- **Push Dispatch Provider**: Used to configure the push-type (automatic) dispatch pattern in Queues.
- **Pull Dispatch Provider**: Used to configure the pull-type (automatic) dispatch pattern in Queues.
- **Work hour Provider**: Used to configure calendars.
- **XML Storage Provider**: Used to configure the XML Variables.
- **UserEntity**: Used to configure the entity provider.
- **UserDSProvider**: Used to configure the entity Data Source Provider.
- **ContentHandler**: Used to configure the Content Handler provider.
- **Wizard**: Used to configure the Wizard provider.

AVEVA Work Tasks provides ready-to-use providers, out-of-the-box for all the seven types it supports. In addition, AVEVA Work Tasks provides the infrastructure to build different types of custom providers.

Provider Type	Namespace	Interface
Role Provider	Skelta.Core Skelta.Entity	ISkeltaRoleProvider IRoleEntityProvider ISkeltaAddInProvider IRoleEntityDataSourceProvider
Channel Provider	Skelta.HWS.WorkListChannel	IWorkListChannel
Form Provider	Skelta.HWS.WorkListChannel	IWorkItemForm
Push Dispatch Provider	Skelta.HWS.Queue	IPushDispatchProvider
Pull Dispatch Provider	Skelta.HWS.Queue	IPullDispatchProvider
Work hour Provider	Skelta.HWS.Calendar	IWorkHoursProvider
XML Storage Provider	Workflow.NET.Interfaces	IXmlStorageType
User Entity	Skelta.Entity.UserEntityProvider	ISkeltaAddInProvider IEntityProvider IUserEntityProvider
UserDSProvider	Workflow.NET Skelta.Entity	IEntityDataSourceProvider IUserEntityDataSourceProvider
Contenthandler	Workflow.NET	IContentHandler

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Provider Type	Namespace	Interface
Wizard	Workflow.NET	-

Creating Custom Providers

You can create your own custom providers by implementing the respective interface along with the ISkeltaAddInProvider interface available under the Skelta.Core namespace.

The following code shows a typical implementation of a custom provider:

```
using Skelta.Core;
namespace myApp.myRoleProvider
{
public class myRoleProvider : ISkeltaRoleProvider, ISkeltaAddInProvider
{
// Implementation of ISkeltaRoleProvider Members
// Implementation of ISkeltaAddInProvider Members
}
}
```

The custom provider should be registered in the SKAddInProviders table for AVEVA Work Tasks to recognize it and offer the desired runtime functionality.

Column Name	Type	Description
Id	Guid	Unique ID of the provider.
Type	nvarchar(20)	Type identification of provider.
Name	nvarchar(100)	Provider name.
Description	nvarchar(150)	Provider description.
ClassName	nvarchar(200)	Class Name implementing the interface.
Assembly	nvarchar(500)	Assembly name with relative or complete file system path.
Settings	nvarchar(max)	Configuration settings known the provider.
IsGacAssembly	bit	Determines assembly resides in Global Assembly Cache.
IsDefault	bit	Determines if the value is default.
IsGlobal	bit	Determines if the value is global.

The settings column is used to store specific settings required for the providers functionality. That is, the provider can access the settings stored in this column and perform the desired routine. The providers ID and settings are

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

available as properties through the ISkeltaAddInProvider interface implementation.

If the provider requires settings for each application or workflow, the provider registration should also happen in the SKAddInProvidersSettings table.

Use the following query to register the Provider in the SKAddInProviders table:

```
INSERT INTO
[SKAddInProviders]([Type],[Name],[Description],[ClassName],[Assembly],[Settings],[IsGacAsse
mbly],[LastUpdatedDatetime]) VALUES (Provider,<your Provider name>,<your Provider
description>,<your project name(which contains YourProvider.cs)>,<location of your project
dll (which contains YourProvider.cs)> ,'' ,0,getdate())
```

The structure of this table supports the storing of an application or a workflow as shown below.

Column Name	Type	Description
Id	Guid	Unique ID of the settings.
Application	nvarchar(50)	Application name
Workflow	nvarchar(50)	Workflow name.
AddInProviderId	Guid	Provider ID.
Settings	nvarchar(max)	Configuration settings known to the provider.
IsDefault	bit	Determines if the value is default.

The provider infrastructure tries to look for any specific settings of the matching application or workflow, and returns the value accordingly, when the Settings property is accessed.

Option to Disable Logging of Strings

You use this feature to control the string messages that are logged into the logger files without making any change to the debug level.

Debug Level: The number of messages getting displayed in the logger console varies based on the Debug level set in the Farm Configuration Wizard.

1. Open the **Farm Configuration** Wizard.
2. Click **Next**.
3. Select **Service Settings**, and then select **Global Settings**.
4. Select **Log/Transaction** tab. Here you can select the **Debug** level from the drop-down list.

   To know more about setting debug level, see *Prerequisites* section in the *Administrator Guide*.

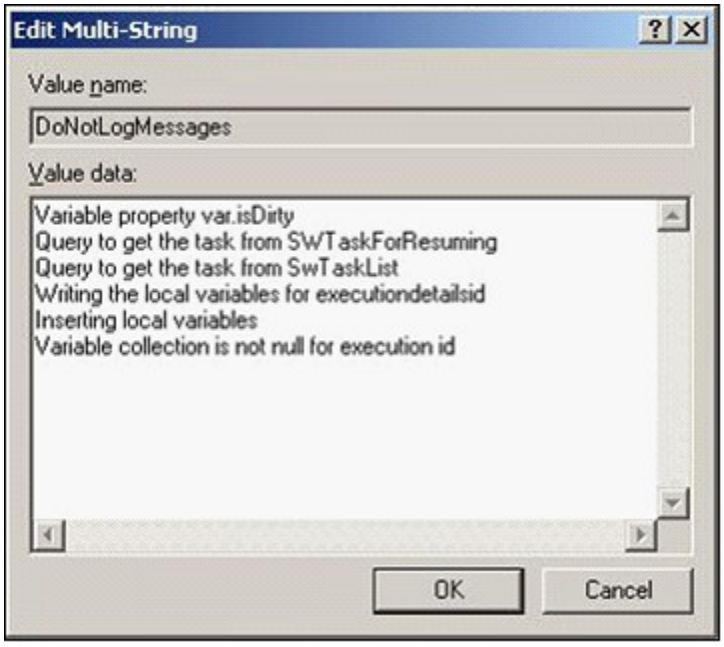
Example Scenario:

Suppose the debug level is set as 2, and you want to restrict some messages containing text, for example,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

"Inserting Local Variables" from getting logged into the logger file without changing your current debug level.

1. Create a new key in the **AVEVA Work Tasks** registry named "DoNotLogMessages" of type multi-string value.
2. Specify the text of the messages which are to be ignored while writing to the log files.



3. Restart the AVEVA Work Tasks services.

Note: These steps control the string messages logged into the AVEVA Work Tasks log files. The same messages appears in the Logger Console.

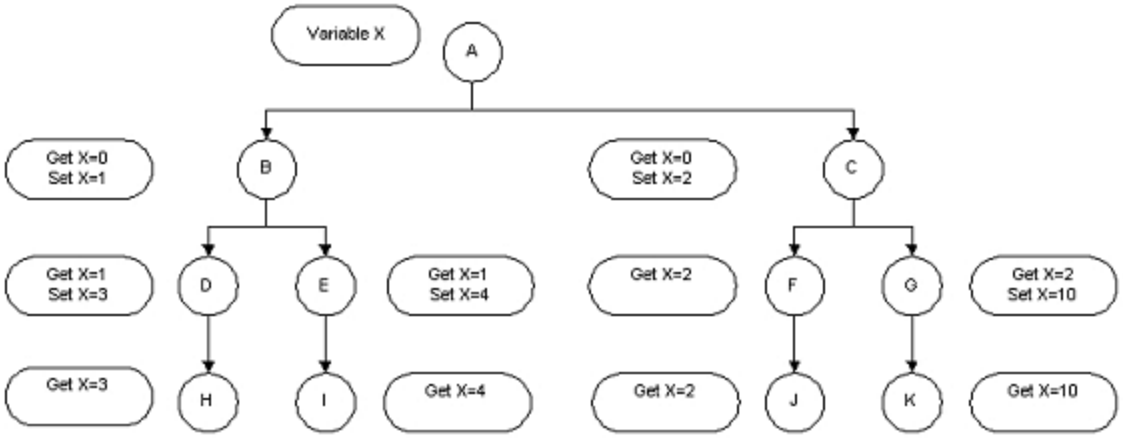
Use Local Variables

The Local Variable stores the values set by the incoming action. The values that are set for such variables are visible only to the subsequent actions.

Example Scenario:

The following image explains the use of a local variable 'X' in a workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The variable is added to the first action 'A'. Action 'A' initially sets the value of the variable 'X' to 0. The action 'B' sets a value 1 to the variable 'X' and 'C' sets a value 2. All the immediate actions after 'B' will get the value 1 from the variable 'X', while the actions after 'C' will get the value 2 from the same variable 'X'. Any changes made by any action to a Local Variable will be visible only to the actions that occur immediately after it.

Example of Local Variable Usage

Add local variable to the Start action only. Prefix the name of the local variables with '@'.

Call the Context.SaveVariables method to persist the state of the variables. A new field is added to the SWExecutionDetails table to persist the local variables.

The local variables are loaded for an action using Context.LoadLocalVariables (executionDetailsId) method.

The local variables can be accessed at run time by the actions through the Variables collection of the Context object.

Set Custom Properties for Repository Data Source

While creating a data source for repository, use the following custom settings to set properties for a connection pool.

```
encrypt=true; nableltm=true; connectionretrywaittime=30; maxconnectionretries=10;
sqlcommandtimeout=300;
```

encrypt: Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption.

enableltm: AVEVA Work Tasks internally implements lightweight transaction mode by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction becoming distributed is reduced to a great extent. With this feature AVEVA Work Tasks control the transaction from escalating to a distributed transaction which is very expensive.

The enableltm property can be used to turn off this feature. The default value is set as **'True'**.

connectionretrywaittime: While opening a new connection, if the database server is not available, the default behavior is to retry for the connection. This setting can be used to set the wait time to retry for a new connection. The default value is set as 30 seconds.

maxconnectionretries: While opening a new connection, if the database server is not available, the default

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite.

sqlcommandtimeout: Sets the wait time before terminating the attempt to execute a SQL command and generating an error. The default value is set as 300.

Note: These settings are case sensitive and should be mentioned only in lower case.

Enable Xcopy Deployment

AVEVA Work Tasks enables Xcopy deployment of dlls. Xcopy enables direct replacement of dlls by ensuring that they are not locked by the processes. Xcopy deployment is not applicable to the Windows Services, Workflow.NET.dll, Workflow.NET.Communication.dll, and Custom dlls configured for getting connection string.

Complete the following steps to enable Xcopy deployment:

1. Stop all the AVEVA Work Tasks services
2. Ensure that the custom application folders have been configured in the Repository Settings in the Central Configuration Site.
3. Reference the "Workflow.NET.Web.SkeltaXcopy.dll" from "bin" folder of the AVEVA Work Tasks installation directory.
4. Ensure that the file watcher has been configured for the application in the Repository Settings.
5. Enable AVEVA Work Tasks Xcopy in the Application_Start method in the Global.asax.cs of the application.

```
/// <summary>
/// Summary description for Global.
/// </summary>
public class Global : System.Web.HttpApplication
{
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    public Global()
    {
        InitializeComponent();
    }

    protected void Application_Start(Object sender, EventArgs e)
    {
        Workflow.NET.Web.SkeltaXcopy.Xcopy.Enable("Default");
    }
    protected void Session_Start(Object sender, EventArgs e)
    {

    }
```

6. After configuring Xcopy, the dlls will not be locked by the processes and can be replaced directly.
7. Restart all AVEVA Work Tasks Services.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Human Activity - Enhancement for Escalation and Time out

You can enter the time span value as:

1. "Thursday 07:30 PM"
2. "Thursday 07:30 "(Assumes AM By default)
3. "19:30 Thursday"
4. "PM 07:30 Friday" (tomorrow)
5. "21:00 Next Friday"/"07:35 PM Next Friday" (takes the Friday which comes in the next week)

Note: Always enter time in the "HH:MM" format. If the Hour is a single digit, it should be prefixed with a "0". For example, 7:30 should be entered as 07:30 and the colon ":" symbol between the hour and minutes is mandatory.

To do this, the you must create a predefined variable in the following format and update the value of this variable from Script or Update Variable Activity. Ensure that the data type of the variable is set to "string".

Syntax: ^<Action_Name(Case Sensitive**)>_<AcronymOfProperty**(Case Insensitive)>

For example, for setting the variable for Timeout for Action, it should be:
^Approval1_TFA

Different **<AcronymOfProperty>** which can be set are:

"TFA": "Timeout for Action"
"MATFA": "Minimum Time for Action"
"TWFO": "Timeout warning for Acknowledgement"
"TFO": "Timeout for Acknowledgement"
"TWFA": "Timeout warning for Action"

You do not need to specify anything in the property. At runtime, this value will be populated to the relevant property of the relevant activity.

If you specify anything in the Property explicitly, the value entered will override the variable value.

If you have a specific date-time value, then you must set the DataType of the Variable as **date**, and update the value of this variable from Script Activity or Update Variable Activity.

Modify the Number of Work Item Fields Limit Per Data Type

You can alter the number of Work Item Fields used for a Repository.

The table SKWorkItemFieldTypeCounts stores the column count for each data type (number, decimal, string, and datetime) supported by Work Item fields. The count for each data type can be varied in the column ColumnCount of this table.

Based on the SKWorkItemFieldTypeCounts table, mapping field gets filled in SKWorkItemFieldsDefinition table when you create a Work Item field from the process designer settings menu.

Consider that you want to increase the string type by one column. That is, currently you can have 18 string Work Item fields and you want to have 19 string Work Item fields. Complete the following steps to increase the string type:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

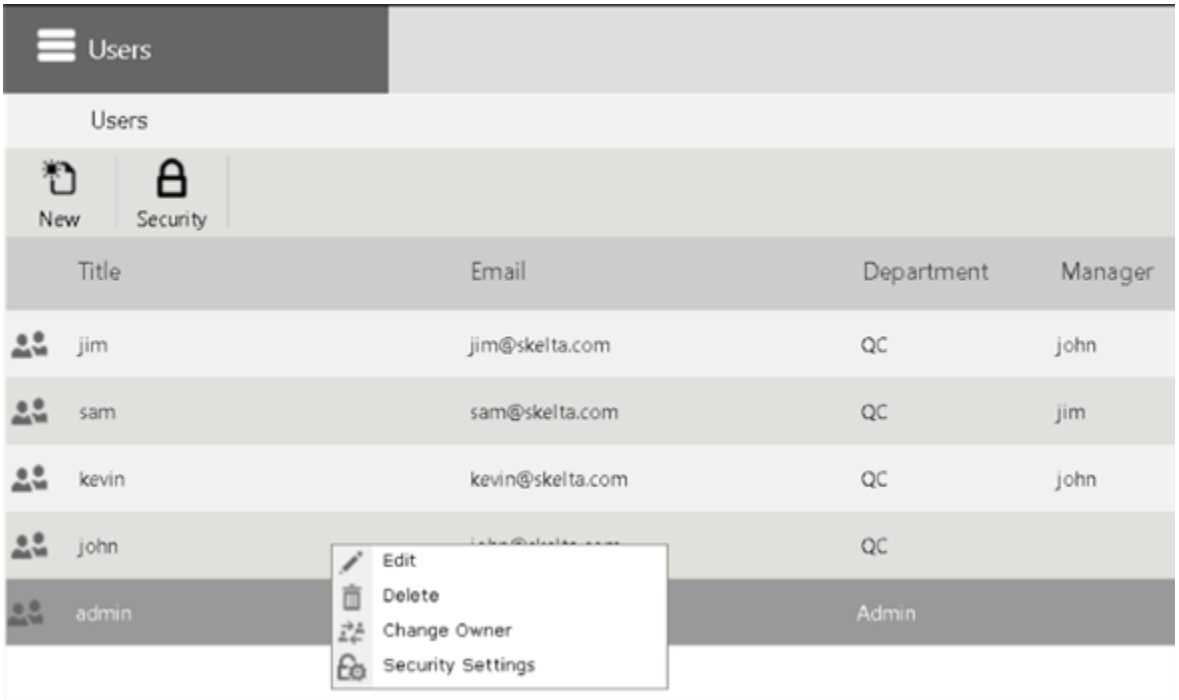
1. In SKWorkItemFieldTypeCounts table, increase the count to 19 from 18 for the string type.
2. Add the column nvarchar19 in the SKWorkItemFields table. SKWorkitemfields table contains all the runtime value details of Work Item fields used in an activity.
3. In the **Process Designer**, create a new string type Work Item field.
4. The created Work Item field can be used in activities by setting it from the Property "Set Workitemfields".

Manager Lookup for Users List

When creating a new user in the List, AVEVA Work Tasks had options for Name, Password, Description, Email, Department, and Role options. Now AVEVA Work Tasks introduced a new option to even select the **Manager** for that new user, where the **Manager Lookup** is a self lookup.

By default, the admin user is created when a new repository is created. For all the newly created user, you can select the existing users as a manager.

The following figure shows that, there are totally four users are existing at repository. Using the Manager lookup, any existing user can be assigned as manager for a user.



Redirect the Log On Page

When you log on to the **Enterprise Console**, by default, the page is redirected to the Workitemlist page.

If you want to redirect the page to another aspx page, then add the following tag needs to the Web.Config file of the Enterprise Console site:

```
<appSettings>
 <add key="redirectPageAfterLogin" value="WorkflowList.aspx"/>
</appSettings>
```

Where,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

value specifies the aspx page to which you want to be redirected after log on. The aspx page must be placed in the following SkeltaBPMUITemplates path: BPMUITemplates\Default\Repository\Site

Value can also be the url of a website.

Since **WorkflowList.aspx** page is specified as value, after log on, the page is redirected to the WorkflowList.aspx page.

You can get the Web.Config file in the following location of the AVEVA Work Tasks install path:

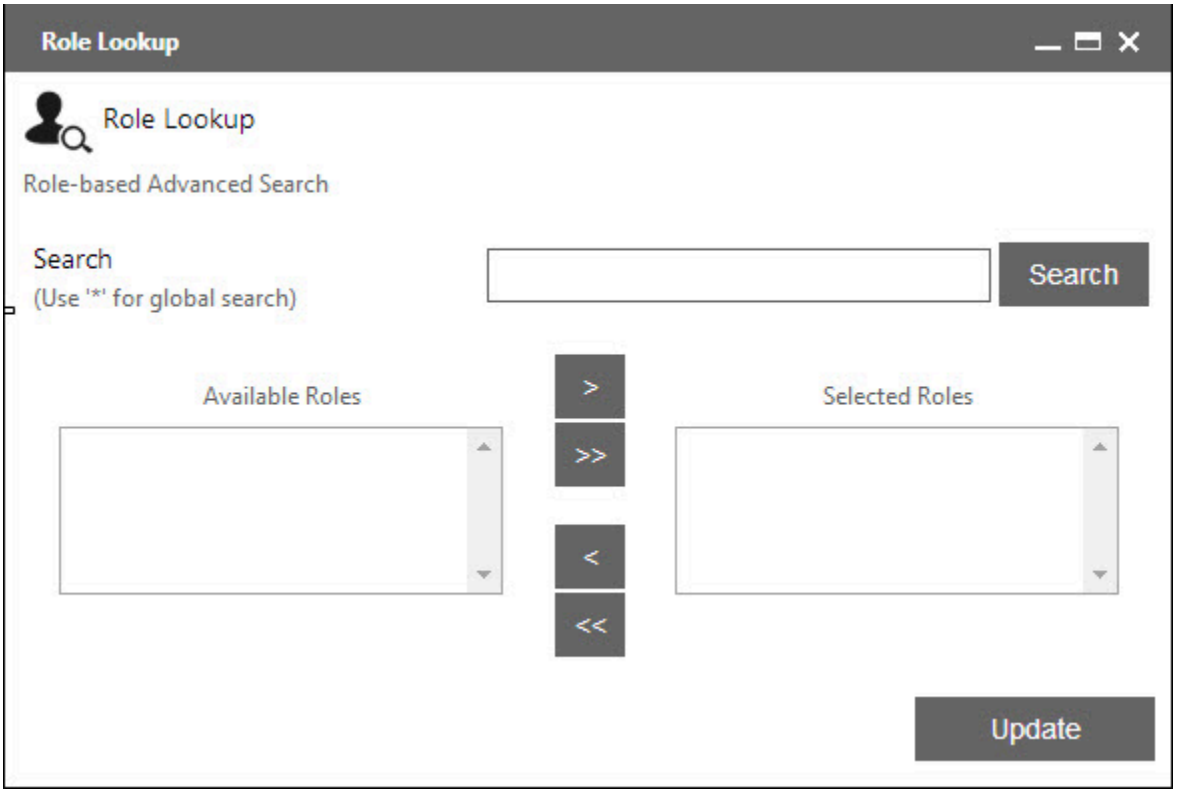
[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\EnterpriseConsole\Web.Config

About Role Lookup

A Role Lookup is used to search a required role from data source providers.

AVEVA Work Tasks provides a **Search** functionality in Role Lookup. This option is available for all data source providers. This enables to search the required role.

You can select the required provider from the provider drop-down.



Use Work Activity

AVEVA Work Tasks provides standard activities, which can be used for creating the workflows. The Work activity is one of the most powerful and flexible among them. The Work activity can be created based on user roles and without roles. The UI to configure the Work activity properties can be displayed in either a Role Based View or a General View. This reduces much of the need for creating custom activities.

The Role-based Work activity is used to provide the flexibility of showing the UI based on the user role. The Work action is configured using XML configuration files and a HTML template to render the UI in the Work Item Detail

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

view.

Work Action with General Functionality (without Role):

- Users will only be able to show the same UI in the Work Item List – based on the AdvanceUI Fields property set for the Work Activity – to all users who log in, without any control over what information they may be allowed to view or may be interested in viewing.
- Users have to select the XML file to be used to specify the configuration in the AdvanceUI Fields property. (The HTML Template file that will be using these configuration settings to render the output is also specified in this XML file.)
- Users also have to specify the possible outputs for the Activity (Action Outputs), Template Placeholders used to render the button value or drop down, and the Output Variables used to store information such as comments in workflow variables, in the AdvanceUI Fields property.

Work Action with Role Based Functionality:

- Users will be able to show a customized UI in the Work Item List – based on the AdvanceUI Fields property set for the Work Activity – to each user who logs in, based on their user role. As a result the UI can be dependent on what information they may be allowed to view or may be interested in viewing.
- Users have to specify in the AdvanceUI Fields property, which XML file is to be used to render the output for each role defined.
- Users also have to specify the possible outputs for the Action (Action Outputs), Template Placeholders used to render the button value or drop down, and the Output Variables used to store information such as comments in workflow variables, in the AdvanceUI Fields property.

The "Approve Reject and Escalate.xml" XML configuration file is shown below

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<?xml version="1.0" encoding="utf-8" ?>
- <WorkTemplate>
  <!-- HTML file name used to show the UI  -->
  <HtmlTemplate>Approve Reject And Escalate.htm</HtmlTemplate>
  <GroupUIHtmlTemplate>GroupUIApprove Reject And Escalate.htm</GroupUIHtmlTemplate>
  <!-- In this ActionOutputs Section we will specify the Possible Outputs for the Action  -->
- <ActionOutputs>
    <ActionOutput name='Approve' CompletionMaker='Yes' Alias='Approve' />
    <ActionOutput name='Reject' CompletionMaker='Yes' Alias='Reject' />
    <ActionOutput name='Escalate' CompletionMaker='Yes' Alias='Escalate' />
  </ActionOutputs>
- <!--
    In this InputVariables section we will be specifying the values for the placeholders used in the html template.
                              The default value can be specified.we can also specify the values for input variables from the
                              AdvanceUI Fields property of Work Action
  -->
- <InputVariables>
    <InputVariable name='approvealias' type='textbox' Desc='Replace Word approvealias with' valueType='Value' Defaultvalue='Approve' />
    <InputVariable name='rejectalias' type='textbox' Desc='Replace Word rejectalias with' valueType='Value' Defaultvalue='Reject' />
    <InputVariable name='escalatealias' type='textbox' Desc='Replace Word escalatealias with' valueType='Value' Defaultvalue='Escalate' />
  </InputVariables>
- <!--
    In this OutputVariables section we will be specifying the value of the HtmlElement to be stored in the
                      Workflow Variable. Name of the Output Variable should be the same as HtmlElement name
  -->
- <OutputVariables>
    <OutputVariable name='txtcomment' desc='Store Comments In' HtmlElementId='txtComment' valuestoset='Variable' operatortype='=' />
  </OutputVariables>
  <htmlelementstopersist>txtcomment</htmlelementstopersist>
</WorkTemplate>
```

Elements and attributes specified in the XML file:

Template Name: The name of the HTML file that will be used by this action to render it in the UI control is specified in the <HtmlTemplate> element.

GroupUIHtmlTemplate: This is not getting used and it is kept for Backward compatibility.

ActionOutPut Variables: The ActionOutput variables that are used to specify all the possible outcomes for the action are specified using the ActionOutput element. The attributes to be specified are described here.

- **Name**: The name of the ActionOutput variable.
- **Completion Maker:** The setting that will determine whether the action is completion maker or not.
- **Alias:** The alias name for the Output that will be displayed as the output when the action is linked in the workflow.

Input Variables: The Input Variables that are used to fill the button values, drop down values, etc., are configured using the InputVariable element. The values for the input variables are associated with workflow variables or with static values. The attributes to be specified are described here.

- **Name**: The name of the input variable that will be used as the placeholder in the HTML template.
- **Type**: The field type that is used to define whether it will be a text box or drop down.
- **Desc**: A short description of the purpose of the Variables.
- **Values**: The values to be associated with the variables; Comma separated values need to be specified in the case of drop down (Yes/No).
- **DefaultValue**: The default value for this variable; it can be static value, variable collection value, and content

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

value.

- **ValueType**: The value type that is used to define whether it will be a variable or content or static value.

Output Variables: The Output Variables that will be used to store the values from various attributes are configured using the OutputVariable elements. The attributes to be specified are described here.

- **Name:** The name of the output variable; it will be the same as the HTML Element Id.
- **Desc:** A short description of the purpose of the Variables.
- **Html Element Id**: The name of the HTML element from which the value has to be read.
- **Valuestoset:** The type to which value has to be stored (Variable/Content).

Note: When you use Input Variables for displaying the button value that will render in the UI, you need to make sure that the Input Variable value and the ActionOutput variable Alias value denoted as Mapped Outputs must be the same.

The "Approve Reject and Escalate.htm" HTML Template file is shown below:

(This template uses the parameters defined in the "Approve Reject and Escalate.xml" file to render the UI to the user.)

The part of the HTML Template where the placeholders are replaced with the input variables configured in the XML file is shown below. This template should have a hidden variable named actionOutPut, for capturing the value inside the control for alerting the action.

```
<link rel="styles" type="text/css" href="<%#VDir%>/StyleSheet/global.css">
<TABLE WIDTH="100%" height="100%" BORDER="0" CELLSPACING="1" CELLPADDING="2"
class="tablebg">
<%#errorMessage%>
<TR VALIGN="TOP">
<TD align="left" width="<%#iframewidth%>" bgcolor="<%#iframebgcolor%>">
<%#frameui%>
</TD>
<TD class="righttdbg">
<TABLE WIDTH="100%" BORDER="0" CELLSPACING="1" CELLPADDING="5">
<TR CLASS="subtitle">
<TD align=left>
@@Comments@@
</TD>
</TR>
<TR>
<TD align=left>
<TEXTAREA name="txtcomment" cols=30 rows=6 wrap="VIRTUAL"
class="inputtextarea"><%#txtcomment%></TEXTAREA>
</TD>
</TR>
<!--<TR>
<TD align="center"> </TD>
</TR>-->
</TABLE>
</TD>
</TR>
<TR class="lefttdbg">
<!--Place holder <%#approvealias%> will be replaced by the input variable approvealias
specified in the Xml Config file -->
<TD align="right" class="lefttdbg" Colspan="2">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<input type="Submit" class="inputbutton" name="btnClickSubmit" value="<%#approvealias%>"
onclick="javascript:BtnClicked(this)">
<!--Place holder <%#rejectalias%> will be replaced by the input variable rejectalias
specified in the Xml Config file -->
<input type="Submit" class="inputbutton" name="btnClickSubmit" value="<%#rejectalias%>"
onclick="javascript:BtnClicked(this)">
<!--Place holder <%#escalatealias%> will be replaced by the input variable escalatealias
specified in the Xml Config file -->
<input type="Submit" class="inputbutton" name="btnClickSubmit" value="<%#escalatealias%>"
onclick="javascript:BtnClicked(this)">
</TD>
<input type="hidden" name="actionOutPut" value="" >
</TR>
</TABLE>
<!--
<%#ExecutingActionName%>
<%#ExecutingActionType%>
<%#ExecutionId%>
<%#ExecutionDetailsId%>
<%#ActivityId%>
<%#WorkflowName%>
<%#ApplicationName%>
<%#ActivityResource%>
<%#OnBehalfOfResource%>
-->
<Script>
function BtnClicked(clickObj)
{
document.getElementsByTagName("input")["actionOutPut"].value = clickObj.value;
}
</Script>
```

Note:The placeholder name used in the template should be the name of the input variables used in the xml file so that the placeholder value will get replaced at run time with the input variable value. In the above example, "approvealias", "rejectalias" and "escalatealias" are the input variables in the Approve Reject and Escalate.xml file.
The function "BtnClicked(clickObj)" is used to capture the output to alert the action with the output.

Steps to create an XML Configuration file for the Work Action:

Consider the example of displaying the UI shown below to the users. For this, you will need to create the XML configuration file and the HTML Template file, say WrkSendBack.xml and WrkSendBack.htm, under the "Workflowelements\Default\en-US\Actions\WorkAction" folder. This section describes the steps to create the XML file.

Specifying the start and end tags:

- The start tag and end tag for the xml file are <WorkTemplate> and </ WorkTemplate> respectively.

Specifying the HTML Template Name:

- The next node to specify is <HtmlTemplate>WrkSendBack.htm</HtmlTemplate>, where WrkSendBack.htm is the HTML Template file corresponding to this XML file. This template will also be created under the Workflowelements\Default\en-US\Actions\WorkAction folder.

Specifying Action Outputs:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The next node is <ActionOutputs></ ActionOutputs>
- Inside this node you need to specify all the possible outputs for this action as shown below
  - <ActionOutput name="Approve" CompletionMaker="Yes" Alias=""></ActionOutput>
  - <ActionOutput name="Reject" CompletionMaker="Yes" Alias=""></ActionOutput>
  - <ActionOutput name="Send Back" CompletionMaker="Yes" Alias=""></ActionOutput>
- You can specify a value in the alias attribute so that it will appear in the AdvanceUI property of Work Action for these action outputs. Alternatively, you can skip setting this attribute here and enter the values in the property.

Specifying Template Placeholders:

- The next node is <InputVariables></ InputVariables>
- Inside this node you need to specify all the input variables that you can use to fill the values in the HTML Template (WrkSendBack.htm) for this action
- For example, let us assume you need to display three buttons in the UI for this work action. The values for these buttons can be filled with the help of Input Variables, which will be used to replace the placeholders in the HTML Template file (WrkSendBack.htm). Inside the node specify the input variables for this action as shown below,
  - `<InputVariable name="approvealias" type="textbox" Desc="Replace Word approvealias with" valueType="Value" Defaultvalue="Approve"></InputVariable>`
  - `<InputVariable name="rejectalias" type="textbox" Desc="Replace Word rejectalias with" valueType="Value" Defaultvalue="Reject"></InputVariable>`
  - `<InputVariable name="sendbackalias" type="textbox" Desc="Replace Word sendbackalias with" valueType="Value" Defaultvalue="Send Back"></InputVariable>`
- You can specify a value in the alias attribute so that it will appear in the AdvanceUI property of Work Action for these action outputs. Alternatively, you can skip setting this attribute here and enter the values in the property.

Specifying Output Variables:

- The next node is <OutputVariables></ OutputVariables>
- Inside this node you need to specify all the Output Variables you will be using to map to workflow variables to store the HTML element values, if required. Specify the variables as shown below.
  - `<OutputVariable name="txtcomment" desc="Store Comments In" HtmlElementId="txtComment" valuestoset="Variable"></OutputVariable>`
- The name of the output variable and the name of the HTML Element Id should be the same.

To create a HTML Template file (WrkSendBack.htm) for the Work Action to render the UI shown in the previous section:

```
<link rel="styles" type="text/css" href="<%#VDir%>/StyleSheet/global.css">
<TABLE WIDTH="100%" Height="100%" BORDER="0" CELLSPACING="1" CELLPADDING="2"
class="lefttdbg">
<%#errorMessage%>
<TR VALIGN="TOP">
<TD align="left" width="<%#iframewidth%>" bgcolor="<%#iframebgcolor%>">
<%#iframeui%>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</TD>
<TD class="righttdbg">
<TABLE WIDTH="100%" BORDER="0" CELLSPACING="1" CELLPADDING="5">
<TR CLASS="subtitle">
<TD align=left>
@@Comments@@
</TD>
</TR>
<TR>
<TD align=left>
<TEXTAREA name="txtcomment" cols=30 rows=6 wrap="VIRTUAL"
class="inputtextarea"><%#txtcomment%></TEXTAREA>
</TD>
</TR>
<TR>
<TD align="center"> </TD>
</TR>
</TABLE>
</TD>
</TR>
<TR class="tablebg">
<!--Place holder <%#approvealias%> will be replaced by the input variable approvealias
specified in the Xml Config file -->
<TD align=right class="lefttdbg" Colspan="2">
<input type="Submit" class="inputbutton" name="btnClickSubmit" value="<%#approvealias%>"
onclick="javascript:BtnClicked(this)">
<!--Place holder <%#rejectalias%> will be replaced by the input variable rejectalias
specified in the Xml Config file -->
<input type="Submit" class="inputbutton" name="btnClickSubmit" value="<%#rejectalias%>"
onclick="javascript:BtnClicked(this)">
<!--Place holder <%#sendbackalias%> will be replaced by the input variable sendbackalias
specified in the Xml Config file -->
<input type="Submit" class="inputbutton" name="btnClickSubmit" value="<%#sendbackalias%>"
onclick="javascript:BtnClicked(this)">
</TD>
<!-- Hidden variable used to handle for Alerting the Output.The name actionOutPut should be
same-->
<input type="hidden" name="actionOutPut" value="" >
</TR>
</Table>
<!--
<%#ExecutingActionName%>
<%#ExecutingActionType%>
<%#ExecutionId%>
<%#ExecutionDetailsId%>
<%#ActivityId%>
<%#WorkflowName%>
<%#ApplicationName%>
<%#ActivityResource%>
<%#OnBehalfOfResource%>
-->
<Script>
<!-- Script used for the logic of assigning the output to the hidden variable actionOutPut
Which will be used by the action to alert the output.-->
function BtnClicked(clickObj)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
document.getElementsByTagName("input")["actionOutPut"].value = clickObj.value;
}
</script>
```

Note: The <%#Acknowledge%> placeholder is used to display the Read Acknowledgement.
The <%#Notes%> placeholder is used to display the notes if you specify in the action. The <IFRAME> tag is used to display the details page inside an IFrame. The <TEXTAREA> tag is used to show the comments. The "<%#approvealias%>", "<%#rejectalias%>" and "<%#sendbackalias%>" placeholders are used as the input variables in the WrkSendBack.htm file. The "actionOutPut" hidden variable is used to store the output to alert the action with. The function "BtnClicked(clickObj)" is used to capture the output to alert the action with.

To use the created xml file (WrkSendBack.xml) and HTML template (WrkSendBack.htm) in the Work Action:

1. Click the **AdvanceUI Fields** property in the **Work Activity Actions** Properties.
2. You will see the file name in the **File Name** drop down (of both General View and Role Based View).
3. Select this file name and then click **Action Outputs** or **Template Placeholders** or **Output Variables** to display the corresponding property screen; you can see the parameters configured in the xml file displayed in the property screen.
4. Enter the values in the respective property screen; then save and deploy the workflow.
5. After you execute the workflow you will find the ActivityUI rendered in the Work Item detailed window.

Auto Mapping of Active Directory Providers

If more than one Active Directory provider is mapped to one Repository then by default, mapping will happen between the active directory providers based on the **Display Name** property. If user does not want the mapping, then uncheck the Auto Mapping flag from Repository Provider Search page for the User Entity.

By default, the entry in the **Properties** column of the **SKRepository** table

<property key="AutoMapActiveDirectoryProviders"><![CDATA[True]]></property>

Steps for UnMapping

If user does not want Auto Mapping between providers, do the following:

1. Log on to the **Central Configuration**.
2. Click **Repository-Provider List**. The **Repository-Provider List** page is displayed with the list of repositories name.
3. Expand the required repository where you do not want AutoMapping between Active Directory providers.
4. Expand **UserEntity**.
5. Click the **DefinitionXML** link to open the DefinitionXML file. Deselect the **Do you want to map active directory providers** check box at first time, and do the respective operations.
6. Go to the Farm Database. Now the entry in the **Properties** column of the **SKRepository** table is

   ```
   <property key="AutoMapActiveDirectoryProviders"><![CDATA[False]]></property>
   ```

Steps for Auto Mapping

1. Log on to the **Central Configuration**.
2. Click **Repository-Provider List**. The **Repository-Provider List** page is displayed with the list of repositories

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

name.

3. Expand the required repository and UserEntity where you want AutoMapping between Active Directory providers.
4. Click the **DefinitionXML** link to open the DefinitionXML file. Select the **Do you want to map active directory providers** check box.
5. Check the **Edit the definition xml**. The **Update** button will be enabled.
6. Click **Update**.
7. Select the respective provider, and then click **DefinitionXML** link to open the DefinitionXML file.
8. Check the **Edit the definition xml**. The Update button will be enabled.
9. Clicking **Update** will map the across the active directory providers.

Notes:

- If user does not want Auto Mapping between providers, deselect the "Do you want to map active directory providers?" check box for first time and do the respective operations similar to before updating provider DefinitionXML.

- Once it is mapped, unchecking the option will not remove the existing mapping.

Customize the Log On Page

You can customize the log in page of AVEVA Work Tasks.

You can have your own log on page instead of AVEVA Work Tasks log on page. You can also use AVEVA Work Tasks product to customize your application.

Note: The code provided below is a sample for reference. You can design the Log On page according to your specific requirements.

To customize the log on page

1. Go to *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**BPMUITemplates**Default**Repository**Site* folder.
2. Create your custom allows you to customize the login page.

```
Open RepositoryList.aspx page. In that, under the  javascript function named
"SetRepositoryValues (provtext, repo, instanceName, providerName,repositoryGuid)" you
can find the below code. This is for redirecting to default log on page [Login.aspx].
var winObj = window.open('Login.aspx?_repo='+repo+' &_instanceName='+instanceName+
'&_provtext='+provtext+'&_prov='+providerName,winName,
'top=0,left=0,width='+strWidth+',height='+strHeight+',
location=0, status=1, scrollbars=0, resizable=yes');
```

   To redirect to your custom login page, update the above code by replacing Login.aspx with your custom login page name and put your login page in the *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**BPM.NET**Web**BPMUITemplates**Default**Repository**Site* folder.

   Example: If the Custom Login Page name is HRLogin.aspx, then the code will be updated as shown below.

```
var winObj = window.open('HRLogin.aspx?_ repo= '+repo+'
&_instanceName='+instanceName+'
&_provtext='+provtext+'&_prov='+providerName,winName,'top=0,left=0,width='+strWidth+'
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
   ,height='+strHeight+',location=0, status=1, scrollbars=0, resizable=yes');
```

3. Add the below code in the CS [codebehind] file of the custom login page to authenticate the user and to redirect to your home page. Add SE.Bpm.Convertors namespace in the 'Using' section of the codebehind file.

   You have to provide:

   a. Page you want to redirect (in this case it is "HRHome.aspx").

   b. Username and Password of the user (make sure the user is available in your data source).

```
      //On click of label HR, this event will occur
      protected void lbSiteAdmin_Click(object sender, EventArgs e)
      {
      // To authenticate the specific User.
      // The function takes in Username and password as parameters.
      this.CreateUserContext("HRadmin", "admin");//Username and Password values are
      //hardcoded here,you can pass these values from the login page

      string redirectPageAfterLogin = "HRHome.aspx";//specify the home page
      Response.Write("<script language='Javascript'> location.href = '" +
      redirectPageAfterLogin + "';</script>");
      }

      private void CreateUserContext(string username, string password)
      {
      string applicationname, providerinstancename, providerinstancedisplayname,
      provider = "";

      applicationname = Convert.ToString(Request.QueryString["_repo"]);
      providerinstancename = Convert.ToString(Request.QueryString["_instanceName"]);
      providerinstancedisplayname =
      Convert.ToString(Request.QueryString["_provtext"]);
      provider = Convert.ToString(Request.QueryString["_prov"]);
      Response.Cookies["_instanceName"].Value =
      HttpUtility.UrlEncode(providerinstancename).Encrypt(15);
      Response.Cookies["_repo"].Value =
      HttpUtility.UrlEncode(applicationname).Encrypt(12);
      Response.Cookies["_provtext"].Value =
      HttpUtility.UrlEncode(providerinstancedisplayname).Encrypt(14);
      Response.Cookies["_prov"].Value = HttpUtility.UrlEncode(provider).Encrypt(13);
      Response.Cookies["_userinfo"].Value =
      HttpUtility.UrlEncode(username).Encrypt(11);

      Skelta.Core.ApplicationObject application = new
      Skelta.Core.ApplicationObject(applicationname);
      Skelta.Entity.Entity userEntity = Skelta.Entity.Entity.GetEntity(application,
      "UserEntity");
      object userUniqueIdentifier =
      userEntity.DataBridge.DataBridgeProviders[providerinstancename].GetUniqueIdent
      ifier(username, password);
      //Getting user context
      Skelta.Entity.UserContext uContext = new
      Skelta.Entity.UserContext(userUniqueIdentifier,
      application, providerinstancename, "", "", false);
      }
```

4. After making changes to the pages, now login to the application [From EC site]. This will redirect to your login page. For more information on creating custom page and redirecting to your login page, refer to Create

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Custom Page.

5. Log on to your custom login page, this will redirect to the page which you had specified. In the above sample, when you click the label named HR, it will redirect to HRLogin.aspx.

Packages

You can use the **Package** feature of AVEVA Work Tasks to deploy one or more artifacts from one environment or repository to another environment or repository.

About Package

You can use the **Package** feature of AVEVA Work Tasks to deploy one or more artifacts from one environment or repository to another environment or repository. You can deploy artifacts with its associated and child artifacts and configuration state from one environment or repository to another environment or repository.

Prerequisites

- Ensure that the AVEVA Work Tasks Client Service is running while installing or uninstalling the package.
- In an environment that contains multiple servers and clients, ensure AVEVA Work Tasks Client Service is running on the machine which has the Enterprise Console site configured.

Advantages of Package

Currently, packaging and deploying one or many artifacts from one environment or repository to another is done either by exporting and importing the artifacts or by using the recovery of the task for packaging and deploying one or many artifacts from one environment or repository to another.

Export/Import feature in Package

- Export and import feature is helpful when you want to deploy a single artifact in a different environment.
- You can export and import only one artifact at a time.
- An imported artifact is not valid on the target environment, if the associated artifacts are not present.

Recovery feature in Package

You can use the recovery feature to replicate the entire repository with all the artifacts in the target environment. A choice to add only few artifacts to the target environment is not available. This option is beneficial when you want to deploy the entire repository in another environment.

A proper understanding of the **Package** interface enables you to use the various features of the interface effectively. The **Package** interface consists of the following two components:

- Package Template: Use **Package Template** to design a package, define artifacts to the template, and complete the entire process of design, build, and export the package file. When you export a packaged file, the file is saved with the extension .**awsPKG**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Manage Package: Use **Manage Package** to import packages which you exported from the **Package Template** and install the package in the target environment. You can also create a new package and export it.

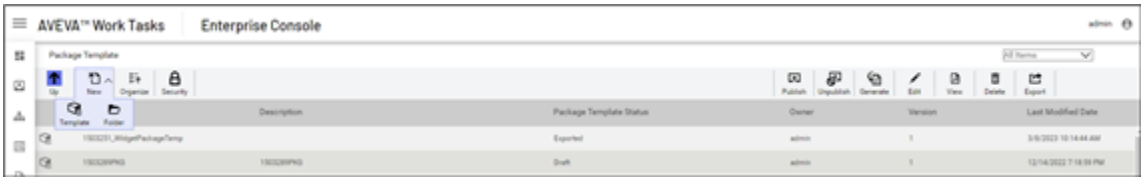
Note:
- Only Administrators have the right to install the package.
- Access to the functionalities of **Package Template** and **Manage Package** requires special user rights. The functionality might either be disabled or might not be visible if the user does not have rights to run them.
- You must reset the IIS before generating a package for the first time after installing the product.

Package Template

You can use the **Package Template** list in the source environment to create and design a package template to which you can add artifacts and update the configuration settings.

You can use the **Package Template** to complete the entire process of designing, publishing, generating, and exporting the package file. The behavior of the package at the time of installation can be pre-defined in a **Package Template**.

The following image shows the **Package Template** window:

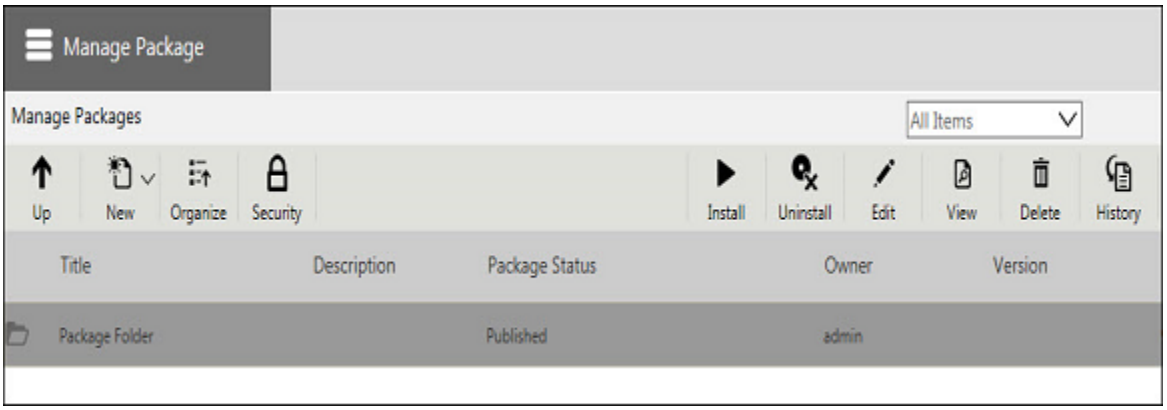


See Also: Working with Package Template

Manage Package

You can use the **Manage Package** list in the target environment to import packages that you exported from the **Package Template** of source environment, and install, update, or uninstall the package in the target environment. You can also change the pre-defined configuration that affects the behavior of package at the time of installation, update, and uninstallation.

The following image shows the **Manage Package** window:



See Also: Manage Package

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Best Practices of Packaging

The following list provides some of the best practices or guidelines which you can adhere to while packaging.

- You can create and export a package in Wonderware Skelta BPM 2014 R2 SP2 P01 and later versions and use this package in AVEVA Work Tasks.
- Before adding artifacts to the package, ensure all the artifacts that need to be added to a package are in the *Published* state.
- Identify the artifact which has the complete relationship for the project to be packaged.
  - If a navigation list item points to the main form (a form which is associated with all the other artifacts), go to the Navigation page and then add the navigation to the package.
  - If there are multiple main forms that do not have a relation among each other, add each of these forms individually to the package template. Ensure that the package template settings are same while adding each form to the package.
  - If the main form is not used in navigation, go to the Forms List page and then add the main form to the package.
  - If there is only a main workflow, then go to the Workflow List page and then add the workflow to the package.
- If you have created a project folder for forms, workflow, or any other lists, which are meant to be packaged together, then add the folder to the package for easier tracking.
- If a package is already shipped and installed, and if you add or modify some artifacts of the package, then you must create another version of the same package template and then add or modify the artifacts in the newly created version of the package template.

  Note: Custom files and scripts will not be identified as modified items.
- You can configure the common parameters for different artifacts in the Runtime Initialization Settings form before installing the package. For more information about runtime initialization settings, see Runtime Initialization Settings section in the *Developer Guide*.
- Ensure that the Runtime Initialization Settings form is a single form which does not have any association with other forms. It is recommended that you configure the artifacts (such as Forms, Workflow, Database Connection, and so on) as different tabs in the Runtime Initialization Settings form for easy configuration.
- Ensure that you install a package only during off-production.
- If you use a case-sensitive database to create a repository, you can create items with the same name in different cases. For example, a case-sensitive database considers DATABASE1 and database1 as two different items. In this case, the package will be imported as two different items. If the database is not case-sensitive, you cannot create an item with the same name.

  We recommend you not to create two different items with the same name in a case-sensitive database. If you package and import such items to a not case-sensitive database, the package gets imported as the next major version.
- Before packaging, verify whether all the custom dependent files, which are used, are identified and added to the package. If any of the custom dependent files are not added, it is recommended that you add these files to the **Additional Package Files** section in the **Package Template** tree.
- Ensure that all the custom files used are placed in the installed Skelta path.

  For example: *C:\Program files\AVEVA\Work Tasks\Bin*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:
- The packages created in the higher versions may not work properly in the earlier versions of the product.
- If the artifacts using the custom dlls in the package are installed in the latest version of the product, make sure that the custom dlls are built with the latest version before installing the package.

Recommendations

- The appropriate target platform is used while installing packages.

  For example: If the package is created in 64-Bit, it is recommended that you install the package in 64-Bit.
- The prerequisite packages are installed before installing the required package.
- The different versions of a specific package template should use the same Runtime Initialization form.
- If you face any issues or errors while adding artifacts to the package, it is recommended that you open the **WTLogger Console** so that you can view the error messages or warnings.
- **Scenario 1:** A package that has many artifacts is already shipped and installed. Of the many artifacts, you modify only few artifacts. However, you do not want to modify the existing package template. In this scenario, it is recommended that you package and ship only the modified artifacts in a new package.
- **Scenario 2:** A package that has many artifacts is already shipped and installed. If there are changes in physical files and/or script files in the package, then you must package and ship these artifacts in a new package template.
- **Scenario 3:** A package has a workflow with many activities. If there is any modification in the workflow definition, then it is recommended to use the “Complete Rebuild” feature to obtain the latest definition of all activities. For more information about complete rebuild, refer the Complete Rebuild section.
- While adding artifacts to a package template, if some of the artifacts are not added due to some exceptions, perform one of the following:
  - Verify the exception, correct it, and then add the artifacts to the package template.
  - Verify the exception, save the package, correct the exception, and then re-add the artifacts (which have exceptions) to the same package template.

    Note: You can validate the artifacts which have exceptions. You can also proceed with the installation of the existing artifacts by saving the package template.

Test Statistics

The following table represents the test statistics for the number of artifacts, which were packaged in a single package.

Type of Artifact Packaged	Number Of Artifacts
• Workflows: 72 • Forms: 92 • Manage List: 29 • DataGrid: 33 • Navigation List: 23	323

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Type of Artifact Packaged	Number Of Artifacts
• Custom Files: 20 • Lookups: 54	
• Workflows: 120 • Forms: 124 • Manage List: 36 • DataGrid: 51 • Navigation List: 23 • Custom Files: 21 • Lookups: 120	495
• Workflows: 107 • Forms: 33 • Manage List:11 • DataGrid: 50 • Navigation List: 88 • Custom Files:30 • Lookups: 84 • Security GroupList: 616 • Report Template: 9	1028

If there are many artifacts in a package, it is recommended to split this package into multiple small packages.

Work with Package Template

You can use the Package Template list in the source environment to create and design a package template to which you can add artifacts and update the configuration settings.

You can use the Package Template list to perform the following tasks that helps you create, design, and export a package template file:

- Create package template
- Edit package template
- Publish package template
- Generate package template
- Export package template
- View package template
- Unpublish package template

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Mark package template as shipped

Actions Executable in the Ribbon Menu

The following table lists the actions that you can perform from the action bar of the Package Template list:

Actions		Description
Up		To navigate from the current folder to the parent folder. This feature is helpful when you are accessing a folder within the Package Templates and you want to navigate back to the parent folder of the Package Template List.
New	Folder	To group and categorize the package templates.
	Template	To create a new package template.
Organize		To organize the Package Templates and group it in specific folders.
Security		To define security at the List Item level.
Edit		To modify or delete the artifacts, define the target values, configuration state, and the prerequisites.
View		To view the package template in read-only mode.
Delete		To delete the selected package template.
Publish		To publish the selected package template.
Unpublish		To unpublish the selected package template.
Generate		To generate the package file. The generated package file has the .awsPKG extension.
Export		To export the packaged file to a specific folder.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Actions Executable in the Shortcut Menu

The following table lists the actions that you can perform from the shortcut menu of the Package Template list:

Shortcut menu options	Description
Edit	To modify or delete the artifacts, define the target values, configuration state, and the prerequisites.
View	To view the details of the selected package template in read only mode.
Publish	To publish a package template that is in draft status.
Unpublish	To unpublish a package template that is in published, generated, or exported status.
Generate	To generate a published package template.
Export	To export the package template file.
Mark as Shipped	To mark the package template as shipped once it is exported.
Save As	To save the package template in different versions. By default, when a package template is created it is saved as version 1.
Delete	To delete the selected package template. You can delete the package template irrespective of the status.
Show All versions	To view all the versions. This option is useful as the package template list displays only the latest published version of the package template or only the latest version of a draft when all the versions of the package template are in draft status.
Security Settings	To define security at the List Item level.

Note: Package template created in previous version of AVEVA Work Tasks must be regenerated.

Create a Package Template

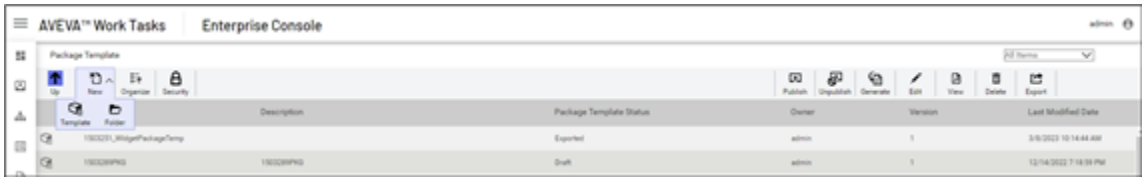
1. In the **Enterprise Console**, click menu icon, click **Package**, and then click **Package Template**.
   The **Package Template** page appears.
2. Click **New** and then, click **Template**.
   The **New Package Template** dialog box appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. In the **Title** box, type a name for the package template.
4. In the **Description** box, type a description for the package template.
5. In the **Additional Information** box, type any additional information about the package template. By default, the name of the current repository is displayed.
6. Click **Save Template**.

   A confirmation message appears.
7. Click **Close**.

   The package template is saved in the Package Template list. You can add artifacts to the package templates created. The following image shows the **Package Template** page with a few package template lists:



Note: If a package template contains the Runtime Initialization Form, then this Form will not be available when you recreate the package template in the installed machine. You have to manually add the Form to the package template. To do this, include the Runtime Initialization Form along with the original package. After creating the package template from the installed package, edit the package and select the Runtime Initialization Form.

See Also: Adding Artifacts to a Package Template

Edit a Package Template

You can edit a package template to modify the prerequisites, update the runtime initialization settings, delete artifacts, and modify the configuration of the artifacts. The package template that you want to edit must be in **Draft** state.

1. In the **Enterprise Console**, click menu icon, click **Package**, and then click **Package Template**.

   The **Package Template** page appears.
2. Select a package template that you want to edit, and then click **Edit**.

   Alternatively, right-click a package template, and then click **Edit**.

   The **Edit Package Template** dialog box appears.
3. Edit the package details.
4. Click **Save**.

   The package template reflects the changes.

Configure Package Template

The **Add to Package Template** page or the **Edit Package Template** page allows you to perform a number of actions such as, defining the prerequisites at the package template level, runtime initialization settings, and configuring properties specific to individual artifacts. In the **Add to PackageTemplate** page, you can also view the Associated and Child Artifacts.

The main components of the Add to Package Template/Edit Package Template screen are:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Prerequisites
- Runtime Initialization Settings
- Exceptions

Artifact level configurations are:

- Configuring the Artifacts
- Rebuilding the Artifacts
- Deleting the Artifacts

Package Template Level Configuration Settings

This section describes the configuration settings of a package template, such as but not limited to prerequisites, runtime initialization settings, and package files.

Prerequisites

The Prerequisite option in the **Add to Package Template** page allows you to define the prerequisites for the Package Template. If the Package Template is dependent on other Package Templates, then the dependent Package Templates can be set as prerequisites.

For example: If the 'Leave' package is dependent on the 'HR' package, then the 'HR' package can be set as a prerequisite for the installation of the 'Leave' package.

You can modify prerequisites by editing the Package Template before it is published.

To set prerequisites

1. Select and right-click the artifact that you want to package, and then click **Add to Package Template** on the shortcut menu. The **Select Package Template** page appears.
2. Select the Package Template.
3. Select **Package Template Version**, **Include Associated Artifacts**, **Include Child Artifacts**, **Include Security Artifacts**, and **Include the automatically generated SQL Script**.
4. Click **Next**. The **Add to Package Template** page appears.
5. In the **Configuration** section, click **Prerequisite**. The **Prerequisite** page appears on the right pane.
6. From the **Package Template** list, select the package template that you want to set as the prerequisite.
7. Select the version of the package template from the **Package Template Version** list. The version defaults to 1.
8. Select the **Consider Higher Versions if selected version does not exist in target environment** option. This will ensure that the higher versions of the prerequisite package template are considered while installing the package in the target environment, if the selected version is not found.

   If this option is not selected, then while installing the package in the target environment only the exact version of the prerequisite will be considered. If the selected version does not exist, the package will not be installed.
9. Click **Add Template**.
10. Repeat Steps 6-9, to add more package templates as prerequisites.

    The grid in the **Package Template** page lists out all the Package Templates which are set as prerequisites.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

11. Click **Save Settings**.

    The **Prerequisite Package Templates** grid contains the following details:

    - **Name**: This column displays the name of the Package Template that is set as a prerequisite.
    - **Version**: This column displays the version number of the prerequisite package template.
    - **Higher Versions**: This column displays either Yes or No.
        - **Yes** appears if the **Consider Higher Version** check box is selected.
        - **No** appears if the **Consider Higher Version** check box is not selected.

    Note: The **Prerequisite Package Templates** grid also appears when you select the following options:
    - **Enterprise Console > Package > Manage Package > Edit > Prerequisite**
    - **Enterprise Console > Package > Manage Package > View > Prerequisite**
    - **Enterprise Console > Package > Manage Package > Install**

12. Click **Save Template**.

    The package template settings are saved.

To delete a prerequisite package template

Perform the following to delete a package template that is set as a prerequisite.

- From the **Prerequisite Package Templates** grid in the **Prerequisite** page, click the icon in the row corresponding to the package template that you want to delete.

  The corresponding prerequisite package template is deleted successfully.

Runtime Initialization Settings

The **Runtime Initialization Settings** option in the **Add to Package Template** allows you to select a form, map the fields of a form to the properties of the artifacts. The values mapped to the artifacts are populated when the package is installed. This feature is useful when the same data has to be entered in multiple places. The following section allows you to create a form, fill a form, and map the values of the form to the properties of the artifacts.

Note: Runtime initialization is not supported for the following artifacts:
- All Event Providers such as Event Associations and Scheduler
- Report Connection, Report Templates, and Reports
- All layouts such as Global Work Item Layout and Custom Layout
- List Views
- Navigation Lists

The following sample topics provide the procedures to use the **Runtime Initialization Settings** form values in the Database Connection properties. Similarly, you can configure the runtime initialization settings for other artifacts.

Adding a database connection to package

For more information about creating a database connection and adding a database connection to a package, refer to the Creating a Database Connection and Adding Database Connection to Package Template sections respectively.

To create a Form

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. In the **Name** box, type the name for the form as **Database Connection Details**.
4. In the **Description** box, type the description for the form as **Database Connection Details**.
5. Click **Save & Continue**.

   The **Form Tasks** page appears.
6. Click **Design this form** and then click **Finish**. The **Forms Designer** page appears.
7. Drag the text input control to the canvas.
8. Edit the properties of the control and change the Name to **Server Name**.
9. Click **Apply** to save the properties.
10. Repeat steps 7 to 9 to create the following fields:
    - **Authentication Type**
    - **User Id**
    - **Password**
    - **Advanced Settings**
    - **Database Name**

    For more information about creating and designing forms, refer the *Forms* and *Forms Designer* sections in the *Forms and Lists User Guide*.

    Note: If the **Package Settings** form has a **Password** field, then its value can be set from the **Runtime Initialization Settings** by selecting one of the available password controls. Ensure to select a password controls whose **Security Mode** property is set to **Encryption** and **Persist Password** property is set to **Yes**.
11. Click **Publish** from the action bar.
12. Click **Close** on the BellyBar.
13. Close the **Forms Designer** page.

Note: Form designed for runtime initialization settings support only the text input, number, drop-down, hyperlink and password controls in Package.

To configure Runtime Initialization Settings

1. To add a DB connection to package, in the **Enterprise Console**, click menu icon, click **Settings**, and then click **Database Connection**. Right-click any existing database connection and then click **Add to Package**. The **Add to Package Template** dialog box appears.

   OR

   To edit an existing package, in the **Enterprise Console**, click menu icon, click **Package**, and then click **Package Template**. The **Package Template** page appears.

   Select the package template for which you want to configure the **Runtime Initialization Settings**, and then click **Edit**. The **Edit Package Template** dialog box appears.
2. Under **Configuration** tree on the left pane, click **Runtime Initialization Settings**. The **Runtime Initialization Settings** form appears.
3. In the **Form Name** list, select the **Database Connection Details** form.
4. In the **Form Version** list, select the version of the form that you want to use. By default, the **Form Version**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

takes the latest version.

5. Click **Next**. The selected form with all the fields appear.
6. Fill the selected form and then click **Finish**.

Note: You can fill the form while installing the package as well.

To map the Runtime Initialization Settings to the properties of the artifacts

The following steps allow you to map the properties of a Database Connection to the Runtime Initialization Settings.

1. Under **Configuration** tree on the left pane, select a Database Connection listed under the **Artifacts** section.
2. Click the **Install** tab**.**
3. Under **Connection Details > Target** section, click **No** for the property **Connect To Repository DataSource**.
4. Select the value for **Server name**.
   - In this example, the **Server Name** property can be mapped to the Server Name box of the *Database Connection Details*. When the package is installed the value of the Server Name property will be populated with the value entered in the *Database Connection Details* form selected in *Runtime Initialization Settings*.
   - Multiple selection from the list will append the value in the property.

   Note:
   - You can select the values by typing and/or selecting from the drop-down list. All the fields of *Database Connection Details form* are available for selection in the drop-down list. The value can be a combination of values entered manually and values selected from the drop-down list.

   - If a value is already available, and if you select another value from the drop-down list to be added to this value, the selected value is always appended at the end of the existing value.

   - If you enter or select a value, then the drop-down icon is changed to **+** icon. Click the **+** icon to select another value from the drop-down list. The selected value is always appended at the end of the existing value.

   - It is recommended that you do not edit the text within *<% Control Name %>*, where "Control Name" is the name of a control available in the **Runtime Initialization Settings** form. If you edit the text, then ensure that the "Control Name" is available in the **Runtime Initialization Settings** form.

   - You can select the values from the drop-down list only if the **Runtime Initialization Settings** are configured.

   - If you select another form in the **Runtime Initialization Settings**, the value is not cleared. You have to manually change the value.
5. Similarly map all the other properties (Authentication type, User ID, Advanced settings, and Database name) of the artifacts to the appropriate fields of the **Database Connection Details** form.

All the values entered in the **Runtime Initialization Settings** form will be populated to the properties of the artifacts when the package is installed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Exceptions

Exceptions are visible only if any exceptions are logged while adding artifact to the Package Template.

Additional Package Files

The **Additional Package Files** option allows you to add additional custom files and installer files to be installed in the target environment This option is available in the **Add to Package Template** and/or **Edit Package Template** pages.

To add Additional Package Files and Installer Files

1. Select and right-click the artifact that you want to package, and then click **Add to Package Template** on the shortcut menu. The **Select Package Template** page appears.
2. Select the Package Template.
3. Select **Package Template Version**, **Include Associated Artifacts**, **Include Child Artifacts**, **Include Security Artifacts**, and **Include the automatically generated SQL Script**.
4. Click **Next**. The **Add to Package Template** page appears.
5. In the **Configuration** section, click **Additional Package Files**.

   The **Additional Package Files** page appears on the right pane.
6. Select the option **Disable adding custom files to the package** as **Yes**, if you do not want to include the individual artifact's custom dependent files and additional files while generating the package.

   The custom dependent files and additional files will be visible in the Settings page. But these files will not be included while generating the package.
7. Click **+** icon to add and configure the additional package files and the installer files.

   Note: Configuring additional package files is optional. You need not configure additional package files prior to configuring the Installer Files.

   a. To configure the additional package files, enter the following details:
      a. **Source Files**: Enter the location of the package file.
      b. **Install in Global Assembly Cache**: Select **Yes** to install custom dependent files in the GAC of the target environment.

   b. To configure the installer files, enter the following details:
      a. **Installer File Path**: Enter the location of the installer file.
      b. **Command Line Parameters**: Enter the command line parameters for the installer file.
      c. **Disable Execution**: Select the check box to skip execution of the Installer File. (This option is available only in Manage Package feature.)

   For more information on configuring the Installer Files, see Configure Settings for Installer Files.
8. Click **Save Settings**.
9. Click **Save Template**.

   The package template settings are saved.

   Note: Ensure that the File Path specified is accessible to IIS_IUSRS.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure Settings for Installer Files

The **Installer Files** has an unique set of configuration settings required to install, update, and uninstall the installer files.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window (can view and edit the configuration settings)
- **Target Environment > Manage Package** window (can only view the configuration settings)

You can define the configuration settings of the Installer Files and set the properties to take appropriate actions while installing, updating, or uninstalling the package.

You can also set the execution timeout for the Installer Files. The execution timeout can be set between 120 to 100000 seconds. The timeout set as 0 second means no timeout is set.

During package installation, if one of the installer file execution is timed out, the package installation stops. To install the package again, set the appropriate timeout to execute the timed out installer files.

As a best practice, each installer file execution should complete within 120 seconds.

During install or update of the package,

- the installer files configured are executed prior to the installation of the artifacts in the package. However, during uninstallation, the installer files are removed after the artifacts are removed/uninstalled.
- if any of the installer file execution times out, the package installation or update stops. Also the subsequent installer file is not executed and the artifacts are not installed or updated.

During uninstall, the installer files are removed after the artifacts are uninstalled/removed and if any of the installer file execution times out, the subsequent installer files are not removed.

During package template generation, the installer file certificate is validated online. So, make sure that the internet connectivity is available on the Server where package template is generated.

Note:
- The installer files which were already executed will not be rolled back, even if the next installer file execution exceeds timeout.
- The timed out installer files will be still running in the background. Make sure that the execution of the timed out installer files is completed before importing or executing any other package.
- The package can be generated only if it contains the installer files signed with valid and trusted certificate.
- Make sure that the firewall does not block the Online Certificate Status Protocol (OCSP) request. The OCSP is used for obtaining the revocation status of the digital certificate.
- For Load Balanced Server, make sure all the servers are online while installing the installer files.

Sample of the Installer File:

*msiexec.exe /i "C:\Example.msi" /qn /norestart /L*V "C:\package.log"* where,

- */i* is install option (possible options are /a for install, /x for uninstall, and /update for update)
- *%INSTALLERFILEPATH% /qn /norestart* gets added during msi execution (*%INSTALLERFILEPATH%* is the place holder for installer file location)
- */L*V "C:\package.log"* is optional

You need to enter *"C:\Example.msi* in the **Installer File Path** text box and */i %INSTALLERFILEPATH% /qn /norestart*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

gets populated in the **Command Line Parameters** text box. You can further enter */L*V "C:\package.log"* (optional) in the **Command Line Parameters** text box

Note:
- In the **Command Line Parameters** text box, */i %INSTALLERFILEPATH% /qn /norestart* is the default parameter and is automatically taken by the system even if deleted. You cannot overwrite it. The *%INSTALLERFILEPATH%* is the place holder for the location of the Installer File.
- If the Installer File is placed on the your machine desktop, the path location will not be fetched and the package cannot be generated.
- You can add only 5 rows for the Installer Files under each tabs (Install, Update, and Uninstall).

Install

Use this tab to set the configurations for Installer Files while installing the package.

- **Installer File Path**: Enter the location of the installer file. For example, *C:\Example.msi*
- **Command Line Parameters**: Enter the command line parameters for the installer file. The default parameter is */i %INSTALLERFILEPATH% /qn /norestart*
- **Disable Execution**: Select the check box to skip execution of the installer files. By default, it is unchecked.

Update

Use this tab to set the configurations for Installer Files while updating the package.

- **Installer File Path**: Enter the location of the installer file. For example, *C:\Example.msi*
- **Command Line Parameters**: Enter the command line parameters for the installer file. The default parameter is */i %INSTALLERFILEPATH% /qn /norestart REINSTALLMODE=amus*
- **Disable Execution**: Select the check box to skip execution of the installer files. By default, it is unchecked.

Uninstall

Use this tab to set the configuration of the Installer File when the package is being uninstalled from the target environment.

- **Installer File Path**: Enter the location of the installer file.
- **Command Line Parameters**: Enter the command line parameters for the installer file. The default parameter is */x %INSTALLERFILEPATH% /qn /norestart*
- **Disable Execution**: Select the check box to skip execution of the installer files. By default, it is unchecked.

The table shows the status of the package before and after the execution of the package in case of timed out execution.

Operation	**Package Status**	
Before Execution	**After Execution**	
Install	Not installed	Not installed

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Operation	Package Status	
	Installed	Installed
	Updated	Updated
	Uninstalled	Uninstalled
Uninstall	Not installed	Not applicable
	Installed	Uninstalled
	Updated	Uninstalled
	Uninstalled	Uninstalled

Additional Settings

The following are the additional configuration settings for a package template:

- Validate connections
- Validate SQL scripts
- Execute SQL scripts
- Create package template

Validate Connections

The **Validate Connections** option allows you to validate all the connections in the package. This validation can be performed before the package is installed or updated. This option is available in the **Additional Settings** pane in the **Add to Package Template** and/or **Edit Package Template** pages.

For example: If you want to install a Package that has multiple database connections, then all the database connections are validated before the package is installed.

To validate database connections

1. Select and right-click the artifact that you want to package, and then click **Add to Package Template** on the shortcut menu. The **Select Package Template** page appears.
2. Select the Package Template.
3. Select **Package Template Version**, **Include Associated Artifacts**, **Include Child Artifacts**, **Include Security Artifacts**, and **Include the automatically generated SQL Script**.
4. Click **Next**. The **Add to Package Template** page appears.
5. In the **Configuration** section, click **Additional Settings**.

   The **Additional Settings** page appears on the right pane.
6. For the **Validate Database Connections** property, select one of the following:
   - **Yes**: Select this option if all the database connections in the package must be validated before the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

package is installed or updated. The default value is **Yes**.

- **No**: Select this option if all the database connections in the package need not be validated before the package is installed or updated. Select this option only when all the database connections in the package are valid.

7. Click **Save Settings**.
8. Click **Save Template**.

   The package template settings are saved.

Validate SQL Scripts

The **Validate SQL Scripts** option allows you to validate the SQL scripts for SDL-related keywords in the package. This validation can be performed before the package is installed or updated. This option is available in the **Additional Settings** pane in the **Add to Package Template** and/or **Edit Package Template** pages.

For example: If you want to install a Package that has multiple SQL scripts, then all the SQL scripts are validated before the package is installed.

To validate SQL scripts

1. Select and right-click the artifact that you want to package, and then click **Add to Package Template** on the shortcut menu. The **Select Package Template** page appears.
2. Select the Package Template.
3. Select **Package Template Version**, **Include Associated Artifacts**, **Include Child Artifacts**, **Include Security Artifacts**, and **Include the automatically generated SQL Script**.
4. Click **Next**. The **Add to Package Template** page appears.
5. In the **Configuration** section, click **Additional Settings**.

   The **Settings** page appears on the right pane.
6. For the **Validate SQL scripts** property, select one of the following:
   - **Yes**: Select this option if all the SQL scripts in the package must be validated before the package is installed or updated. The default value is **Yes**.

     If this option is selected, then the **Validate SQL Scripts** button is enabled in the **Settings** form of each of the following artifacts, and the SQL scripts for these artifacts can be validated.

     - Database Connection
     - Database Activity
     - Custom Queue
     - Database Query Lookup
     - Procedure Lookup
     - Custom List

     If you edit the settings for an artifact and click any other tree node, the SQL script is validated. If there are no restricted keywords, settings will be cached and the focus will be moved to the newly selected tree node.

     However, if there are no restricted keywords, then the settings will not be cached. Also, you will neither be allowed to select another tree node nor save the package template.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **No**: Select this option if all the SQL scripts in the package need not be validated before the package is installed or updated. Select this option only when all the SQL scripts in the package are valid.

  If this option is selected, then the **Validate SQL Scripts** button is disabled in the **Settings** form of each of the above-mentioned artifacts, and the SQL scripts for these artifacts are not validated.

  If you edit the settings for an artifact and click any other tree node, the settings will be cached. The SQL script is not validated.

  Note: If you do not want to validate SQL scripts, then it is recommended to set the value of the **Validate SQL scripts** property to **No** while creating a package template, and then add the required artifacts to the package template.

7. In the **Settings** page, click **Save Settings** and then click **Save Template**.

   The package template settings are saved.

Execute SQL Scripts

The **Execute SQL Scripts** option allows you to execute the SQL scripts available in the package. This option is available in the **Additional Settings** pane in the **Add to Package Template** and/or **Edit Package Template** pages.

You can execute the SQL scripts for the following artifacts.

- Database Connection
- Database Activity
- Custom Queue
- Database Lookup
- Procedure Lookup
- Custom List
- Manage List

To execute SQL scripts

1. Select and right-click the artifact that you want to package, and then click **Add to Package Template** on the shortcut menu. The **Select Package Template** page appears.
2. Select the Package Template.
3. Select **Package Template Version**, **Include Associated Artifacts**, **Include Child Artifacts**, **Include Security Artifacts**, and **Include the automatically generated SQL Script**.
4. Click **Next**. The **Add to Package Template** page appears.
5. In the **Configuration** section, click **Additional Settings**.

   The **Additional Settings** page appears on the right pane.
6. For the **Execute SQL scripts** property, select one of the following:
   - **Yes**: Select this option if you want to execute all the SQL scripts in the package before installing or updating the package. The default value is **Yes**.
   - **No**: Select this option if you do not want to execute all the SQL scripts in the package before installing or updating the package.
7. In the **Settings** page, click **Save Settings,** and then click **Save Template**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The package template settings are saved.

Create Package Template

The **Create Package Template** option allows you to create package template from the **Manage Package** page. The package template with the same name as the installed package, will be created in the **Package Template** window.

Note: When you want to create package template from a package having a list item enabled with security group, the security group will not be added to the package template.

In the **Additional Settings** of the **Edit Package Template** window, there is an option to choose to display **Create Package Template** option in the shortcut menu of the **Manage Package** page.

To choose the options available in the Additional Settings

1. Select and right-click the artifact that you want to package, and then click **Add to Package Template** on the context menu.

   The **Select Package Template** page appears.
2. Select the package template.
3. Select **Package Template Version**, **Include Associated Artifacts**, **Include Child Artifacts**, **Include Security Artifacts**, and **Include the automatically generated SQL Script**.
4. Click **Next**. The **Add to Package Template** page appears.
5. In the **Configuration** section, click **Additional Settings**.

   The **Additional Settings** page appears on the right pane.
6. For the **Create Package Template** property, select one of the following:
   - **Yes**: Allows you to create package template from the **Manage Package** page.
   - **No**: The **Create Package Template** option will not be available in the shortcut menu of the **Manage Package** page.

   The default value is **No**.
7. Click **Save Settings**.
8. Click **Save Template**.

   The package template settings are saved.

The package template will be created from the manage package definition. If you make any changes in the artifacts, the changes will not be reflected in the package template. In that case, use **Rebuild** option to get the changes. The version of the artifacts varies from the original package template.

The **Rebuild** option in the tree view appears irrespective of the artifact modified or not.

While creating a package template, the latest version of artifact available in the target environment will be displayed in the tree view of the version list artifact. If the package is not rebuilt, the artifact which was installed by the package will be added to the package template. However, the tree view displays the latest version of the artifact. If you rebuild the package, the artifact version displayed in the tree view is added to the package template.

If you create an artifact with different version, after the package template is created, then the artifact version displayed in the tree view will be considered, even if you rebuild the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: The **Create Package Template** option will be available in **Manage Package** page only if the option **Create Package Templates** is selected as **Yes** under **Additional Settings** in the Source Environment.

Consider that you create a package template from **Manage Package** page with **Runtime Initialization Settings** configured. When you install the package template, the form which is configured in Runtime Initialization Settings will be installed in the target environment in **Draft** status. You need to publish the form to utilize it in the package template created from the **Manage Package** page.

Note: In case the form already exists in the target environment, another form is created with the next major version in the **Draft** status and will get updated in the Package Template.

The **Create Package Template** option will be available only for the following package status:

Package Status	Allow to Create Package Template
Installed	Yes
Updated	Yes
Uninstalled	No
Not installed	No
Not Imported	No

Artifact Level Configuration Settings

In AVEVA Work Tasks, you can configure, rebuild, or delete artifacts of a package template.

Configure the Artifacts

The **Settings** tab and the **Install**, **Update**, and **Uninstall** sub tabs in the **Package Template** page allow you to set the properties for each of the artifacts added to the Package Template. Every single artifact has its own unique properties that you can configure based on the requirement. These properties decide the behavior of the artifacts while installing, updating or uninstalling the package.

The **Associated Artifacts** and the **Child Artifacts** tab is available only if the child and associated artifacts are added to the Package template.

To configure the artifacts within the package template

1. Select the artifact that you want to package and add it to a package template. The **Add to Package Template** page appears.
2. Expand **Package Template** under the **Configuration** section and set the **Prerequisite** and **Runtime Initialization Settings**.
3. Select the artifact for which you want to configure the properties.

   All similar artifacts will be grouped under one heading. For example: All Database Connections within the Package Template will be grouped under **Database Connections**, similarly all Lookups will be grouped under the heading **Lookup Settings**. The **Settings** tab along with its **Install**, **Update**, and **Uninstall** sub tabs appears on the right side of the page. The following section gives you in-depth knowledge of the properties in each of the **Install**, **Update**, and **Uninstall** tabs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The properties of the **Database Connection** artifact is explained in the **Install**, **Update**, and **Uninstall** tabs.

Settings

Install: The settings in this tab are referred when you install the Package.

- **Connection Exists**: This property allows you to take appropriate action when the Database Connection present in the package is already present in the target environment.
  - **Install to overwrite existing Connection**: select this option if you want to use the database connection that is present in the target environment.
  - **Do not install Connection**: select this option if you want to overwrite the database connection that is present in the target environment with the one that is present in the packaged.
- **Connect To Repository Data Source**: Using this property you can choose to use the current repository's connection string or choose a different connection string.
- The **Connect To Repository Data Source** option in the **Source** section allows you to view the connection string that is used in the source environment.
- The **Connect To Repository Data Source** option in the **Target** section allows you to edit the connection string of the target environment. You can either select Yes or No.
  - **Yes**: Select this option if you want to open the database connection using the current source repository's connection string.
  - **No**: Select this option if you want to define a new connection string.

Update: The settings in this tab are referred to when the same Package is installed for the consecutive time.

- Delete Connection: Select this option if you want to delete the Connection that is present in the target environment.
  - **Yes**: Select this option to remove the Connection in the target environment. On selection of this option, the property **Connection Is Modified** appears. For more information, refer to the following table:

Connection Is Modified	Retain	Select this option if you want to retain the Connection along with the modifications, in the target environment.
	Rollback	Select this option if you want to roll back to the original state. All modifications that were done to the Connection after it was installed in the target environment will be reverted and the connection will be retained.
	Delete	Select this option if you want to delete the Connection from the target environment when uninstalling the Package.

  - **No**: Select this option if you want to retain the Connection in the target environment. On selection of this option, the property **Connection Exists** and **Associated Artifacts not present** appears. For more

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

information, refer to the following table:

Connection Exists	Use Existing Connection	Select this option if you want to use the Connection that is present in the target environment.
	Do not update Connection	Select this option if you want to overwrite the Connection that is present in the target environment with the one that is currently being packaged.
Associated Artifacts not present	Update Connection	Select this option if you want to update the Connection.
	Do not update Connection	Select this option if you want to do not want to update the Connection as associated artifacts are not present.

- **Connect To Repository Data Source**: Using this property, you can choose to use the current repository's connection string or choose a different connection string.

  Note: The **Connect To Repository Data Source** option in the **Source** section allows you to view the connection string that is used in the source environment. The **Connect To Repository Data Source** option in the **Target** section allows you to edit the connection string of the target environment.

  - In the **Target** > **Connect To Repository Data Source** section, if **Yes** is selected then the database connection will be connected using the current repository's connection string.
  - In the **Target** > **Connect To Repository Data Source** section, if **No** is selected, then you can define a new connection string. For more information about defining a new connection, see Database Connection.

Uninstall: This settings in this tab is referred to when the Package is being uninstalled in the target environment.

- Delete Connection: This property allows you to decide if you want to retain or delete the Database Connection in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option if you want to delete the Database Connection from the target environment when the Package is uninstalled.
  - **No**: Select this option if you want to retain the Database Connection in the target environment even after the Package is uninstalled.
- **Connection Is Modified**: This option allows you to decide if you want to retain, roll back to original state or delete the Database Connection in the target environment even after the Package is uninstalled.

  Note: This property is applicable only if **Delete Connection** is set to **Yes** and the connection is modified.

  - **Retain**: Select this option if you want to retain the Database Connection along with the modifications in the target environment even after the Package is uninstalled.
  - **Roll back to original State**: Select this option if you want to roll back to the original state, reverted and the connection will be retained.
  - **Delete**: Select this option if you want to delete the Database Connection from the target environment

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

when uninstalling the Package.

Associated Artifacts

The Associated Artifacts tab is disabled for Database Connection artifact as it cannot have any associated artifacts.

Child Artifacts

The Child Artifacts tab is disabled for Database Connection artifact as it cannot have any child artifacts.

Rebuild the Artifacts

You can use the **Rebuild** option available on the shortcut menu of the modified artifact to get the latest definition of the artifact in the package template. When an artifact is rebuilt, the latest definition of the artifact is packaged. Associated artifacts, child artifacts, security artifacts, and automatically generated SQL scripts are also added based on the initial configuration settings of the artifact.

Rebuild

- An **orange icon** appears near the artifact indicating that the artifact is modified after it was added to the package template.
- In the shortcut menu, the **Rebuild** option is available only if the artifact is modified after it was added to the package template.
- After the artifact is rebuilt, a **green icon** appears indicating the artifact has been updated.
- If an artifact is disabled or deleted, then a **red icon** appears near the artifact.
- When an artifact is rebuilt, if any additional associated artifacts are found it will be added to the package template. However, associated artifacts present in the package template are not deleted or overwritten. For more details, see the scenario below.
- When an artifact is rebuilt, child artifacts are not rebuilt.

To rebuild the artifacts in the Package Template

1. In the **Enterprise Console**, click menu icon, click **Package**, and then click **Package Template**.
   The Package Template page appears which displays the list of available package templates.
   For information about creating a package template, see Create a Package Template.
2. Select a package template, and click **Edit** on the action menu.
   Alternatively, right-click a package template, and then click **Edit** on the context menu.
   The Edit Package Template page appears.
3. Click Save & Continue.
   The Edit Package Template page appears.
4. Right-click an artifact with an orange icon beside it, and then click **Complete Rebuild** or **Activity Rebuild** from the shortcut menu.
   A confirmation message about the rebuild appears.
5. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Repeat the process for all artifacts that needs to be rebuilt.
7. Click **Save Template** to save all the changes made to the template.

   The artifacts will be rebuilt and the latest definition of the artifacts will be updated in the package template.

Scenario

1. Create a Database Lookup on May 25, 2012, as explained:
   a. Enter **Title** as **TelematicsDBLookup**.
   b. Give a description to the Database Lookup.
   c. Select **Database Connection** as **TelematicsProductionDB**.
   d. Enter **Query** as **Select * from SWExecute Where ID=1**.
   e. Enter **Value**, **Display Name** and save the Database Lookup.
2. Add TelematicsDBLookup to a Package Template, **TelematicsDatabasePackage**.
3. On June 01, 2012, edit **TelematicsDBLookup** as follows:
   a. Change Database Connection to **TelematicsProductionDB2**
   b. Change the query as **Select * from SWExecute**.
4. On the June 02, 2012, edit the package template **TelematicsDBLookup**.

   An orange icon appears beside **TelematicsDBLookup** indicating that the definition has been updated after the lookup was added to the Package Template.
5. To get the latest definition of the Database Lookup, right-click *TelematicsDBLookup,* and then select **Rebuild** from the shortcut menu.

   A confirmation message appears.
6. Click **OK**.

   A confirmation message about the rebuild appears.
7. Click **OK**.
8. Verify the changes in the **Package Template** page.
   - A green icon appears beside *TelematicsDBLookup* indicating that the latest definition is updated in the Package Template.
   - The associated artifact *TelematicsProductionDB2* is also added to the Package Template. However, *TelematicsProductionDB* is not removed from **Package Template** > **Artifacts** > **Database Connections** even though it is not associated to the *TelematicsProductionDB2*.
9. Click **Save Templates** to save all the changes made to the artifacts.

Note:

- When an artifact, for example, a form is added to a package template, its associated artifacts are also added to the package template. If you delete an artifact from the form, and then perform a complete rebuild on the same package template, the deleted artifact is removed from the package template. However, the associated artifacts of the deleted artifact are retained in the package template. For example, consider that a form contains a database lookup, and this form is added to the package template. Now, if the database lookup artifact is deleted from the form, and if the package template is rebuilt again, though the database lookup is removed, its associated artifacts, such as the database connection is retained in the package template.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- In the **Associated Artifacts** tab of Database Lookup **TelematicsDBLookup**, the database connection **TelematicsProductionDB** will not be listed as it is not associated to database lookup.

Delete the Artifacts

You can delete the artifacts added to a Package Template by using the **Delete** option available on the shortcut menu of the artifact within the **Package Template** page.

To delete the artifact from the Package Template

1. Access the **Package Template** page.
2. Navigate to **Configuration** > **Package Template** > **Artifacts**.
3. Right-click the artifact that you want to delete and then select Delete from the shortcut menu.
4. A confirmation message appears.
5. Click **OK**.
6. Click **Save Templates** to save the changes done.
7. The artifact is removed from **Configuration** > **Package Template** > **Artifacts**.

Save Package Template

When you add new artifacts to a package template or edit an existing package template, you have to save the package template. This will ensure that the newly added artifacts and the modified settings of the package template are saved.

You can edit the following global and/or artifact level settings:

- Global Settings
  - Prerequisite
  - Runtime Initialization Settings
  - Additional Package Files
  - Additional Settings
- Artifacts

To save Global Settings

1. If you are adding artifacts to a package template, open the **Add to Package Template** dialog box.

   Or

   If you are editing an existing package template, then open the **Edit Package Template** dialog box.
2. Edit the global settings in the required tree node.
3. Click **Save Settings**.
4. Click **Save Template**.

   The changes to the package template are saved successfully.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To save settings for artifacts

1. If you are adding artifacts to a package template, open the **Add to Package Template** dialog box.

   Or

   If you are editing an existing package template, then open the **Edit Package Template** dialog box.
2. Select the artifact for which you want to edit the settings.
3. Edit the settings.
4. Click **Save Template**.

   The changes to the artifacts are saved successfully.

   Note: If you have edited the settings for an artifact and if you click any other tree node, the changes are saved in cache. Click **Save Template** to save these changes in the database.

To validate SQL Scripts

If there is a SQL script generated for the current artifact, the SQL script will be validated when you perform either of the following:

- Click any other tree node.
- Click the **Save Template** button.

The SQL server validation is performed based on the configuration defined for the **Validate SQL scripts** option in **Additional Settings** page.

For more information about validating SQL scripts, see Validate SQL Scripts.

Note: In System Platform IDE, while configuring Data Grid, the **Column Settings** window and the **Edit Operations Settings** window may not appear as expected. Resize the windows to get the proper view.

Publish a Package Template

You must publish and generate a package template before exporting it. You cannot edit or update a package template after publishing.

To publish a package template

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, point to **Package**, and then click **Package Template**.

   The Package Template page appears.
3. Select the package template that you want to publish, and then click **Publish**.

   Alternatively, right-click the package template that you want to publish, and then select **Publish** from the short-cut menu.

   A confirmation message about successfully publishing the package template appears.
4. Click **Close**.

   The package template status of the package changes to **Published** in the Package Template page.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Unpublish a Package Template

A Package Template that has a status of Published, Generated, or Exported can be unpublished to modify the configuration settings, prerequisites, modify, delete, or add new artifacts.

To unpublish a package template

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, point to **Package**, and then click **Package Template**.

   The Package Template page appears.
3. Select the package template that you want to unpublish, and then click the **Unpublish**.

   A message about successful unpublishing of the package template appears.

   Alternatively, right-click the selected package template, and then select the **Unpublish** option from the shortcut menu. The **Unpublish** option is available on the shortcut menu only if the Package Template is published.
4. Click **OK**.

   The Package Template is unpublished and the status of the Package Template is changed to **Draft**.

Generate a Package Template

You can generate a package template to create package file that can be exported from an existing Package Template. The Generate option creates a package file with an extension **.awsPKG**.

To generate a package template

1. In the **Enterprise Console**, click menu icon, point to **Package**, and then click **Package Template**.

   The **Package Template** page appears.
2. Select the package template that you want to generate, and the click the **Generate**.

   The **Generate** dialog box appears with a message about the successful generation of the package file.

   Alternatively, you can right-click the selected package template, and then select the **Generate** option from the shortcut menu.
3. Click **OK**.

   The **.awsPKG** file for the selected package template is created. You can export this package file.

   If the artifacts present in the package are modified after it was added to the package, the **Generate** dialog box alerts the user before generating the package file.

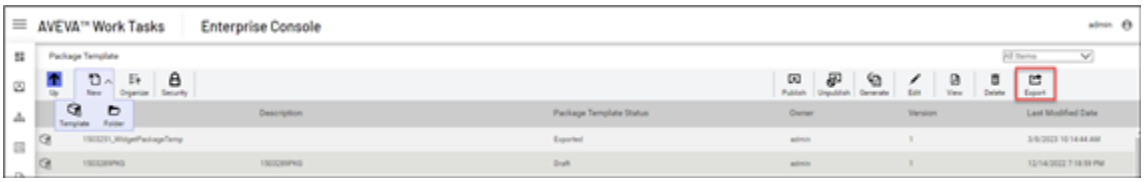
   Note: When the source file path specified in the **Package Template** is invalid or the source file is missing, the **Package Template** cannot be generated. In such cases, correct the filename or file path in the **Settings** tab of the respective artifact

Export a Package Template

Use the **Export** option to export and save the **.awsPKG** file generated from a package template to a location that

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

you want.



To export a package template file

1. In the **Enterprise Console**, click menu icon, click **Package**, and then click **Package Template**.
   The **Package Template** page appears.
2. Select the required package template, and then click the **Export** icon on the action bar.
   A confirmation message to export the package template appears.
   Alternatively, right-click the selected package template, and then select **Export** in the shortcut menu.
3. Click **OK** to export the package template, else click **Cancel**.
4. Provide the location information to save the **.awsPKG** file, and then save the file.
   A confirmation message about the successful export appears.
   The **.awsPKG** for the selected package template is saved in the specified location.

Note:

- The exported package file name format is **RepositoryName_PackageTemplateName_PackageTemplateVersion_TargetPlatform.awsPKG**.

- You can export package templates with status **Generated** or **Shipped** only.

- When you export a package template that is in **Draft** status, the system automatically publishes and generates the package template before exporting the same. Similarly, when the package template is in **Published** status, the system automatically generates the package template before exporting the same.

- You can import an exported package template from source environment to the target environment. For more information about importing a package, see Import a Package.

Mark Package Template as Shipped

You can mark exported Package Template files to "Shipped" using the **Mark as Shipped** option on the shortcut menu. This enables you to track all the package templates that were shipped. Before marking a package template as "Shipped", ensure that the package template is published, generated, and exported.

To mark a Package Template as shipped

1. In the **Enterprise Console** , click menu icon, point to **Package**, and then click **Package Template**.
   The **Package Template** page appears.
2. Right-click the selected package template, and then select the Mark as Shipped option on the shortcut menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The status of the Package Template changes to **Mark as Shipped**.

You can import a Package Template that you have exported from source environment to the target environment.

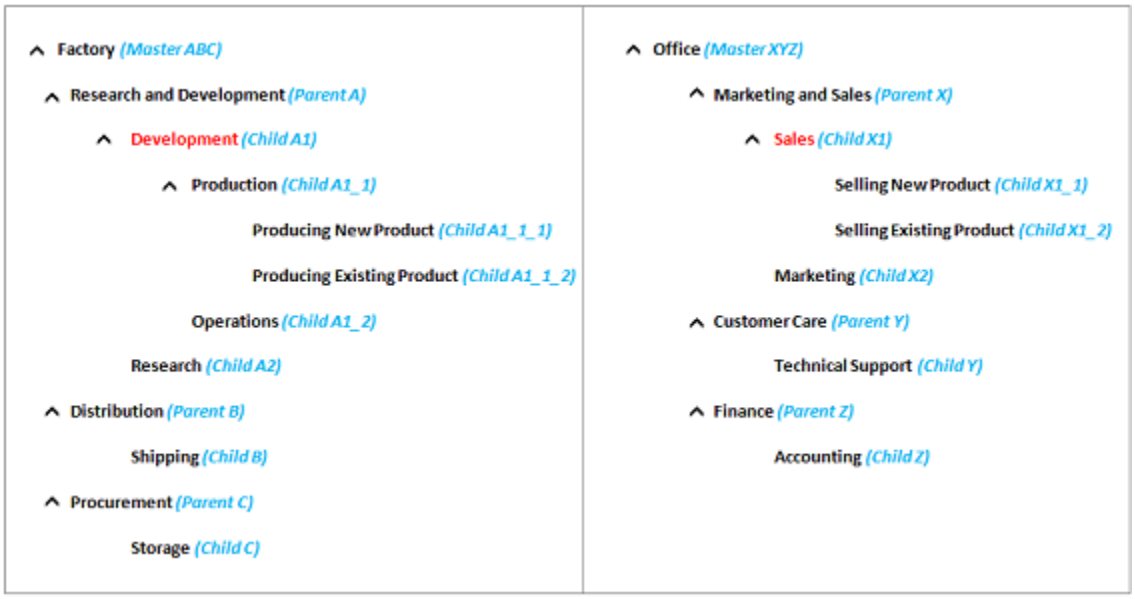
Note: You cannot edit package templates that are marked as "Shipped".

Include Associated and Child Artifacts to a List Item

When you are packaging a list item, you can include any one or both of the following:

- **Associated artifacts**: Following are considered as associated artifacts for the item to be packaged:
  - The parent item of the item to be packaged.
  - The artifacts on which the item to be packaged has a dependency upon.
- **Child Artifacts**: The direct child items/folders of the item to be packaged. If the direct child has child items, then these child items are also included in the package. Therefore, all the child items, under the item to be packaged, are included in the package.

For example: Consider the folder structures with the following hierarchy.



Let us consider that you want to package "Development", which has a dependency on "Sales".

The following table describes the various scenarios based on the selection of "Include Child Artifacts" and "Include Associated Artifacts" options.

Include Child Artifacts	Include Associated Artifacts	Folders/Artifacts selected along with "Development"
No	No	Only "Development" is selected.
No	Yes	- Factory - Research and Development - Office

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Include Child Artifacts	Include Associated Artifacts	Folders/Artifacts selected along with "Development"
		- Marketing and Sales - Sales
Yes	No	- Production - Producing New Product - Producing Existing Product - Operations
Yes	Yes	- Factory - Research and Development - Production - Producing New Product - Producing Existing Product - Operations - Office - Marketing and Sales - Sales - Selling New Product - Selling Existing Product

You can create child artifacts for the following lists. Therefore, the above scenario is applicable only to these lists.

- Workflow
- Forms
- ArchestrA Events
- Forms Controls
- Navigation List
- Assembly list
- Manage list
- Lookup
- Grid Configuration
- Reports
- Security Rights

Let us consider another example where the folder structures have the following hierarchy.

Let us consider that the artifact to be packaged is "Form A", which has a dependency on "Lookup A" and "Workflow A" and that "Workflow A" in turn has dependency on "Workflow X".

The following table describes the various scenarios based on the selection of "Include Child Artifacts" and "Include Associated Artifacts" options.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Include Child Artifacts	Include Associated Artifacts	Folders/Artifacts selected along with "Development"
No	No	Only "Form A" is selected.
No	Yes	- Main Form - Form AB - Main Lookup - Lookup AB - Lookup A - Main Workflow - Workflow AB - Workflow A - Workflow XY - Workflow X
Yes	No	- Form A1 - Form A2
Yes	Yes	- Main Form - Form AB - Form A1 - Form A2 - Main Lookup - Lookup AB - Lookup A - Lookup A1 - Lookup A1_1 - Lookup A1_2 - Lookup A2 - Main Workflow - Workflow AB - Workflow A - Workflow A1 - Workflow A2 - Workflow XY - Workflow X - Workflow X1

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Include Child Artifacts	Include Associated Artifacts	Folders/Artifacts selected along with "Development"
		- Workflow X2

To include associated and child artifacts to a List Item

1. Select and right-click the artifact that you want to package, and then click **Add to Package Template** on the shortcut menu. In the **Select Package Template** page, select the **Package Template**.

   Or

   Create a new package template. For more information about creating a package template, see Creating a Package Template.
2. In the **Include Associated Artifacts** section, perform the following:
   - Select **Yes** to package all the associated artifacts of the selected artifact. For example, for the Database Lookup example here, if you select Yes, the associated database connection is also added to the package template. The associated artifact of the database lookup is a database connection.
   - Select **No** to exclude the associated artifacts of the selected artifact. For example, for the Database Lookup example here, if you select No, the associated database connection is not added to the package template.
3. In the **Include Child Artifacts** section, perform the following:
   - Select **Yes** to package all the child artifacts of the selected artifact. Here, the child artifacts of the associated artifact is also added to the package template. For example, if Database Lookup has child lookups, the child lookup can also be packaged by selecting Yes.
   - Select **No** to exclude the child artifacts of the selected artifact from the Package Template.
4. Click **Save Template**.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Include Security Artifacts

Complete the following tasks to include the list item level security of an artifact to the package template:

Prerequisite

Enable the security settings for an artifact/list item that you want to package.

1. In the List Item page, click **Security** on the Action bar.

   Or

   In the List Item page, right-click the artifact and select **Security Settings** from the context menu.
2. In the **Security** window, go to **Advanced Settings** tab, and click **Enable List Item specific security**.

   The **Enable Security Wizard** dialog box appears.
3. Select **Security Group Customization**.
4. Publish the artifact.

For more information about enabling security at list item level, see the *Enabling Security at ListItem Level* section in the *User Guide*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To include security artifacts

1. Select and right-click the artifact that you want to package, and then click **Add to Package Template** on the shortcut menu. In the **Select Package Template** page, select the **Package Template**.

   Or

   Create a new package template. For more information about creating a package template, see Creating a Package Template.
2. In the **Include Security Artifacts** section, perform one of the following:
   - **Yes**: Select this option if you want to include the list item level security artifacts in the package template.
   - **No**: Select this option if you do not want to include the list item level security artifacts in the package template.
3. Click **Save Template**.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Include Automatically Generated SQL Script

You can include the automatically generated SQL script to the package template for the following artifacts:

- Database Lookup
- Manage List pointing to existing table
- Custom Queue
- Workflow with Database activity

To include the automatically generated SQL script

1. Select and right-click the artifact that you want to package, and then click **Add to Package Template** on the shortcut menu. In the **Select Package Template** page, select the **Package Template**.

   Or

   Create a new package template. For more information about creating a package template, see Creating a Package Template.
2. In the **Include the automatically generated SQL Script** section, perform one of the following:
   - **Yes**: Select this option if you want to include the automatically generated SQL Script for the artifact to the package template.
   - **No**: Select this option if you do not want to include the automatically generated SQL Script for the artifact to the package template.
3. Click **Next**. The **Add to Package Template** page appears.
4. In the **Configuration** section, click the artifact. The **Settings** page appears on the right pane.
5. In the **Install** tab:
   - The automatically generated SQL Script is displayed by default in the **SQL script** box, if the value of the **Include the automatically generated SQL Script** option is set to **Yes**.
   - The **SQL script** box is blank, if the value of the **Include the automatically generated SQL Script** option is set to **No**. If required, enter the SQL Script in the **SQL script** box.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Click **Save Template**.

For more information about adding artifacts to Package Template and validating SQL scripts, see Adding Artifacts to a Package Template and Validate SQL Scripts respectively.

Manage Package

You can use the **Manage Package** list in the target environment to import packages that are exported from the package template of the source environment. You can then install, update, or uninstall the package in the target environment.

Note: Consider the repository with a **Blue** theme, and a Form Link is packaged from this repository. When you install the package in the target environment with a **Lite** theme, the Form Link icon will reflect the colors representing the **Blue** theme of the source repository and not the **Lite** theme of the target repository.

Action Executables in Ribbon bar

The following table lists the icons of all the actions that you can perform from the ribbon of the **Manage Package** window:

Actions		**Description**
Up		To navigate from the current folder to the parent folder. This feature is helpful when you are accessing a folder within the package and you want to go back to the parent folder of Manage Package.
New	Import	To import a package file to the target environment. A package file has the extension **awsPKG**.
	Folder	To group and categorize the packages.
Organize		To organize the packages and group it in specific folders.
Security		To define security at the list item level.
Edit		To modify or delete the artifacts, modify the configuration of the artifacts, and the prerequisites.
View		To view the package details in read-only mode.
Delete		To delete the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Actions	Description
	You can delete packages that are in *Draft* status. You cannot delete a package after you install it.
Install	To run and install the imported package.
Uninstall	To uninstall the package.
History	To view the entries logged in history when a package is imported, installed, updated, and uninstalled.

Action Executables in Shortcut Menu

The following table lists the icons of all the actions that you can perform from the shortcut menu of the **Manage Package** window:

Options	Description
Edit	To modify or delete the artifacts, modify the target values, configuration state, and view the prerequisites.
View	To view the details of the selected package in read-only mode.
Install	To run and install the imported package.
Uninstall	To uninstall the package.
History	To view the entries logged in history when a package is imported, installed, and uninstalled.
Delete	To delete the package. You can delete packages that are in *Draft* status. You cannot delete a package after you install it.
Show All versions	To view all the versions of the selected package. This option is useful as the Manage Package list will display only the latest published version of the package.
Security Settings	To define security at the list item level.

Import a Package

You can execute and install package files that are imported to the **Manage Package** list in the target

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

environment. Use the **Import** option in the **Manage Package** list to import the **.awsPKG** file in the target environment. A package does not get imported, if any background msi runs in the target machine.

To import a package

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, point to **Package**, and then click **Manage Package**.

   The **Manage Package** page appears.
3. On the action bar, click **New**, and then click **Import**.

   The **Import Package** dialog box appears.
4. Browse to select the **Package File**.
5. Enter a description for the package file to be imported.
6. Click **Import**.

   The **Manage Package** list in the target environment displays the imported package. The status of the imported package is **Not Installed**. If you import the same package again, the package is saved with the next major version.

Manage Versions of a Package

You can create a new version of package by importing an existing package again in the **Manage Package** list of the target environment. When a package is imported again, the system prompts you to save the package as a next major version. This is beneficial if the package file has been updated with new artifacts or modifications are done to the artifacts and the configuration settings.

To save a package as next major version

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, point to **Package**, and then click **Manage Package**.

   The Manage Package window appears.
3. On the action bar, click **New**, and then click **Import**.

   The **Import Package** dialog box appears.
4. Click **Browse** to navigate to the location where the exported package file is saved.
5. Select a package file that is already imported and present in the target environment.
6. In the **Description** box, enter a description for the package file that is being imported.
7. Click **Import**.

   A confirmation message *about the package* already existing appears.
8. Click **OK**.

   A confirmation message successful package importing appears.
9. Click **OK**.

   The next major version of the package is created.

If all the versions of the package are in **Not Installed** status, the latest version appears in the Manage Package list. If one of the version is installed, then the installed version of the package appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Store Package Information for Lists

AVEVA Work Tasks Lists store package-related information for ListItems.

The following table describes the columns, which store the package-related information.

Column Name	Description
Additional Information	Displays the additional information specified while creating the package template.
Package Name (Version)	The name and version of the package.
Artifact Version	The version of the installed artifact.
Is Artifact Modified	Indicates whether the artifact is modified after package installation.

When a package created in the earlier versions is installed in the latest version, the **Additional Information** column will be blank.

These package-related columns are available for the following lists:

- Forms
- Workflows
- Database Connection
- Database Lookup
- Grid Configuration

  Note: The **Artifact Version** column is not applicable for Database Connection, Database Lookup, and Grid Configuration lists.

- In a List grid, only the **Package Name (Version)** column is displayed by default. If you want to include the other columns in the List grid, then you have to create a view, include the other columns in the list, and specify the order in which these columns must be displayed. The selected columns will be available as part of the List grid.

  For more information about creating a new view, see the *Creating a New View* topic in the *Forms and Lists User Guide*.
- Consider a ListItem which has package-related information. If you save this ListItem as next major, minor, or specific versions, then the value of the **Is Artifact Modified** column is set to **False**. The values of the other columns will be blank.
- Consider a ListItem, which was installed by a package. If you edit this ListItem, then all the package-related column values are retained. The value of the **Is Artifact Modified** column is set to **True**, indicating that the ListItem was modified after package installation.
- For ListItems installed by a package or created manually in Wonderware Skelta BPM 2017 or earlier, the column values will be blank. If you edit these ListItems, then the value of the **Is Artifact Modified** column is set to **False**.
- For manually created ListItems, the value of the **Is Artifact Modified** column is set to **False**. The values of the other columns will be blank.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The package-related columns provide the following information:

- The name of the package which installed the artifact.
- The name of the source repository of the package.
- Whether the artifact was manually modified after installation.

Scenario

For example, consider an application package named **Work Order Mgt** with version 1.0. The artifacts available in the package are listed in the following table.

Artifact Name	**Artifact Version**
WO Management Overview Form	2.0
Start Work Order Form	1.5
End Work Order Form	1.4
Pause Work Order Form	1.0

You import this package into a new repository named “Target Repository” and then install this package in the target environment.

The package-related information appears as shown in the following table.

Artifact Name	**Source Artifact Version**	**Additional Information**	**Package Name (Version)**	**Artifact Version**	**Is Artifact Modified**
WO Management Overview Form	1.0(*)	SE MES Software	Work Order Mgt (1.0)	2.0	False
Start Work Order Form	1.0(*)	SE MES Software	Work Order Mgt (1.0)	1.5	False
End Work Order Form	1.0(*)	SE MES Software	Work Order Mgt (1.0)	1.4	False
Pause Work Order Form	1.0(*)	SE MES Software	Work Order Mgt (1.0)	1.0	False

* indicates the "Latest" Form that will be used by the application.

When you save an artifact as the next version in the target environment, then the package-related information columns is displayed as blank values, as shown in the following table.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Artifact Name	Source Artifact Version	Additional Information	Package Name (Version)	Artifact Version	Is Artifact Modified
WO Management Overview Form	1.0	SE MES Software	Work Order Mgt (1.0)	2.0	False
	1.1				False
	1.2(*)				False
Start Work Order Form	1.0(*)	SE MES Software	Work Order Mgt (1.0)	1.5	False
End Work Order Form	1.0(*)	SE MES Software	Work Order Mgt (1.0)	1.4	False
Pause Work Order Form	1.0(*)	SE MES Software	Work Order Mgt (1.0)	1.0	False

* indicates the "Latest" Form that will be used by the application.

Note: The “Save artifact as next version” example is applicable only for Version Lists, such as Forms and Workflow.

When you edit any artifact in the target environment, then the value of the **Is Artifact Modified** column is set to “True” for the edited artifact, as shown in the following table.

Artifact Name	Source Artifact Version	Additional Information	Package Name (Version)	Artifact Version	Is Artifact Modified
WO Management Overview Form	1.0	SE MES Software	Work Order Mgt (1.0)	2.0	False
	1.1				False
	1.2(*)				False
	2.0	SE MES Software	Work Order Mgt (2.0)	3.0	False
Start Work Order Form	1.0	SE MES Software	Work Order Mgt (1.0)	1.5	False
	2.0(*)	SE MES Software	Work Order Mgt (2.0)	2.0	True
End Work Order Form	1.0	SE MES Software	Work Order Mgt (1.0)	1.4	False
	2.0(*)	SE MES Software	Work Order Mgt (2.0)	1.5	False
Pause Work	1.0	SE MES Software	Work Order Mgt	1.0	False

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Artifact Name	Source Artifact Version	Additional Information	Package Name (Version)	Artifact Version	Is Artifact Modified
Order Form			(1.0)		
	2.0(*)	SE MES Software	Work Order Mgt (2.0)	2.5	False

* indicates the "Latest" Form that will be used by the application.

Is Artifact Modified Column for Workflow

The **Is Artifact Modified** column is set to **True** when the workflow definition changes for a workflow, which was installed by a package.

This is applicable only when:

- The workflow is installed by the package and contains the package-related information.
- The workflow is modified for the first time.

Conditions when the Is Artifact Modified column is set to True

The **Is Artifact Modified** column is set to **True** when:

- The workflow design changes. For example: An activity is added, an existing activity is removed, property of an activity is modified, and so on.
- The workflow is edited.
- The workflow is imported.
- A task that modifies the workflow definition is selected from the Workflow Task list.
- An association for File Watcher or ArchestraA (ASB) Events, is created in the workflow.
- An ArchestraA Event is mapped from IDE to a workflow.
- An **Invoke Workflow** control is used in form.
- A form is associated to the workflow.

Note: When you create an association (for example, Scheduler), an XML variable named **WorkflowSchedulerXMLData** is created. Due to this, the workflow definition changes and the **IS Artifact Modified** column is set to **True**. However, if the workflow already contains this XML variable, then the value of the **IS Artifact Modified** column is retained as-is. This is also applicable for other associations to workflow.

Conditions when the Is Artifact Modified column is not updated

The **Is Artifact Modified** column is not updated when:

- The workflow is published or unpublished.
- The workflow is enabled or disabled.
- The workflow is locked or unlocked.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The owner of the workflow is changed.
- The security settings of the workflow are updated.
- BAM is configured.

Edit a Package

You can edit an imported package by using the **Edit** option on the action bar. You can edit the package to modify the configuration settings of the artifacts and the pre-defined behavior of the package at the time of installation.

To edit the package

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, point to **Package**, and then click **Manage Package**.

   The Manage Package page appears.
3. Select a package that you want to edit, and then click **Edit**.

   The **Edit Package** dialog box appears.

   Alternatively, you can right-click the selected package, and then select **Edit** from the shortcut menu.
4. Make the required changes such as, editing the configuration settings of the artifacts or viewing the prerequisites.
5. Click **Save**.

View Prerequisites of a Package

The **Prerequisite** option in the **Manage Package** page > **Configuration** displays the details of the packages that are required for installation prior to installing the current package. The Prerequisite option appears in the **Manage Package** page only if prerequisites are defined while designing the package template in the source environment.

Note: You cannot modify the prerequisites in the Manage Package page of a target environment.

Install a Package

Use the **Install** option in the Manage Package list to import, install, update, and uninstall the package. While running a package, you also have an option to modify the configuration settings and the pre-defined settings that determines the behavior of the installation.

Prerequisites

Before installing the package, ensure that the **Client Service** is running.

To run and install a package

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, point to **Package**, and then click **Manage Package**.

   The **Manage Package** page appears.
3. Select the package that you want to install, and then click **Install**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Alternatively, you can right-click the selected package, and then click **Install**.

The **Pre-install Validation and Configuration** page appears.

- If you have configured any Runtime Initialization settings, then the **Runtime Initialization Settings** form appears. You can update the runtime initialization values, if required. For more information about updating the runtime initialization values, see Update or modify the runtime initialization values.
- Click **Edit** to edit the configuration for any artifact, if required. For more information about editing a package, see Editing a Package Template.
- You can also reload the validation log messages for artifacts. For more information about reloading the validation messages, see the **Reloading the validation log messages for artifacts** section below.
- You can validate the log messages before proceeding with the installation. For more information about validating log messages, see Validate the log messages if present.
- The **Pre-install Validation and Configuration** page shows the count for **Errors**, **Warnings**, **Information**, and **All**. Click a category of message to view its related information. For example: If you want to view the warning messages, click **Warnings**.
- If the prerequisites of the package are not installed, the **Prerequisites** page appears which lists out all the prerequisites along with its status. To proceed with the installation, all the packages which are set as prerequisites must be installed.

Note:
- At this stage of the installation process (when the **Install** option is selected), if the artifact present in the package is already present in the target environment, a backup of the artifact as present in the target environment is taken.

4. Click **Install Package**.

   The **Package Installation** page appears.

   The progress of the package installation is displayed in the progress bar.

   After the package installation is complete, the status of each artifact and installer files (if applicable) is displayed.

   If there were any errors/warnings during the installation of any artifact, then the count of the errors, warnings, and information appears in this page.
5. Click the **Errors** / **Warnings** option to view the details.
6. Click **Close**.

   For more information about package history, see View Package History.

   The status of the package is updated as **Installed**.

Note:

- A package can be installed, updated, or uninstalled only by a repository administrator who is having the **Can Install** rights.

- If you package a form with custom controls, then you must install the custom control files (DLL files) in the target environment before installing the package.

Reload the validation log messages for artifacts

When you select an artifact and edit it before installing the package, the updated messages are displayed only for that artifact. However, if you edit other artifacts while editing the selected artifact, then the messages are

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

updated only for the selected artifact, and not for the other artifacts.

For example: Consider a scenario where “Package1” consists of three artifacts namely Database Connection, Grid Configuration, and Lookup Settings.

- Scenario 1: You select the Database Connection artifact and click **Edit**, and then you save the settings and the package. In this scenario, the updated messages for Database Connections are displayed and the Database Connection artifact is reloaded.
- Scenario 2: You select the Database Connection artifact and click **Edit**. You also edit some of the settings for the Grid Configuration and Lookup Settings artifacts in the respective **Settings** tab, along with the settings for Database Connection. After you save the settings and the package, only the updated messages for Database Connections are displayed. The messages related to Grid Configuration and Lookup Settings artifacts will be displayed in the **Pre-install Validation and Configuration** page, only after clicking **Reload**.

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, point to **Package**, and then click **Manage Package**.

   The **Manage Package** page appears.
3. Select the package that you want to install, and then click **Install**.

   Alternatively, you can right-click the selected package, and then click **Install**.

   The **Pre-install Validation and Configuration** page appears.
4. Click **Edit** to modify the settings of artifacts.
5. After all the changes are done for the artifacts, click **Save** to save the package.
6. Click **Reload**.

   The updated messages are displayed for all the edited artifacts and these artifacts are reloaded.

Configure the Package

Ensure to fulfill the installation prerequisites for the package for a successful configuration.

Package Installation - Prerequisites

While installing a package, if the prerequisites defined for the package are not installed, the **Package Installation** page appears. This page displays the list of all the packages which are set as prerequisites, along with the version number and the status.

Column Name	Description
Name	Name of the package which is set as prerequisite.
Version	The version of the package template as in the source environment.
Higher Version	Displays either Yes or No. • **Yes** appears if the option *Consider higher version if selected version does not exist in target*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Column Name	Description
	environment is selected, while designing the package template. • **No** appears if the option *Consider higher version if selected version does not exist in target environment* is not selected, while designing the package template.
Status	This column displays the following different statues: • **Installed**: This status appears when the prerequisite is installed in the target environment. • **Updated**: This status appears when the prerequisite is updated in the target environment. • **Uninstalled**: This status appears when the prerequisite is uninstalled in the target environment. • **Not Installed**: This status appears when the prerequisite is imported to the target environment but not installed. • **Not Imported**: This status appears when the prerequisite is not imported to the target environment.

Runtime Initialization Values

The **Runtime Initialization Values** is the first screen that appears while installing the package, wherein you need to enter the runtime initialization values if it not already entered while designing the package template or edit the existing values. This page does not appear if the Prerequisites of the package are not installed.

In this screen, you can edit the existing values if present or enter the required values. Click **Next** to proceed with the package installation.

Additional Package Files

While configuring the package, you can view the additional custom files and the installer files to be installed in the target environment.

Click **Additional Package Files** on the configuration tree view at the left hand side of the Add to Package Template window. The settings form appears where you can view the additional custom files and the installer files.

View Validation Log

The **Package Configuration** screen displays all the exceptions and validations found while installing the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The exceptions and validation messages are displayed in a grid and each line item of the grid represents an exception encountered due to which an artifact was not installed or a validation message which indicates the artifact was successfully installed.

You can validate the exceptions by clicking the **Edit** button available for every single line item in the **Package Configurations** grid.

To validate the exceptions

Consider that you have several exceptions to validate.

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, click **Package**, and then click **Manage Package**.

   The **Manage Package** page appears.
3. Select the package that you want to install, and then click **Install**.

   The **Package Configuration** page appears.

   Before installation, all the artifacts are validated based on the settings configured in the **Settings** form.

   The relevant messages are displayed in the **Package Configurations** grid.

   Note: All these validations are performed by the Client service.

View Package History

You can view the history of a package using the **History** option. An entry is logged in package history when a package is imported, installed, and uninstalled. The **Package History** page automatically appears when the package installation or uninstall is in progress. The records are displayed in reverse chronological order.

To view the history of a package

1. Log on to the Enterprise Console.
2. On the **Enterprise Console** menu, point to **Package**, and then click **Manage Package**.

   The Manage Package page appears.
3. Select the package and click **History**.

   The **Package History** dialog box appears where you can view, export, and filter the data.

   Alternatively, you can right-click the selected package, and then select **History** on the shortcut menu.

Package History Features

- **Filtering Data**: You can filter the data in the Package History dialog box by enabling the **Filter Bar**.

  To enable the filter bar, right-click the column label, and then select **Filter Bar** on the shortcut menu. A new row with the filters is added under the column header using which you can filter the data. Data filter functionality in the Package History dialog box is similar to *Filtering Data in Work Item List*.
- **Sort Data:** You can sort data in the Package History dialog box using the **Sort** option available on the shortcut menu of column headers. The columns having numerical value or value in the form of date and time are sorted in the ascending order on selection of A-Z. On selection of Z-A, the values are sorted in descending order.
- **Export**: You can export the data in the Package History dialog box to a html, excel, pdf, xml, tiff, and text

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

format using the **Export** feature.

- **View more records**: The grid in the Package History dialog box displays only 50 records. To view next 50 records, click the **View more records** icon available in the lower-right corner of the window.
- **Error Counts**: The Package History dialog box shows the count for **Errors**, **Warnings**, **Information**, and **All**. Click an error count to see the related messages.

Upgrade Scenarios

After the product is upgraded, you might come across the following observations, while opening any existing package:

- For database connection artifact in a package, the password specified before the upgrade, is not retained. You have to manually enter the password again in the Package Settings form.
- For the artifacts such as Manage List, Custom List, Database Lookup, Queue, or Database Activity, SQL scripts are not available for the existing package. To resolve this issue, you have to open the package and rebuild the artifact or create a new package template.
- For report template path, the **Resolving Template Path** grid is not displayed. To resolve this issue, you have to open the package and rebuild the artifact or create a new package template.
- If some lists and/or items are modified during upgrade, some artifacts might be marked as modified in the existing packages.

Uninstall a Package

An installed package can be uninstalled using the **Uninstall** option. When a package is uninstalled, all the artifacts with the associated and child artifacts will be uninstalled from the target environment and all configuration settings that were done will be reverted.

Prerequisite

Before uninstalling the package, ensure that the Client Service is running.

To uninstall a Package

1. Log on to the Enterprise Console.
2. On the **Enterprise Console** menu, point to **Package**, and then click **Manage Package**.
   The **Manage Package** window appears.
3. Select the package that you want to uninstall, and then click **Uninstall**.

   Alternatively, you can right-click the selected package, and then click **Uninstall**.

   The **Pre-uninstall Validation and Configuration** page appears.
   - You can edit the configuration for any artifact in the **Pre-uninstall Validation and Configuration** page, if required. For more information about editing a package, see Editing a Package Template.
   - You can also reload the validation log messages for artifacts. For more information about reloading the validation messages, see the **Reloading the validation log messages for artifacts** section below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- You can validate the log messages before proceeding with the uninstallation. For more information about validating log messages, see Validate the log messages if present.
- The **Pre-uninstall Validation and Configuration** page shows the count for **Errors**, **Warnings**, **Information**, and **All**. Click a category of message to view its related information. For example: If you want to view the warning messages, click **Warnings**.

4. Click **Uninstall**.

   The **Package Uninstallation** screen appears when the uninstall is in progress.

   The progress of the package uninstallation is displayed in the progress bar.

   After the package uninstallation is complete, the status of each artifact and installer files (if applicable) is displayed.

   If there were any errors/warnings during the uninstallation of any artifact, then the count of the errors, warnings, and information appears in this page.

5. Click **Close**.

The artifacts in the package are deleted. The status of the package is updated as **Uninstalled**.

Note:
- While uninstalling the package all the artifacts may not be deleted, if a backup of the artifact is present in the target environment.
- If you are uninstalling an artifact which has multiple versions, then all the versions of the artifact will be uninstalled. However, the artifact will be considered as only one artifact. For example: If you are uninstalling a Workflow artifact which has versions 1 and 2, then both the versions will be uninstalled. However, the Workflow artifact will be considered as only one artifact.

Reload the validation log messages for artifacts

When you select an artifact and edit it before uninstalling the package, the updated messages are displayed only for that artifact. However, if you edit other artifacts while editing the selected artifact, then the messages are updated only for the selected artifact, and not for the other artifacts.

For example: Consider a scenario where “Package1” consists of three artifacts namely Database Connection, Grid Configuration, and Lookup Settings.

- Scenario 1: You select the Database Connection artifact and click **Edit**, and then you save the settings and the template. In this scenario, the updated messages for Database Connections are displayed and the Database Connection artifact is reloaded.
- Scenario 2: You select the Database Connection artifact and click **Edit**. You also edit some of the settings for the Grid Configuration and Lookup Settings artifacts in the respective **Settings** tab, along with the settings for Database Connection. After you save the settings and the package, only the updated messages for Database Connections are displayed. The messages related to Grid Configuration and Lookup Settings artifacts will be displayed in the **Pre-uninstall Validation and Configuration** page, only after clicking **Reload**.

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, point to **Package**, and then click **Manage Package**.

   The **Manage Package** page appears.

3. Select the package that you want to uninstall, and then click **UnInstall**.

   Alternatively, you can right-click the selected package, and then click **UnInstall**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Pre-uninstall Validation and Configuration** page appears.

4. Click **Edit** to modify the settings of artifacts.
5. After all the changes are done for the artifacts, click **Save Settings** and then click **Save** to save the package.
6. Click **Reload**.

   The updated messages are displayed for all the edited artifacts and these artifacts are reloaded.

Delete a Package

You can delete a package from the target environment if the package is not installed. Packages once installed, updated or uninstalled cannot be deleted.

To delete a package

1. Log on to the Enterprise Console.
2. On the **Enterprise Console** menu, point to **Package**, and then click **Manage Package**.
   The **Manage Package** window appears.
3. Select a package and then click the **Delete** icon on the ribbon.

   Or

   Right-click the selected package and then select **Delete** from the shortcut menu.

   The Package will be deleted.

Note: For installed packages, **Delete** option will not be available on the shortcut menu.

Migration of Packages

A package exported in Wonderware Skelta BPM 2014 R2 SP2 P01 and above versions, can now be seamlessly imported and installed in the latest version of AVEVA Work Tasks.

When a package containing:

- Custom DLLs, is built with lower version of Skelta DLL, the custom DLLs must be rebuilt manually.
- Form and Embed Form, which has an Image/Hyperlink control with a different control ID, is migrated, the form along with the embed form will be migrated successfully.
- Database Connection is migrated, the Database Connection proceeds seamlessly, without any password encryption.
- Forms and Grid Configurations that are modified before the upgrade are not marked as modified in the package template after migration of the same.

Remotely Managing Packages using Console Application

A user with Administrator privileges can export, import, install, or uninstall packages using the **WWSKBPMRemotePackageManager** Console Application. You can also remotely export, import, install, or uninstall packages using the **WWSKBPMRemotePackageManagerConfig.xml** file.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Remotely Export, Import, Install, or Uninstall Packages using Console Application

A user with **Administrator** privileges can export, import, install, or uninstall packages using the **WWSKBPMRemotePackageManager** Console Application.

1. Open the Command Prompt in the Administrative mode.
2. Enter the path for the folder, where the Remote Package Manager files are available, and then press **Enter**.
3. On the Command Prompt, type **WWSKBPMRemotePackageManager.exe**.
4. Pass one of the values for the following command line arguments:
   - **To export a package from the repository**
     WWSKBPMRemotePackageManager.exe /export
   - **To import a package into the repository**
     WWSKBPMRemotePackageManager.exe /import
   - **To install a package from the repository**
     WWSKBPMRemotePackageManager.exe /install
   - **To uninstall a package from the repository**
     WWSKBPMRemotePackageManager.exe /uninstall
5. Press **Enter**.

   The package is exported, imported, installed, or uninstalled remotely depending on the value passed for the command line argument.

Note:
- The values for all the related parameters such as package name, repository name, and so on will be considered from the **WWSKBPMRemotePackageManagerConfig.xml** file.
- Ensure that the folder to which the package must be exported is a shared folder.

Restricting Export, Import, Install, or Uninstall operations on remote machines (Optional)

You can update the IP addresses of the remote machines. Only the machines whose IP addresses are included in the tag (explained below), will be allowed to execute the **WWSKBPMRemotePackageManager.exe**. All the other remote machines will be restricted from executing this exe.

To update remote machine IP addresses in the Farm DB

1. Access the Farm DB.
2. Execute the following query

```
Select Properties from SKService where Name = 'SkeltaGlobalSettings'
```

3. Copy the value in the **Properties** to a Notepad.
4. Add the **<property key="validremotepackageclientiplist"><![CDATA[xxxx.xxxx.xxxx.xxxx]]></property>** tag as shown below.

```
Update SKService set Properties = '<properties><property>……..</property><property
key="validremotepackageclientiplist"><![CDATA[xxxx.xxxx.xxxx.xxxx]]></property></prop
erties>' where Name = 'SkeltaGlobalSettings'
```

Note:
- Ensure that you add this tag before the closing **</properties>** tag.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The **WWSKBPMRemotePackageManager** will verify whether the **<property key="validremotepackageclientiplist"><![CDATA[xxxx.xxxx.xxxx.xxxx]]></property>** tag is available in the **Properties** of the **AVEVAWorkTasksGlobalSettings** record in the **SKService** table. If the tag is available, then only the remote machine IP addresses present in the tag are allowed to execute the **WWSKBPMRemotePackageManager.exe**.

- You can add multiple IPV4 and/or IPV6 IP addresses by using ';' as a separator. For example: *[fe80::8d01:b4f8:489:f4c8%12;fe80::8d01:b4f8:489:f4c8%19;10.13.88.100;10.13.88.101]*

- Ensure that you specify only machine IP addresses.

5. Add the IP addresses of the remote machines.
6. Execute the query in the Farm DB.
7. Restart the **Client Service**.

Configure WWSKBPMRemotePackageManager Using XML

You can remotely export, import, install, or uninstall packages using the **WWSKBPMRemotePackageManagerConfig.xml** file.

Consider the following scenario:

- *Server1* is the server on which AVEVA Work Tasks is installed
- *Machine1* is the client machine on which AVEVA Work Tasks is not installed

You can remotely export, import, install, or uninstall packages to/from the Server1 or Machine1 by using the **WWSKBPMRemotePackageManagerConfig** console application.

1. Browse to the folder where the Remote Package Manager files (WorkTasks.Pacakge.Remote.dll, AVEVAWorkTasksRemotePackageManager.exe, and WWSKBPMRemotePackageManagerConfig.xml) are available.
2. Open the **WWSKBPMRemotePackageManagerConfig.xml** file.
3. In the XML file, pass the values for the mandatory parameters listed in the following table.

Settings Category	Node	Description
Export	RepositoryName	Enter the name of the source repository from which the package must be exported.
	Server	Enter the name of the source server.
	ServerPort	Enter the port number. This is the port number used for listening by the WWSKBPM client service. The default value is **8863**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	UserId	Enter the user ID in the **instancename::Identifier Column** format. For example: If the repository provider is Active Directory, then enter the authentication as **activedirectory: SHELBYLAB\skuser1**. If the repository provider is AVEVA Work Tasks List, then enter the authentication as **worktaskslist: : GUID**.
	Name	Enter the name of the package that must be exported from the repository. Ensure that the package is published and generated.
	Version	Enter the version of the package that must be exported from the repository
	PhysicalPath	Enter the shared physical path of the client machine. For example: If the AWSPkg file is present in the Packages folder in "Machine1", then enter the path as *Machine1\Packages*. Ensure the following: • The folder where the Remote Package Manager files are available is a shared folder. • The AVEVA Work Tasks admin has the *Read/Write* permission for the network folder.
Import	RepositoryName	Enter the name of the target repository to which the package must be exported.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
	Server	Enter the name of the target server.
	ServerPort	Enter the port number. This is the port number used for listening by the AVEVAWorkTasks client service. The default value is **8863**.
	UserId	Enter the user ID in the **instancename::Identifier Column** format. For example: If the repository provider is Active Directory, then enter the authentication as **activedirectory: SHELBYLAB\skuser1**. If the repository provider is AVEVA Work Tasks List, then enter the authentication as **worktaskslist: : GUID**.
	Name	Enter the name of the package that must be imported to the repository.
	Description	Enter the description for the package.
	Parent	Enter the GUID of the parent list item or folder.
	PhysicalPath	Enter the shared physical path of the client machine. For example: If the AWSPkg file is present in the Packages folder in "Machine1" (where AVEVA Work Tasks is not installed), then enter the path as *Machine1\Packages*. Ensure the following: • The folder where the Remote Package Manager files are available is a shared folder. • The AVEVA Work Tasks admin

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		has the *Read/Write* permission for the network folder.
Install	RepositoryName	Enter the name of the repository on which the package must be installed.
	Server	Enter the name of the server on which the package must be installed.
	ServerPort	Enter the port number. This is the port number used for listening by the WWSKBPM client service. The default value is **8863**.
	UserId	Enter the user ID in the **instancename::Identifier Column** format. For example: If the repository provider is Active Directory, then enter the authentication as **activedirectory: SHELBYLAB\skuser1**. If the repository provider is Work Tasks List, then enter the authentication as **worktaskslist : : GUID**.
	Name	Enter the name of the package that must be installed.
	Version	Enter the version of the package that must be installed.
Uninstall	RepositoryName	Enter the name of the repository from which the package must be uninstalled.
	Server	Enter the name of the server from which the package must be uninstalled.
	ServerPort	Enter the port number. This is the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
		port number used for listening by the WWSKBPM client service. The default value is **8863**.
	UserId	Enter the user ID in the **instancename::Identifier Column** format. For example: If the repository provider is Active Directory, then enter the authentication as **activedirectory: SHELBYLAB\skuser1**. If the repository provider is AVEVA Work Tasks List, then enter the authentication as **worktaskslist : : GUID**.
	Name	Enter the name of the package that must be uninstalled.
	Version	Enter the version of the package that must be uninstalled.

4. Save the XML file.
5. Run the **WWSKBPMRemotePackageManager.exe**.

   For more information about any errors during these operations, refer the **WTLogger Console**.

Import, Install, or Uninstall Packages using Console Application

You can import, install, or uninstall packages using the **AVEVAWorkTasksPackageManager** Console Application.

1. Open the Command Prompt in the Administrative mode.
2. Enter the path for the "Bin" folder, which is available in the path where the product is installed, and then press **Enter**.

   For example: CD <Install Path>\Bin
3. On the Command Prompt, type **AVEVAWorkTasksPackageManager.exe**.
4. Pass one of the values for the following command line arguments:
   - **To import a package into the repository**

     AVEVAWorkTasksPackageManager.exe /import
   - **To install a package from the repository**

     AVEVAWorkTasksPackageManager.exe /install

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **To uninstall a package from the repository**

  AVEVAWorkTasksPackageManager.exe /uninstall

5. Press **Enter**.

   The package is imported, installed, or uninstalled depending on the value passed for the command line argument.

   Note: The values for all the related parameters such as package name, repository name, and so on will be considered from the **WWSKBPMPackageManagerConfig.xml** file.

Configure WWSKBPMPackageManager Using XML

You can import, install, or uninstall packages using the **WWSKBPMPackageManagerConfig.xml** file.

1. Browse to the "Bin" folder, which is available in the path where the product is installed.
2. Open the **WWSKBPMPackageManagerConfig.xml** file.
3. In the XML file, pass the values for the mandatory parameters listed in the following table.

Settings Category	Node	Description
Import Package Settings	Repository Name	Enter the name of the repository to which the package must be imported.
Import Package Settings > Package	Name	Enter the name of the package that must be imported into the repository.
	Description	Enter the description for the package.
	Parent	Enter the name of the parent folder for the package to be imported.
	Physical Path	Enter the physical path where the package must be imported.
Install Package Settings	Repository Name	Enter the name of the repository to which the package must be imported.
	LoggedInUserID	Enter the ID of the logged in user.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Category	Node	Description
Install Package Settings > Package	Name	Enter the name of the package that must be installed.
	Version	Enter the version of the package that must be installed.
Uninstall Package Settings	Repository Name	Enter the name of the repository from which the package must be uninstalled.
	LoggedInUserID	Enter the ID of the logged in user.
Uninstall Package Settings > Package	Name	Enter the name of the package that must be uninstalled from the repository.
	Version	Enter the version of the package that must be uninstalled from the repository

4. Save the XML file.
5. Run the **AVEVAWorkTasksPackageManager.exe**.

Add Artifacts to a Package Template

You can add an item to a package template using the **Add to Package Template** shortcut menu.

Note: You can package only published artifacts. Thus, the **Add to Package Template** option is available only for the published items.

In this illustration, Database Lookup is used as an example of list items artifact.

1. Log on to the Enterprise Console.
2. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Lookup Settings**.
   The **Lookup Settings** list appears.
3. Select the Database Lookup that you want to package.
4. Right-click the Database Lookup and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.
5. Select an existing package template from the **Package Template** list. Package templates in the **Draft** status appear in the **Package Template** list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

OR

Create a new package template by performing the following:

a. Select **Create Package Template** from the **Package Template** drop-down list. The **New Package Template** page appears.
b. Enter the title and description for the package template in the **Title** and **Description** fields respectively.
c. In the **Target Platform** section, select either **32-bit** or **64-bit** based on the machine where the package needs to be installed.
d. Click **Save Template**. The **Package Template** list is updated with the newly created package template.

6. In the **Package Template Version** drop-down list, select the version of the package template. By default, the version is updated as **1**.
7. In the **Include Associated Artifacts** section:
   - Select **Yes** to package all the associated artifacts of the selected artifact. For example, for the Database Lookup example here, if you select **Yes**, the associated database connection is also added to the package template. *The associated artifact of the database lookup* is a database connection.
   - Select **No** to exclude the associated artifacts of the selected artifact. For example, for the Database Lookup example here, if you select **No**, the associated database connection is not added to the package template.
8. In the **Include Child Artifacts** section:
   - Select **Yes** to package all the child artifacts of the selected artifact. Here, the child artifacts of the associated artifact is also added to the package template. For example, if Database Lookup has child lookups, the child lookup can also be packaged by selecting **Yes**.
   - Select **No** to exclude the child artifacts of the selected artifact from the Package Template.
9. In the **Include Security Artifacts** section:
   - Select **Yes** to package all the security rights of the selected artifact.
   - Select **No** to exclude the security rights of the selected artifact from the Package Template.

Note: You can see the **Include Security Artifacts** option only if you enable the security group customization of the artifact at the List Item level.

10. In the **Include the automatically generated SQL Script** section:
    - Select **Yes** to include the automatically generated SQL Script for the artifact to the package template.
    - Select **No** to exclude the automatically generated SQL Script for the artifact to the package template.

Note: The **Include the automatically generated SQL Script** option is available only for Lookup Settings, Manage List pointing to existing table, Queue, and Workflow with Database activity.

11. Click **Next**.

    The **Add to Package Template** page appears. On the left pane, you can see the artifact Database Lookup. You can make all required changes to the configuration settings and prerequisites.

12. Click **Save Template**.

    The Database Lookup along with the configuration settings and prerequisites is saved to the Package Template.

Note: When you select the **Include Associated Artifacts** option as **Yes** while adding an item (or artifact) to a Package Template, the associated artifacts are added to the package template. You can then view the associated artifacts of the item in the tree view of a Package. If you select the **Include Associated Artifacts** option as **No**, the associated artifacts are not added to the package template. Then, you cannot see

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the associated artifacts in the tree view of a Package. This condition holds good for both the **Include Child Artifacts** and **Include Security Artifacts** options as well. In the case, when you do not have any child artifact, selecting the **Include Child Artifacts** option as **Yes** or **No** does not make any difference.

See Also

Configure a Package Template

ArchestrA Events

You can package and deploy the ArchestrA Events along with its associated artifacts and child artifacts from one environment or repository to another environment or repository.

To add ArchestrA Events to Package Template

1. Log on to the Enterprise Console.
2. On the menu, point to **Tools**, and then click **ArchestrA Events**.

   The **ArchestrA Events** window appears.
3. Select the list item that you want to package.
4. Right-click the list item and then click **Add to Package Template**.

   The **Add to Package Template** window appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of the ArchestrA Events in the Add to Package Template window. The left side of the configuration window displays the package template and its configurations. You can set the configuration properties of the ArchestrA Events package in the Settings tab form on the right side of the Add to Package Template window.
5. Configure the required settings.
6. Click **Save Template** to save the configured settings.

Configuration Settings for ArchestrA Events

The **ArchestrA Events** has an unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Associated Artifacts
- Child Artifacts

Settings Tab

You can use the properties in the **Settings** tab to take appropriate actions to install, update, or uninstall a package.

Install

Use this tab to set the configurations for installing the package.

- **ArchestrA Event Exists**: Select this property to take appropriate action when the ArchestrA events in the package is already present in the target environment.
  - **Do not install ArchestrA Event**: Select this option to use the ArchestrA Events, present in the target environment.
  - **Install to overwrite the existing ArchestrA Event**: Select this option to overwrite the ArchestrA Events, present in the target environment with the packaged ArchestrA event.

    During the package installation, backup will be taken before overwriting the existing artifact.

Update

Use the settings in this tab to update an installed package.

- **Delete ArchestrA Events**: Select this property to delete or retain the ArchestrA events that is present in the target environment.
  - **Yes**: Select this option to delete the ArchestrA Events, that was installed by the package.

    ArchestrA Event is modified: Select this option to retain, roll back the ArchestrA Events to original state, or delete the ArchestrA Events from the target environment even after the package is updated. This property is applicable only if **Delete ArchestrA Events** is set to *Yes* and the ArchestrA Events is modified.
    - **Retain**: Select this option to retain the ArchestrA Events along with the modifications in the target environment.
    - **Rollback to original state**: Select this option to roll back the artifact to the backup taken. All modifications that were done to the ArchestrA Events after it was installed in the target environment will be reverted. However, the ArchestrA Events will remain in the target environment.
    - **Delete**: Select this option to delete the ArchestrA Events from the target environment.
  - **No**: Select this option to retain the ArchestrA Events in the target environment.

    ArchestrA Event exists: Select this property to take appropriate action when the ArchestrA Events that is being installed, is present in the target environment.
    - **Do not install ArchestrA Event**: Select this option to use the ArchestrA Events, present in the target environment.
    - **Install to overwrite existing ArchestrA Event**: Select this option to overwrite the ArchestrA Event, present in the target environment with the packaged ArchestrA Event.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete ArchestrA Event**: Select this property to retain or delete the ArchestrA Events from the target environment even after the package is uninstalled.
  - **Yes**: Select this option to delete the ArchestrA Events from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the ArchestrA Events in the target environment even after the package is uninstalled.
- **ArchestrA Event is modified**: Select this option to retain, roll back the ArchestrA Events to original state, or delete the ArchestrA Events from the target environment even after the package is uninstalled. This property is displayed only if **Delete ArchestrA Event** is set to *Yes*. However, if **Delete ArchestrA Event** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the ArchestrA Events along with the modifications in the target environment.
  - **Rollback to original state**: Select this option to roll back to the artifact to the backup taken. All modifications that were done to the ArchestrA Events after it was installed in the target environment will be reverted. However, the ArchestrA Events will remain in the target environment.
  - **Delete**: Select this option to delete the ArchestrA Events from the target environment.

Associated Artifacts Tab

ArchestrA Events can include the following associated artifacts:

- Parent Folder
- Parent ArchestrA Event

Child Artifacts for Tab

ArchestrA Events can include the following child artifacts:

- Child ArchestrA Event
- Folder

Include Child Item

For more information, see Include Child Item.

Security Artifacts for ArchestrA Events

You can enable the security settings of the ArchestrA Events that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

BAM

You can package and deploy the BAM configuration associated with workflow from one environment or repository to another environment or repository. The BAM events that can be associated with the workflow are Overdue Analysis, KPI Analysis, and KPIs.

Note: You can rebuild Overdue Analysis, KPI Analysis, and KPIs associated with the workflow but cannot rebuild the Basic Configuration associated with the workflow.

To add BAM to Package Template

1. Log on to the Enterprise Console.
2. Create a workflow and configure the workflow with BAM configuration using the **BAM Configuration** option. For more information, see Associate Workflow with Enterprise Console Events and Access BAM Configuration
3. Save BAM configuration associated with the workflow and publish the workflow.
4. Right-click the workflow and then click **Add to Package Template**.

   The **Add to Package Template** window appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of the BAM in the Add to Package Template window. The left side of the configuration window displays the package template and its configurations. You can set the configuration properties of the workflow configured with BAM on the right side of the Add to Package Template window.
5. Configure the required settings.
6. Click **Save Template** to save the configured settings.

Configuration Settings for BAM associated with Workflow

The **BAM** has an unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Child Artifacts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Tab

You can use the properties in the **Settings** tab to take appropriate actions to install, update, or uninstall a package.

Install

Use this tab to set the configurations for installing the package.

- **BAM Exists**: Select this property to take appropriate action when the BAM in the package is already present in the target environment.
  - **Overwrite existing BAM**: Select this option to overwrite the BAM, present in the target environment with the packaged BAM.

    During the package installation, backup will be taken before overwriting the existing artifact.
  - **Use existing BAM**: Select this option to use the BAM, present in the target environment.

Update

Use the settings in this tab to update an installed package.

- **Delete BAM Settings**: Select this property to delete or retain the BAM settings that is present in the target environment.
  - **Yes**: Select this option to delete the BAM, that was installed by the package.

    If BAM Settings are modified after installation: Select this option to retain, roll back the BAM settings to original state, or delete the BAM settings from the target environment even after the package is updated. This property is applicable only if **Delete BAM** is set to *Yes* and the BAM setting is modified.
    - **Retain**: Select this option to retain the BAM settings along with the modifications in the target environment.
    - **Rollback to original state**: Select this option to roll back the artifact to the backup taken. All modifications that were done to the BAM settings after it was installed in the target environment will be reverted. However, the BAM settings will remain in the target environment.
    - **Delete**: Select this option to delete the BAM settings from the target environment.
  - **No**: Select this option to retain the BAM settings in the target environment.

    BAM exists: Select this property to take appropriate action when the BAM that is being installed is present in the target environment.
    - **Overwrite existing BAM**: Select this option to overwrite the BAM settings present in the target environment with the packaged BAM.
    - **Use existing BAM**: Select this option to use the BAM settings present in the target environment.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete BAM Settings**: Select this property to retain or delete the BAM settings from the target environment

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

even after the package is uninstalled.

- **Yes**: Select this option to delete the BAM settings from the target environment when the package is uninstalled.
- **No**: Select this option to retain the BAM settings in the target environment even after the package is uninstalled.

- **If BAM Settings are modified after installation**: Select this option to retain, roll back the BAM settings to original state, or delete the BAM settings from the target environment even after the package is uninstalled. This property is displayed only if **Delete BAM Settings** is set to *Yes*. However, if **Delete BAM** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the BAM settings along with the modifications in the target environment.
  - **Rollback to original state**: Select this option to roll back to the artifact to the backup taken. All modifications that were done to the BAM settings after it was installed in the target environment will be reverted. However, the BAM settings will remain in the target environment.
  - **Delete**: Select this option to delete the BAM settings from the target environment.

Associated Artifacts Tab

BAM can include the following associated artifacts:

- Parent Folder
- Parent Workflow

Child Artifacts for Tab

BAM can include the following child artifacts:

- Child Workflow
- Folder

Include Child Item

For more information, see Include Child Item.

Security Artifacts for BAM

You can enable the security settings of the BAM that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

Calendar

You can package and deploy an organization Calendar from one environment or repository to another environment or repository using the package feature in AVEVA Work Tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To add Calendar to Package Template

1. Log on to the Enterprise Console.
2. On the menu, point to **Settings**, and then click **Calendar**.
   The **Calendar Control** window appears.
3. Select the calendar that you want to package.
   If you want to create a calendar, see "Chapter: About Calendar" in the *User Guide*.
4. In the new **Calendar Control** window for the calendar that you opened, click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Configuration Settings for Calendar

The **Calendar** has an unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment** > **Package Template** window
- **Target Environment** > **Manage Package** window

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Child Artifacts

Settings Tab

You can use the properties in the **Settings** tab to take appropriate actions to install, update, or uninstall a package.

Install

Use this tab to set the configurations for installing the package.

- **Calendar Exists**: Select this property to take appropriate action when the calendar in the package is already present in the target environment.
  - **Do not install calendar**: Select this option to use the calendar, present in the target environment.
  - **Install to overwrite the existing calendar**: Select this option to overwrite the calendar, present in the target environment with the packaged calendar. During the package installation, backup will be taken before overwriting the existing artifact.
- **Associated Artifacts Not Present**: Select this property to take appropriate action when the associated

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

artifacts of the calendar are not present in the target environment.

- **Install calendar**: Select this option to install the calendar, even if the associated artifacts are not present in the target environment.
- **Do not install calendar**: Select this option, if you do not want to install the calendar as its associated artifacts are not present in the target environment.

Update

Use the settings in this tab to update an installed package.

- **Delete Calendar**: Select this property to delete the calendar that was installed by the package.
  - **Yes**: Select this option to remove the calendar from the target environment.

    If Calendar Modified After Installation: Select this property if calendar was present prior to package installation or the calendar installed by the package is modified.
    - **Retain**: Select this option to retain the calendar along with the modifications in the target environment even after the package is updated.
    - **Rollback to original state**: Select this option to roll back the artifact to the backup taken. All modifications that were done to the calendar after it was installed in the target environment will be reverted. However, the calendar will be retained in the target environment.
    - **Delete**: Select this option to delete the calendar from the target environment.
  - **No**: Select this option to retain the calendar in the target environment.

    Calendar Exists: Select this property to take appropriate action when the calendar that is being installed is present in the target environment.
    - **Do not install calendar**: Select this option to use the calendar that is present in the target environment.
    - **Install to overwrite existing calendar**: Select this option to overwrite the calendar present in the target environment with the packaged calendar.

      Associated Artifacts Not Present: Select this property if associated artifact is not present in the target environment
    - **Update calendar**: Select this option to update the calendar.
    - **Do not update calendar**: Select this option if you do not want to update the calendar as associated artifacts are not present.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete Calendar**: Select this property to retain or delete the calendar from the target environment even after uninstalling the package.
  - **Yes**: Select this option to delete the calendar from the target environment when the package is uninstalled.

    If Calendar Modified After Installation: Select this option to retain, roll back the calendar to original state, or delete the calendar from the target environment even after the package is uninstalled. This property is displayed only if **Delete Calendar** is set to *Yes*. However, if **Delete Calendar** is set to *No*, this

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

property is not displayed.

- **Retain**: Select this option to retain the calendar along with the modifications in the target environment.
- **Rollback to original state**: Select this option to roll back to the backup taken. All modifications that were done to the calendar after it was installed in the target environment will be reverted. However, the calendar will remain in the target environment.
- **Delete**: Select this option to delete the calendar from the target environment.

- **No**: Select this option to retain the calendar in the target environment after uninstalling the package.

Associated Artifacts Tab

By default, a calendar does not have any associated artifact.

While creating a calendar, if you inherit another calendar into it, then the inherited calendar appear as the associated artifact of the calendar.

Child Artifacts Tab

Calendar does not have any child artifact.

Database Connection

You can package and deploy a Database Connection from one environment or repository to another environment or repository using the package feature in AVEVA Work Tasks.

Note:The Database connection configured with **Database Type** as **Oracle** and **Connection Type** as **Connection provider** is not supported.

To add Database Connection to Package Template

1. Log on to the Enterprise Console.
2. On the menu, point to **Settings**, and then click **Database Connections**.

   The Connection List window appears.
3. Select the database connection that you want to package.
4. Right-click the database connection and then click **Add to Package Template**.

   The Add to Package Template dialog box appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of a Database Connection in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the Database Connection package in the **Settings** form on the right side of the Add to Package Template window.

Runtime Initialization Settings

You can configure runtime initialization settings for database connections. For more information about

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration Settings for Database Connection

The **Database Connection** has an unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment** > **Package Template** window
- **Target Environment** > **Manage Package** window

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Child Artifacts

Settings Tab

You can define the properties in the **Settings** tab to take appropriate actions to install, update, or uninstall a package.

Install

Use this tab to set the configurations for installing the package.

- **Connection Exists**: Select this property to take appropriate action when the database connection in the package is already present in the target environment.
  - **Do not install connection**: Select this option to use the database connection, present in the target environment.
  - **Install to overwrite the existing connection**: Select this option to overwrite the database connection in the target environment with the one that is present in the package. During the package installation, backup will be taken before overwriting the existing artifact.

Update

Use the settings in this tab to update an installed package.

- **Delete connection**: Use this property to delete the database connection present in the target environment.
  - **Yes**: Select this option to remove the database connection from the target environment.

    Connection is modified: Select this option to retain, roll back to original state, or delete the database connection in the target environment even after the package is updated. This property is applicable only

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

if **Delete connection** is set to *Yes* and the database connection is modified.

- **Retain**: Select this option to retain the database connection with the modifications in the target environment even after the uninstalling the package.
- **Rollback to original state**: Select this option to rollback the database connection to the original state. All modifications done to the database connection after installation in the target environment will be rolled back. However, the database connection will be retained in the target environment.
- **Delete**: Select this option to delete the database connection from the target environment while updating the package.

- **No**: Select this option to retain the database connection in the target environment.

  Connection exists: Select this option to take appropriate action when the database connection that is being installed is present in the target environment.

  - **Do not install connection**: Select this option if you do not want to update the database connection as associated artifacts are not present.
  - **Install to overwrite existing connection**: Select this option to overwrite the database connection present in the target environment with the one that is currently being packaged.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete connection**: Select this property to retain or delete the database connection in the target environment after you uninstall the package.
  - **Yes**: Select this option to delete the database connection from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the database connection in the target environment even after the package is uninstalled.
- **Connection is modified**: Select this property to retain, rollback, or delete the database connection in the target environment after uninstalling the package. This property is displayed only if **Delete connection** is set to *Yes*. However, if **Delete connection** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the database connection with the modifications in the target environment even after the package is uninstalled.
  - **Rollback to original state**: Select this option to rollback the database connection to the original state. All modifications that were done to the connection after it was installed in the target environment will be reverted. However, the connection will be retained in the target environment.
  - **Delete**: Select this option to delete the database connection from the target environment when uninstalling the package.

Connection Details

Use the **Connect to Repository DataSource** settings to choose between an existing repository connection string or a different connection string.

Source:

- Use the **Connect to Repository DataSource** option in the **Source** to view the connection string that is used in the source environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Target:

- Use the **Connect to Repository DataSource** option in the **Target** to edit the connection string of the target environment.
  - Select *Yes* to use the connection string of the current repository to connect to the database connection.
  - Select *No* to define a new connection string.

Field	Description
Database type	Select SQL Server.
Connection type	Select either Connection string or Connection provider.

If Connection type is Connection string

Field	Description
Server name	Enter the name of the data source server.
Authentication type	Provide information on the process used to validate a user. Both Windows and Microsoft SQL Server authentication process is supported. If Database type is Microsoft SQL Server, then authentication type can be either Microsoft SQL Server or Windows.
User Id	Enter the Microsoft SQL Server user ID for Microsoft SQL Server type of authentication. For Windows Authentication, **User Id** is disabled.
Password	Enter the Microsoft SQL Server password for Microsoft SQL Server type of authentication. For Windows Authentication, **Password** is disabled.
Advanced settings	(Optional). You can enter any additional connection information, such as connection timeout. For example, Connection Timeout = 20 While creating a data source for a repository, use the following custom settings to set properties for connection pool: `encrypt=true; nableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;` where, • **encrypt:** Ensures secure communication between

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption. • **enableltm**: Internally light weight transaction mode is implemented by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same connection is used within the transaction, the transaction distribution is reduced to a great extent. With this feature the transaction is controlled from being escalated to a distributed transaction which is very expensive. The above property can be used to turn off this feature. The default value is set as *True*. • **connectionretrywaittime**: While opening a new connection, if the database server is not available, the default behavior is to retry the connection. This setting can be used to set the wait time to retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds. • **maxconnectionretries**: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite. • **sqlcommandtimeout:** Sets the wait time before terminating the attempt to execute an SQL command and generating an error. The default value is set as 300. **Note:** These settings are case sensitive and should be mentioned only in lower case.
Database	Enter the name of the database.

If Connection Type is Connection Provider

Field	Description
Assembly path	Enter the path of the connection dll. In a client-server environment, the dll must be present on the machine that hosts the Enterprise Console site. All requests of install and uninstall are sent to the host

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	site.
Param	Specify the parameters required by the connection dll.
SQL script	Enter the SQL script in the query window. The SQL script written in the query window can be packaged from one SQL server machine and can be installed in another SQL server machine only. Oracle is not supported. The **Validate SQL script** button validates the script for the usage of the restricted SQL commands, but does not validate syntax of the SQL query in the script. Ensure to enter an error free SQL script. The following SQL commands are restricted: • Delete • Truncate • Drop • Update • Create User • Grant • Revoke • Execute
Attach SQL script	Use this button to upload a SQL script. You can attach files of .sql type only. By default, the **Attach SQL script** button is invisible. Click the **Attach SQL script file** check box to see it. The SQL scripts are executed in sequence. First the script in the SQL script query window is executed followed by the attached SQL script. Even if the script execution in the SQL script query window fails, the attached SQL script continues to execute. You can attach only one script per database connection. Maximum size of the attached file is restricted to 10 mb.

Note:

If the SQL script is executed successfully , then the following message appears in the **Package History** dialog box. *The SQL script is executed successfully.*

If the SQL script execution fails, then the following message appears:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The SQL script execution failed.

If you click the message, the **Message Details** page appears, where you can view the message details.

See Also: SQL Script Generation and Execution

Associated Artifacts of Database Connection

Database Connection does not have any associated artifact.

Child Artifacts of Database Connection

Database Connection does not have any child artifact.

Security Artifacts of Database Connection

You can enable the security settings of the database connection that you want to include in the package.

For more information about including security artifacts, see Include Security Artifacts.

Event Associations

You can package and deploy the Event Associations with its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

You can package and deploy the Event Associations of the following types:

- Scheduler
- Enterprise Console
- File Watcher
- System Platform Events
- Event Associations of Type Logic App
- Event Associations of Type Message Queue

Event Associations of Type Scheduler

You can package and deploy the Event Associations of type Scheduler and its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add Event Associations of type Scheduler to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click ☰ menu icon, and then click **Workflows**. The **Workflows** window appears.
3. Select a Workflow with an event association of type scheduler associated to it.
4. Right-click the workflow, and then select **View Association**. All associations of the selected workflow appear.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Right-click **Event Associations of type Scheduler** that you want to package, and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Runtime Initialization Settings

You can configure runtime initialization settings for Event Associations of type Scheduler. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration Settings for Event Associations of type Scheduler

The Event Associations of type Scheduler has a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the Settings tab to define the configuration settings. The Associated Artifacts tab displays the associated artifacts of the Event Association of type Scheduler definition. The Child Artifacts tab displays child artifacts of the Event Association of type Scheduler definition.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Security Artifacts

Settings Tab

You can use the properties of the Settings tab to take appropriate actions to install, update, or uninstall the package.

Behavior of Event Associations of type Scheduler while installing, updating or uninstalling the package:

- You cannot overwrite an already existing item.
- You can delete or uninstall an Event Associations of type Scheduler only if it was installed by the Package.

Install

You can use to the settings in this tab when install the package. When installing the package, if the Event Scheduler dates are invalid, then the package will not be installed. Edit the dates and install the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Workflow Version**: This displays the version of Workflow to which Event Association of type Scheduler is mapped for Scheduling in the Source environment. It also allows you to specify the version of Workflow to which Event Association of type Scheduler should be mapped in the Target environment.
  - **Source**: Displays the version of Workflow in the source environment to which the Event Association of type Scheduler is mapped.

    Target: Enter the version of the workflow:

    Specify the version of the Workflow to which you want to map the Event Association of type Scheduler in target environment.

    If you do not specify the Workflow version in the Target environment, the Workflow version from the Source environment will be considered while installing the package.

    Note:
    - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is not present in the target environment, then the Event Association will not be installed.
    - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is in Draft status, then the Event Association of type Scheduler will be installed and mapped to the latest version of the workflow.

- **Scheduler Event Association Settings**: This displays the settings of the Event Association of type Scheduler in the Source environment and it also allows you to specify the settings of the Event Association of type Scheduler in the Target environment.
  - **Start Date**
    - **Source**: Displays the start date of the Event Association of type Scheduler in the source environment.
    - **Target**: Select the start date of the scheduler on which you want the event to be triggered.

      If you do not use the date picker, ensure that the date is in the MM\dd\yyyy format.
  - **End Date**
    - **Source**: Displays the end date of the Event Association of type Scheduler in the source environment.
    - **Target**: Select the end date of the scheduler on which you want the event to be triggered.

      If you do not use the date picker, ensure that the date is in the MM\dd\yyyy format.
  - **Start Time**
    - **Source**: Displays the start time of the Event Association of type Scheduler in the source environment.
    - **Target**: Select the start time of the scheduler on which you want the event to be triggered.

      If you do not use the date picker, ensure that the time is in any one of the following formats:

      hh:mm:tt

      hh:mm tt

      h:mm:tt

      h:mm tt

      Note: Ensure that you follow the following guidelines while installing the Event Associations of type scheduler package in the target environment:
      - Start date, Start time, and End date should be in valid format.
      - Start date should be less than or equal to End date respectively.
      - End date should be greater than or equal to current date.
      - When Start date and End Date is the current date, the start time should be greater than the current

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

time.
- Enter null in the target start and end date to run the event daily.

Update

You can use the settings in this tab when you install the same Package for the consecutive time.

- **Delete Event Association**: Select this option to delete the Event Association that is present in the target environment.
  - **Yes**: Select this option to remove the Event Association in the target environment.
  - **No**: Select this option to retain the Event Association in the target environment or install the Event Association if it is not present in the target environment.

  Note: The settings configured in the **Workflow Version** and **Scheduler Event Association Settings** grids are common for the **Install** and **Update** tabs.

Uninstall

You can refer to the settings in this tab when you uninstall the Package from the target environment.

- **Delete Event Association**: This property allows you to retain or delete the Event Association of the type Scheduler in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to delete the Event Association from the target environment when the Package is uninstalled.
  - **No**: Select this option to retain the Event Association in the target environment even after the Package is uninstalled.

Click **Save Event Association Settings**.

Note: You cannot uninstall individual versions, when uninstalling a package. All the versions will be uninstalled and a message is logged for every single version that is uninstalled.

Associated Artifacts Tab

The **Associated Artifacts** tab displays all the associated artifacts of the selected Event Associations of type Scheduler definition.

Event Associations of type Scheduler has the following associated artifacts: Workflow

Child Artifacts Tab

Event Associations of type Scheduler does not have any child artifact.

Security Artifacts Tab

The Security Artifacts tab displays the security artifacts of an Event Associations of type Scheduler, if any.

You can enable the security settings of the Event Associations of type Scheduler that you want to include in the package.
For more information about including security artifacts, see Include Security Artifacts.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Event Associations of Type Enterprise Console

You can package and deploy the Event Associations of type Enterprise Console and its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add Event Associations of Type Enterprise Console to Package Template

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, click **Workflows**. The **Workflows** window appears.
3. Select a Workflow with an event association of type enterprise console associated to it.
4. Right-click the workflow, and then select **View Association**. All associations of the selected workflow appears.
5. Right-click Event Associations of type enterprise console that you want to package, and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Note: If you create an Event Association of type Enterprise Console for a dynamic list item and if you delete the list item, then you cannot add the Enterprise Console Event Association to the package template.

Package Template Tree View

In the Package Template tree view, the Event Association of type Enterprise Console appears in the following format:

Format in the tree view in case of List level

<Workflow Name_EC_Event Name_Type_List Name>

Here,

Workflow Name: Name of the workflow to which the Enterprise Console event is associated

EC: Name of the Event Provider, in this case it is the Enterprise Console

Event Names: On Create, On Delete, On Publish, On Unpublish, and On Version Change

Type: List

List Name: Name of the list which is associated to the workflow

The same information is also seen in the Configuration Settings.

Format in the tree view in case of List item level

<**Workflow Name_EC_Event Name_Type_List Name_List item Name/Unique Values**>

Here,

Workflow Name: Name of the workflow to which the Enterprise Console event is associated

EC: Name of the Event Provider, in this case it is the Enterprise Console

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Event Names: On Create, On Delete, On Publish, On Unpublish, and On Version Change

Type: List or List Item.

List Name: Name of the list in which the associated list item is present

List Item Name/Unique Values: Default value of the list item or the Unique field values of the list items. If the list item has multiple unique values defined, then each of the value is displayed separated by (_).

The same information is also seen in the Configuration Settings.

Associated Artifacts for Event Associations of Type Enterprise Console

Event Associations of type Enterprise Console has the following associated artifacts:

- Workflow
- Dynamic List (if the dynamic list is selected as the list name while creating Enterprise Console Event Association. Users list is also considered as a dynamic list.)

In case of Enterprise Console Event Association of list item level, along with the above two associated artifact, the following additional artifacts are identified:

- List items of the Standard list
- List item of Dynamic list (these items will not be listed as an individual artifact in the package template tree view)

Dynamic List item which is associated to Enterprise Console at list item level Event Association will not appear in the Package template as an individual artifact even though it is identified as an associated artifact for the Event Association. However, the same will be listed in the Associated Artifacts tab as an associated artifact of Enterprise Console at list item level Event Association. This occurs since Dynamic list item is not a packageable artifact. Therefore, to install Enterprise Console at list item level Event Association, the dynamic list item should be available in the target environment.

Enterprise Console Event Association at list item level, will package the following items if "include associated artifacts" is set to true in package template:

- DB connection or Connection List
- Lookup list
- Forms Data Grid View
- Grid Configuration
- Report Connection
- Report Template
- Reports
- FileWatcher List
- Form Controls
- Navigation details
- DB connection
- Lookup list
- List Views

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Forms List
- Assembly List
- Folder List
- Security Right
- Security Group

Enterprise Console Event Association package will identify the items of following lists as associated artifacts and will not add to package template. However, in target if the item with the same title or default field value is available in the same list, then event association will be installed by mapping to the identified list item.

- Manage Package
- Package Template
- Impersonation Users
- Initiate form
- External Connections
- Users
- List-Form Association List

Enterprise Console Event association package is not supported for the following list items:

- Deleted Items
- Event Providers
- Security Group Resource mapping
- Event Association

Child Artifacts for Event Associations of Type Enterprise Console

Event Associations of type Enterprise Console does not have any child artifact.

Security Artifacts for Event Associations of Type Enterprise Console

You can enable the security settings of the grid view that you want to include in the package.
For more information about including security artifacts, see Include Security Artifacts.

Configuration settings for Event Associations of Type Enterprise Console

The Event Associations of type Enterprise Console has an unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the Settings tab to define the configuration settings. The **Associated Artifacts** tab displays the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

associated artifacts of the Event Association of type Enterprise Console definition. The **Child Artifacts** tab displays child artifacts of the Event Association of type Enterprise Console definition.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Security Artifacts

Settings Tab

The properties in this tab allows you to take appropriate actions while installing, updating, or uninstalling the Package.

Behavior of Event Associations of type Enterprise Console while installing, updating or uninstalling the package:

- You cannot overwrite an already existing item.
- You can delete or uninstall an Event Associations of type Enterprise Console only if it was installed by the Package.

Install

The settings in this tab is referred to when installing the Package. When installing the package, if the Event Enterprise Console dates are invalid, then the package will not be installed. Edit the dates and install the package.

- **Workflow Version**: This displays the version of the Workflow to which Event Association of type Enterprise Console is mapped in the Source environment and it also allows you to specify the version of Workflow to which Event Association of type Enterprise Console should be mapped in the Target environment.

  Source: Displays the version of Workflow in the source environment to which the Event Association of type Enterprise Console is mapped.

  Target: Enter the version of the workflow.

  - Specify the version of the Workflow to which you want to map the Event Association of type Enterprise Console in target environment.
  - If you do not specify the Workflow version in the Target environment, the Workflow version from the Source environment will be considered while installing the package.

  If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is not present in the target environment, then the Event Association will not be installed.

  If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is in Draft status, then the Event Association of type Enterprise Console will be installed and mapped to the latest version of the workflow.
- **Enterprise Console Event Association Settings**: This displays the settings of the Event Association of type Enterprise Console in the Source environment and it also allows you to specify the settings of the Event Association of type Enterprise Console in the Target environment.

  Source

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Effective From**: Displays the date from when the Event should be triggered in the source environment.
- **Effective To**: Displays the date till when the Event should be triggered in the source environment.

Target

- **Effective From**: Select the date from when the Event should be triggered in the target environment.

  If you are not using the date picker ensure that the date is in the MM\dd\yyyy format.
- **Effective To**: Select the date till when the Event should be triggered in the target environment.

  If you are not using the date picker ensure that the date is in the MM\dd\yyyy format.

Note: Ensure that you follow the following guidelines while packaging the Event Associations of type Enterprise Console in the target environment:

- Effective From and Effective To should be in a valid format.
- Effective From should be less than or equal to Effective To respectively.
- Effective To should be greater than or equal to current date.
- Effective From and Effective To can be 'Null'. However, you cannot have either one of them as 'Null'.

If you want to leave the dates as blank, then enter **Null** in the Effective From and Effective To fields in the Target section.

Update

The settings in this tab is referred to when the same Package is installed for the consecutive time.

- **Delete Event Association**: Select this option to delete the Event Association that is present in the target environment.
  - **Yes**: Select this option to remove the Event Association in the target environment.
  - **No**: Select this option to retain the Event Association in the target environment, or install the Event Association if it is not present in the target environment.
- **Workflow Version**: This displays the version of Workflow to which Event Association of type Enterprise Console is mapped in the Source environment and it also allows you to specify the version of Workflow to which Event Association of type Enterprise Console should be mapped in the Target environment:

  Source: Displays the version of Workflow in the source environment to which the Event Association of type Enterprise Console is mapped.

  Target: Enter the version of the workflow:
  - Specify the version of the Workflow to which you want to map the Event Association of type Enterprise Console in target environment.
  - If you do not specify the Workflow version in the Target environment, the Workflow version from the Source environment will be considered while installing the package.

    If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is not present in the target environment, then the Event Association will not be installed.

    If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is in Draft status, then the Event Association of type Enterprise Console will be installed and mapped to the latest version of the workflow.
- **Enterprise Console Event Association Settings**: This displays the settings of the Event Association of the type Enterprise Console in the Source environment and also allows you to specify the settings of the Event

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Association of the type Enterprise Console in the Target environment while updating the Package.

Source

- **Effective From**: Displays the date from when the Event should be triggered in the source environment.
- **Effective To**: Displays the date till when the Event should be triggered in the source environment.

Target

- **Effective From**: Update the date from when the Event should be triggered in the target environment.

  If you are not using the date picker ensure that the date is in the MM\dd\yyyy format.
- **Effective To**: Update the date till when the Event should be triggered in the target environment.

  If you are not using the date picker ensure that the date is in the MM\dd\yyyy format.

Uninstall

This settings in this tab is referred to when the Package is being uninstalled in the target environment.

- **Delete Event Association**: This property allows you to retain or delete the Event Association of the type Enterprise Console in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to delete the Event Association from the target environment when the Package is uninstalled.
  - **No**: Select this option to retain the Event Association in the target environment even after the Package is uninstalled.

Click **Save Event Association Settings**.

Associated Artifacts Tab

The Associated Artifacts tab displays all the associated artifacts of the selected Event Associations of type Enterprise Console definition.

Child Artifacts Tab

Event Associations of type Enterprise Console does not have any child artifact.

Security Artifacts Tab

The Security Artifacts tab displays the security artifacts of an Event Associations of type Enterprise Console, if any.

Event Associations of Type File Watcher

You can package and deploy the Event Associations of type File Watcher along with its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To add Event Association of Type File Watcher to Package Template

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, click **Workflows**. The **Workflows** window appears.
3. Select a Workflow with an event association of type file watcher associated to it.
4. Right-click the workflow, and then select **View Association**. All associations of the selected workflow appear.
5. Right-click Event Associations of type file watcher that you want to package, and then click **Add to Package Template**.
   The **Add to Package Template** window appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

In the Package Template tree view, the File Watcher Event Association will appear in the following format:

<**Workflow Name_Provider Name_Event Name_Path_File Name/Pattern**>

Here,

- **Workflow Name**: Name of the workflow to which the file watcher is associated
- **Provider Name**: Name of the file watcher
- **Event Names**: Created, deleted, changed, or renamed
- **Path**: Physical path of the file watcher in the source environment
- **File Name/Pattern**: Extension of the file or a name of the file. For example, .txt or .doc file.

Configuration Settings for Event Association of Type File Watcher

The Event Associations of type File Watcher has a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the Settings tab to define the configuration settings. The Associated Artifacts tab displays the associated artifacts of the Event Association of type File Watcher definition. The Child Artifacts tab displays child artifacts of the Event Association of type File Watcher definition.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Security Artifacts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Tab

The properties in this tab allow you to take appropriate actions while installing, updating, or uninstalling the Package.

Behavior of File Watcher Event Association while installing, updating or uninstalling the package:

- You cannot overwrite an already existing item.
- You can delete or uninstall a File Watcher Event Association only if it was installed by the Package.

Install

The settings in this tab are referred to when installing the Package. When installing the package, if the Event File Watcher dates are invalid, then the package will not be installed. Edit the dates and install the package.

- **Workflow Version**: This displays the version of the Workflow to which Event Association of type File Watcher is mapped in the Source environment and it also allows you to specify the version of Workflow to which Event Association of type File Watcher should be mapped in the Target environment.
  - **Source**: Displays the version of Workflow in the source environment to which the Event Association of type File Watcher is mapped.
  - **Target**: Enter the version of the workflow:
    - Specify the version of the Workflow to which you want to map the Event Association of type File Watcher in target environment.
    - If you do not specify the Workflow version in the Target environment, the Workflow version from the Source environment will be considered while installing the package.

      Note:
      - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is not present in the target environment, then the Event Association will not be installed.
      - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is in Draft status, then the Event Association of type File Watcher will be installed and mapped to the
      latest version of the workflow.

- **File Watcher Event Association Settings**: This displays the settings of the Event Association of type File Watcher in the Source environment and it also allows you to specify the settings of the Event Association of type File Watcher in the Target environment.
  - **Source**
    - **Event Folder**: Displays the physical path of File Watcher in the source environment.
    - **Effective From**: Displays the date from when the Event should be triggered in the source environment.
    - **Effective To**: Displays the date till when the Event should be triggered in the source environment.
  - **Target**
    - **Event Folder**: Enter the physical path of the File Watcher in the target environment. If this field is left blank then the physical path for the target environment will be the same as the path specified in source environment.

      Note: Ensure that the path specified in this field and the path specified in the Physical Path of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

associated File Watcher configuration settings is the same. If it is not, then the File watcher Event Association will not be mapped to the File Watcher.

- **Effective From**: Select the date from when the Event should be triggered in the target environment.

  If you are not using the date picker ensure that the date is in the MM\dd\yyyy format.
- **Effective To**: Select the date till when the Event should be triggered in the target environment.

  If you are not using the date picker ensure that the date is in the MM\dd\yyyy format.

Note:

- Ensure that you follow the following guidelines while packaging the File Watcher Event Association in the target environment:
- Effective From and Effective To should be in a valid format.
- Effective From should be less than or equal to Effective To respectively.
- Effective To should be greater than or equal to current date.
- Effective From and Effective To can be 'Null'. However, you cannot have either one of them as 'Null'.
- If you want to leave the dates as blank, then you have to enter 'Null' in the Effective From and Effective To fields in the Target section.

Update

The settings in this tab are referred to when the same Package is installed for the consecutive time.

- **Delete Event Association**: Select this option to delete the Event Association that is present in the target environment.
  - **Yes**: Select this option to remove the Event Association in the target environment.
  - **No**: Select this option to retain the Event Association in the target environment or install the Event Association if it is not present in the target environment.
- **Workflow Version**: This displays the version of Workflow to which Event Association of type File Watcher is mapped in the Source environment and it also allows you to specify the version of Workflow to which Event Association of type File Watcher should be mapped in the Target environment.
  - **Source**: Displays the version of Workflow in the source environment to which the Event Association of type File Watcher is mapped.
  - **Target**: Enter the version of the workflow:

    Specify the version of the Workflow to which you want to map the Event Association of type File Watcher in target environment.

    If you do not specify the Workflow version in the Target environment, the Workflow version from the Source environment will be considered while installing the package.

    Note:
    - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is not present in the target environment, then the Event Association will not be installed.
    - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is in Draft status, then the Event Association of type File Watcher will be installed and mapped to the latest version of the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **File Watcher Event Association Settings**: This displays the settings of the Event Association of the type File Watcher in the Source environment and also allows you to specify the settings of the Event Association of the type File Watcher in the Target environment while updating the Package.
  - **Source**
    - **Event Folder**: Displays the physical path of File Watcher in the source environment.
    - **Effective From**: Displays the date from when the Event should be triggered in the source environment.
    - **Effective To**: Displays the date till when the Event should be triggered in the source environment.
  - **Target**
    - **Event Folder**: Enter the physical path of the File Watcher in the target environment. If this field is left blank then the physical path for the target environment will be the same as the path specified in source environment.
    - **Effective From**: Update the date from when the Event should be triggered in the target environment.

      If you are not using the date picker ensure that the date is in the MM\dd\yyyy format.
    - **Effective To**: Update the date till when the Event should be triggered in the target environment.

      If you are not using the date picker ensure that the date is in the MM\dd\yyyy format.

Uninstall

The settings in this tab are referred to when the Package is being uninstalled in the target environment.

- **Delete Event Association**: This property allows you to retain or delete the Event Association of the type File Watcher in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to delete the Event Association from the target environment when the Package is uninstalled.
  - **No**: Select this option to retain the Event Association in the target environment even after the Package is uninstalled.

Click **Save Event Association Settings**.

Note: You cannot uninstall individual versions, when uninstalling a package. All the version will be uninstalled and a message is logged for every single version that is uninstalled.

Associated Artifacts

The **Associated Artifacts** tab displays all the associated artifacts of the selected Event Association of type File Watcher definition.

File Watcher Event Association has the following associated artifacts:

- Workflow
- File Watcher: While installing the package we will verify if the file watcher with the same path as specified in the configuration settings is available in the target environment. If available, then the File Watcher Event Association will be installed by mapping to the identified (one with the same path) file watcher. If it is not available then the File Watcher Event Association will not be installed.
  If the File Watcher title is same and path is different than the one specified in the configuration settings of the File Watcher Event Association, then the File Watcher Event Association will not be installed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Child Artifacts

Event Association of type File Watcher does not have any child artifact.

Security Artifacts

The **Security Artifacts** tab displays the security artifacts of an Event Association of type File Watcher, if any.

You can enable the security settings of the grid view that you want to include in the package.
For more information about including security artifacts, see Include Security Artifacts.

Event Associations of Type System Platform Events

You can package and deploy the Event Associations of type System Platform Events along with workflow and its associations, from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

Consider the following scenario where:

- An environment is set up with System Platform and AVEVA Work Tasks.
- A WorkflowGateway object is configured and the repository is provisioned.
- An object in System Platform has an event and this event is associated with a workflow in AVEVA Work Tasks.

When this System Platform event is triggered, the associated workflow is executed.

To make the above scenario work in the target environment, perform the following in the source and target environments.

Source Environment

- Export the object and its dependent objects from System Platform.
- Package the Workflow with System Platform Events in AVEVA Work Tasks , and then export the package.

Target Environment

- Import the object and its dependent objects to System Platform.

Note:
- Ensure that the WorkflowGateway object name is same in the target environment.
- You can map multiple event associations for the same workflow. If you package workflows with multiple System Platform Event Associations and install this package in the target environment, then only the first event association mapping will be retained. You have to manually map all the other event associations for the same workflow after the package is installed.

- Import the package to AVEVA Work Tasks and then install the package.

  For more information about importing a package, see Importing a Package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Adding Event Associations of type System Platform Events to Package Template

To package Event Associations through Workflows

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, and then click **Workflows**. The **Workflows** window appears.
3. Right-click the workflow, and then click **Add to Package Template**. The **Add to Package Template** dialog box appears.
4. In the **Include Associated Artifacts** section, select **Yes**.
5. Click **Save Template**.
   The System Platform events along with the workflow are included in the package.

To package Workflows through Event Association

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, and then click **Workflows**. The **Workflows** window appears.
3. Select a Workflow with an event association of type System Platform Events.
4. Right-click the workflow, and then select **View Association**. All associations of the selected workflow appear.
5. Right-click **Event Associations of type System Platform Events** that you want to package, and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Runtime Initialization Settings

You can configure runtime initialization settings for Event Associations of type System Platform Events. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration Settings for Event Associations of type System Platform Events

The Event Associations of type System Platform have a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the Settings tab to define the configuration settings. The **Associated Artifacts** tab displays the associated artifacts of the Event Association of type System Platform Events definition. The **Child Artifacts** tab displays child artifacts of the Event Association of type System Platform Events definition.

The Package Template configuration window contains the following tabs:

- Settings

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Install
- Update
- Uninstall

- Associated Artifacts
- Security Artifacts

Settings Tab

You can use the properties of the Settings tab to take appropriate actions to install, update, or uninstall the package.

Behavior of Event Associations of type System Platform Events while installing, updating or uninstalling the package:

- You cannot overwrite an already existing item.
- You can delete or uninstall an Event Associations of type System Platform Event only if it was installed by the Package.

Install

You can use the settings in this tab while installing the package.

- The following event association details are displayed:
  - **Type**: The type of event. In this scenario, the type is displayed as **ArchestrA**.
  - **Define in Object**: The System Platform object to which this event is associated.
  - **Event**: The name of the event.
  - **Workflow**: The workflow associated to the event.
- **Workflow Version**: This displays the version of Workflow to which Event Association of type System Platform Events is mapped in the Source environment. It also allows you to specify the version of Workflow to which Event Association of type System Platform should be mapped in the Target environment.
  - **Source**: Displays the version of Workflow in the source environment to which the Event Association of type System Platform Events is mapped.

    Target: Enter the version of the workflow:

    Specify the version of the Workflow to which you want to map the Event Association of type System Platform Events in target environment.

    If you do not specify the Workflow version in the Target environment, the Workflow version from the Source environment will be considered while installing the package.

    Note:
    - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is not present in the target environment, then the Event Association will not be installed.
    - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is in Draft status, then the Event Association of type System Platform Events will be installed and mapped to the latest version of the workflow.

- **Galaxy Configuration Settings**: This displays the galaxy configuration settings in the Source environment and

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

it also allows you to specify the galaxy configuration settings of the Event Association of type System Platform Events in the target environment.

- **Unique Environment Name**
  - **Source**: Displays the unique environment name in the source environment.
  - **Target**: Enter or select the unique environment name in the target environment.
- **Galaxy Name**
  - **Source**: Displays the galaxy name in the source environment.
  - **Target**: Select the galaxy name in the target environment.

Update

You can refer to the settings in this tab when you install the same Package for the consecutive time.

- **Delete Event Association**: Select this option to delete the Event Association that is present in the target environment.
  - **Yes**: Select this option to remove the Event Association in the target environment.
  - **No**: Select this option to retain the Event Association in the target environment or to install the Event Association if it is not present in the target environment.

  Note: The settings configured in the **Workflow Version** and **Galaxy Configuration Settings** grids are common for the **Install** and **Update** tabs.

Uninstall

You can refer to the settings in this tab when you uninstall the Package from the target environment.

- **Delete Event Association**: This property allows you to retain or delete the Event Association of the type System Platform Events in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to delete the Event Association from the target environment when the Package is uninstalled.
  - **No**: Select this option to retain the Event Association in the target environment even after the Package is uninstalled.

Click **Save Event Association Settings**.

Note: You cannot uninstall individual versions, when uninstalling a package. All the versions will be uninstalled and a message is logged for every single version that is uninstalled.

Associated Artifacts Tab

The **Associated Artifacts** tab displays all the associated artifacts of the selected Event Associations of type System Platform Events definition.

Event Associations of type System Platform Events has the following associated artifacts: Workflow

Child Artifacts Tab

Event Associations of type System Platform Events does not have any child artifact.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Security Artifacts Tab

The Security Artifacts tab displays the security artifacts of an Event Associations of type System Platform Events, if any.

You can enable the security settings of the Event Associations of type System Platform Events that you want to include in the package.
For more information about including security artifacts, see Include Security Artifacts.

Event Associations of Type Logic App

You can package and deploy the Event Associations of type Logic App along with its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add Event Association of Type Logic App to Package Template

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, click **Workflows**. The **Workflows** window appears.
3. Select a Workflow with an event association of type Logic App.
4. Right-click the workflow, and then select **View Association**. All associations of the selected workflow appear.
5. Right-click Event Associations of type Logic App that you want to package, and then click **Add to Package Template**.
   The **Add to Package Template** window appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

In the Package Template tree view, the Logic App Event Association will appear in the following format:

<**LogicAppAssociation_Workflow Name**>

Here,

- **LogicAppAssociation**: Name of the Logic App Association
- **Workflow Name**: Name of the workflow to which the Logic App is associated

Configuration Settings for Event Association of Type Logic App

The Event Associations of type Logic App has a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the Settings tab to define the configuration settings. The Associated Artifacts tab displays the associated artifacts of the Event Association of type Logic App definition. The Child Artifacts tab displays child artifacts of the Event Association of type Logic App definition.

The Package Template configuration window contains the following tabs:

- Settings

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Security Artifacts

Settings Tab

The properties in this tab allow you to take appropriate actions while installing, updating, or uninstalling the Package.

Behavior of Logic App Event Association while installing, updating or uninstalling the package:

- You cannot overwrite an already existing item.
- You can delete or uninstall a Logic App Event Association only if it was installed by the Package.

Install

The settings in this tab are referred to when installing the Package.

- **Workflow Version**: This displays the version of the Workflow to which Event Association of type Logic App is mapped in the Source environment and it also allows you to specify the version of Workflow to which Event Association of type Logic App should be mapped in the Target environment.
  - **Source**: Displays the version of Workflow in the source environment to which the Event Association of type Logic App is mapped.
  - **Target**: Enter the version of the workflow:
    - Specify the version of the Workflow to which you want to map the Event Association of type Logic App in target environment.
    - If you do not specify the Workflow version in the Target environment, the Workflow version from the Source environment will be considered while installing the package.

      Note:
      - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is not present in the target environment, then the Event Association will not be installed.
      - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is in Draft status, then the Event Association of type Logic App will be installed and mapped to the latest version of the workflow.

- **Logic App Event Association Settings**: This displays the settings of the Event Association of type Logic App in the Source environment and it also allows you to specify the settings of the Event Association of type Logic App in the Target environment.
  - **Source**
    - **Enterprise Console**: Displays the name of the Enterprise Console site with the Server name in the Source environment.
    - **Logic App URI**: Displays the automatically generated URI based on the selected Enterprise Console site in the Source environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Logic App Alert Key**: Displays the automatically generated value of the alert key.
- **Authorization Token**: Displays the automatically generated authorization token based on the selected Enterprise Console site in the Source environment.
- **Workflow Version**: Displays the version of the Workflow to which Event Association of type Logic App is mapped in the Source environment.

- **Target**
  - **Enterprise Console**: Enter the name of the Enterprise Console site with the Server name or select the Enterprise Console site that is fetched by run time initialization control in the Target environment.
  - **Logic App URI**: Enter the automatically generated URI of the selected Enterprise Console site or select the URI generated by run time initialization control in the Target environment.
  - **Logic App Alert Key**: Enter the automatically generated value of the Logic App alert key or select the key generated by run time initialization control in the Target environment.
  - **Authorization Token**: Enter the automatically generated authorization token of the selected Enterprise Console site or select the token generated by run time initialization control in the Target environment.
  - **Workflow Version**: Enter the version of the Workflow to which Event Association of type Logic App is mapped in the Target environment.

Update

The settings in this tab are referred to when the same Package is installed for the consecutive time.

- **Delete Event Association**: Select this option to delete the Event Association that is present in the target environment.
  - **Yes**: Select this option to remove the Event Association in the target environment.
  - **No**: Select this option to retain the Event Association in the target environment or install the Event Association if it is not present in the target environment.
- **Workflow Version**: This displays the version of Workflow to which Event Association of type Logic App is mapped in the Source environment and it also allows you to specify the version of Workflow to which Event Association of type Logic App should be mapped in the Target environment.
  - **Source**: Displays the version of Workflow in the source environment to which the Event Association of type Logic App is mapped.
  - **Target**: Enter the version of the workflow:

    Specify the version of the Workflow to which you want to map the Event Association of type Logic App in target environment.

    If you do not specify the Workflow version in the Target environment, the Workflow version from the Source environment will be considered while installing the package.

    Note:
    - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is not present in the target environment, then the Event Association will not be installed.
    - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is in Draft status, then the Event Association of type Logic App will be installed and mapped to the latest version of the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Logic App Event Association Settings**: This displays the settings of the Event Association of the type Logic App in the Source environment and also allows you to specify the settings of the Event Association of the type Logic App in the Target environment while updating the Package.
  - **Source**
    - **Enterprise Console**: Displays the name of the Enterprise Console site with the Server name in the Source environment.
  - **Target**
    - **Enterprise Console**: Enter the name of the Enterprise Console site with the Server name or select the Enterprise Console site that is fetched by run time initialization control in the Target environment. If this field is left blank then the name of Enterprise Console site for the target environment will be the same as the name of Enterprise Console site in source environment.

Uninstall

The settings in this tab are referred to when the Package is being uninstalled in the target environment.

- **Delete Event Association**: This property allows you to retain or delete the Event Association of the type Logic App in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to delete the Event Association from the target environment when the Package is uninstalled.
  - **No**: Select this option to retain the Event Association in the target environment even after the Package is uninstalled.

Click **Save Event Association Settings**.

Note: You cannot uninstall individual versions, when uninstalling a package. All the version will be uninstalled and a message is logged for every single version that is uninstalled.

Associated Artifacts

The **Associated Artifacts** tab displays all the associated artifacts of the selected Event Association of type Logic App definition.

Logic App Event Association has the following associated artifacts:

- Workflow
- Logic App: While installing the package we will verify if the Logic App with the same path as specified in the configuration settings is available in the target environment. If available, then the Logic App Event Association will be installed by mapping to the identified (one with the same path) Logic App. If it is not available then the Logic App Event Association will not be installed.
  If the Logic App title is same and path is different from the one specified in the configuration settings of the Logic App Event Association, then the Logic App Event Association will not be installed.

Child Artifacts

Event Association of type Logic App does not have any child artifact.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Security Artifacts

The **Security Artifacts** tab displays the security artifacts of an Event Association of type Logic App, if any.

You can enable the security settings of the grid view that you want to include in the package.
For more information about including security artifacts, see Include Security Artifacts.

Event Associations of Type Message Queue

You can package and deploy the Event Associations of type Message Queue along with its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add Event Association of Type Message Queue to Package Template

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, click **Workflows**. The **Workflows** window appears.
3. Select a Workflow with an event association of type Message Queue.
4. Right-click the workflow, and then select **View Association**. All associations of the selected workflow appear.
5. Right-click Event Associations of type Message Queue that you want to package, and then click **Add to Package Template**.
   The **Add to Package Template** window appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

In the Package Template tree view, the Message Queue Event Association will appear in the following format:

<**Message Queue_Workflow Name**>

Here,

- **Message Queue_Workflow Name**: Name of the Message Queue Association

Configuration Settings for Event Association of Type Message Queue

The Event Associations of type Message Queue has a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the Settings tab to define the configuration settings. The Associated Artifacts tab displays the associated artifacts of the Event Association of type Message Queue definition. The Child Artifacts tab displays child artifacts of the Event Association of type Message Queue definition.

The Package Template configuration window contains the following tabs:

- Settings

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Security Artifacts

Settings Tab

The properties in this tab allow you to take appropriate actions while installing, updating, or uninstalling the Package.

Behavior of Message Queue Event Association while installing, updating or uninstalling the package:

- You cannot overwrite an already existing item.
- You can delete or uninstall a Message Queue Event Association only if it was installed by the Package.

Install

The settings in this tab are referred to when installing the Package.

- **Workflow Version**: This displays the version of the Workflow to which Event Association of type Message Queue is mapped in the Source environment and it also allows you to specify the version of Workflow to which Event Association of type Message Queue should be mapped in the Target environment.
  - **Source**: Displays the version of Workflow in the source environment to which the Event Association of type Message Queue is mapped.
  - **Target**: Enter the version of the workflow:
    - Specify the version of the Workflow to which you want to map the Event Association of type Message Queue in target environment.
    - If you do not specify the Workflow version in the Target environment, the Workflow version from the Source environment will be considered while installing the package.

      Note:
      - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is not present in the target environment, then the Event Association will not be installed.
      - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is in Draft status, then the Event Association of type Message Queue will be installed and mapped to the latest version of the workflow.

- **Message Queue Event Association Settings**: This displays the settings of the Event Association of type Message Queue in the Source environment and it also allows you to specify the settings of the Event Association of type Message Queue in the Target environment.
  - **Source**
    - **Workflow Version**: Displays the version of the Workflow to which Event Association of type Message Queue is mapped in the Source environment.
  - **Target**
    - **Workflow Version**: Enter the version of the Workflow to which Event Association of type Message

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Queue is mapped in the Target environment.

Update

The settings in this tab are referred to when the same Package is installed for the consecutive time.

- **Delete Event Association**: Select this option to delete the Event Association that is present in the target environment.
  - **Yes**: Select this option to remove the Event Association in the target environment.
  - **No**: Select this option to retain the Event Association in the target environment or install the Event Association if it is not present in the target environment.
- **Workflow Version**: This displays the version of Workflow to which Event Association of type Message Queue is mapped in the Source environment and it also allows you to specify the version of Workflow to which Event Association of type Message Queue should be mapped in the Target environment.
  - **Source**: Displays the version of Workflow in the source environment to which the Event Association of type Message Queue is mapped.
  - **Target**: Enter the version of the workflow:

    Specify the version of the Workflow to which you want to map the Event Association of type Message Queue in target environment.

    If you do not specify the Workflow version in the Target environment, the Workflow version from the Source environment will be considered while installing the package.

    Note:
    - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is not present in the target environment, then the Event Association will not be installed.
    - If the version of the workflow that is considered while installing the package (based on the two conditions specified above) is in Draft status, then the Event Association of type Message Queue will be installed and mapped to the latest version of the workflow.

- **Message Queue Event Association Settings**: This displays the settings of the Event Association of the type Message Queue in the Source environment and also allows you to specify the settings of the Event Association of the type Message Queue in the Target environment while updating the Package.
  - **Source**
    - **Workflow Version**: Displays the version of the Workflow to which Event Association of type Message Queue is mapped in the Source environment.
  - **Target**
    - **Workflow Version**: Enter the version of the Workflow to which Event Association of type Message
    - Queue is mapped in the Target environment.

Uninstall

The settings in this tab are referred to when the Package is being uninstalled in the target environment.

- **Delete Event Association**: This property allows you to retain or delete the Event Association of the type Message Queue in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to delete the Event Association from the target environment when the Package is uninstalled.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **No**: Select this option to retain the Event Association in the target environment even after the Package is uninstalled.

Note: You cannot uninstall individual versions, when uninstalling a package. All the version will be uninstalled and a message is logged for every single version that is uninstalled.

Associated Artifacts

The **Associated Artifacts** tab displays all the associated artifacts of the selected Event Association of type Message Queue definition.

Message Queue Event Association has the following associated artifacts:

- Workflow
- Message Queue Connection: While installing the package we will verify if the Message Queue with the same path as specified in the configuration settings is available in the target environment. If available, then the Message Queue Event Association will be installed by mapping to the identified (one with the same path) Message Queue. If it is not available then the Message Queue Event Association will not be installed.
  If the Message Queue title is same and path is different from the one specified in the configuration settings of the Message Queue Event Association, then the Message Queue Event Association will not be installed.

Child Artifacts

Event Association of type Message Queue does not have any child artifact.

Security Artifacts

The **Security Artifacts** tab displays the security artifacts of an Event Association of type Message Queue, if any.

You can enable the security settings of the grid view that you want to include in the package.
For more information about including security artifacts, see Include Security Artifacts.

Event Association of Type BAM

You can package and deploy the Event Associations of type BAM along with its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add Event Association of Type BAM to Package Template

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, click **Workflows**. The **Workflows** window appears.
3. Select a Workflow with an event association of type BAM associated to it.
4. Right-click the workflow, and then select **View Association**. All associations of the selected workflow appear.
5. Right-click Event Associations of type BAM that you want to package, and then click **Add to Package Template**.
   The **Add to Package Template** window appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

In the Package Template tree view, the BAM Event Association will appear in the following format:

<**Workflow Name_Associated Event Name_Associated Event Type**>

Here,

- **Workflow Name**: Name of the workflow to which BAM is associated
- **Associated Event Name**: Name of the BAM event associated
- **Associated Event Type**: Name of the BAM event type i.e. Overdue Analysis or KPI

Configuration Settings for Event Association of Type BAM

The Event Associations of type BAM has a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the Settings tab to define the configuration settings. The Associated Artifacts tab displays the associated artifacts of the Event Association of type BAM definition. The Child Artifacts tab displays child artifacts of the Event Association of type BAM definition.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Security Artifacts

Settings Tab

The properties in this tab allow you to take appropriate actions while installing, updating, or uninstalling the Package.

Behavior of BAM Event Association while installing, updating, or uninstalling the package:

- You cannot overwrite an already existing item.
- You can delete or uninstall a BAM Event Association only if it was installed by the Package.

Install

The settings in this tab are referred to when installing the Package. When installing the package, if the Event BAM dates are invalid, then the package will not be installed. Edit the dates and install the package.

- The following event association details are displayed:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Type**: The type of event. In this scenario, type is Overdue Analysis or KPI.
- **Associated Workflow**: The workflow associated to the BAM event type.
- **Workflow**: The workflow associated to the event.

- **BAM Event Association Settings**: This displays the settings of the Event Association of type BAM in the Source environment and it also allows you to specify the settings of the Event Association of type BAM in the Target environment.
  - **Source**
    - **Effective From**: Displays the date from when the Event should be triggered in the source environment.
    - **Effective To**: Displays the date till when the Event should be triggered in the source environment.
  - **Target**
    - **Effective From**: Select the date from when the Event should be triggered in the target environment.

      If you are not using the date picker ensure that the date is in the MM\dd\yyyy format.
    - **Effective To**: Select the date till when the Event should be triggered in the target environment.

      If you are not using the date picker ensure that the date is in the MM\dd\yyyy format.

      Note:
      Ensure that you follow the following guidelines while packaging the BAM Event Association in the target environment:
      - Effective From and Effective To should be in a valid format.
      - Effective From should be less than or equal to Effective To respectively.
      - Effective To should be greater than or equal to current date.
      - Effective From and Effective To can be 'Null'. However, you cannot have either one of them as 'Null'.
      - If you want to leave the dates as blank, then you have to enter 'Null' in the Effective From and Effective To fields in the Target section.

Update

The settings in this tab are referred to when the same Package is installed for the consecutive time.

- **Delete Event Association**: Select this option to delete the Event Association that is present in the target environment.
  - **Yes**: Select this option to remove the Event Association in the target environment.
  - **No**: Select this option to retain the Event Association in the target environment or install the Event Association if it is not present in the target environment. By default, **No** is selected.

Uninstall

The settings in this tab are referred to when the Package is being uninstalled in the target environment.

- **Delete Event Association**: This property allows you to retain or delete the Event Association of the type BAM in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to delete the Event Association from the target environment when the Package is uninstalled. By default, **Yes** is selected.
  - **No**: Select this option to retain the Event Association in the target environment even after the Package is uninstalled.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Click **Save Event Association Settings**.

Note: You cannot uninstall individual versions, when uninstalling a package. All the version will be uninstalled and a message is logged for every single version that is uninstalled.

Associated Artifacts

The **Associated Artifacts** tab displays all the associated artifacts of the selected Event Association of type BAM definition.

BAM Event Association has the following associated artifacts:

- Workflow
- BAM

Child Artifacts

Event Association of type BAM does not have any child artifact.

Security Artifacts

The **Security Artifacts** tab displays the security artifacts of an Event Association of type BAM, if any.

You can enable the security settings of the grid view that you want to include in the package.
For more information about including security artifacts, see Include Security Artifacts.

File Watcher

You can package and deploy the File Watcher along with its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add File Watcher to a Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **File Watcher**.
   The **FileWatcher List** window appears.
3. Select the file watcher list that you want to package.
4. Right-click the file watcher list, and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Runtime Initialization Settings

You can configure runtime initialization settings for the File Watcher. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration settings for File Watcher

The **File Watcher** has a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment** > **Package Template** window
- **Target Environment** > **Manage Package** window

You can use the **Settings** tab to define the configuration settings. The File Watcher does not have any associated artifact or child artifact. So, the **Associated Artifacts** and **Child Artifacts** tab are not enabled.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Security Artifacts

Settings Tab

The properties in this tab allow you to take appropriate actions while installing, updating, or uninstalling the Package.

Install

The settings in this tab are referred to when installing the Package.

- **File Watcher exists**: Select this property to take appropriate action when the File Watcher that is being installed is present in the target environment.
  - **Do not install File Watcher**: Select this option to use the File Watcher that is present in the target environment.
  - **Install to overwrite existing File Watcher**: Select this option to overwrite the File Watcher that is present in the target environment with the one that present in the package.
- **Physical Path**: This displays the path of the File Watcher in the Source environment and it also allows you to specify path of the File Watcher in the Target environment:
  - **Source**: Displays the path of the File Watcher in the Source environment.
  - **Target**: Specify the path of the File Watcher in the target environment. If it is left blank, then the source path will be considered.

Update

The settings in this tab are referred to when the same Package is installed for the consecutive time.

- **Delete File Watcher**: Select this option to delete the File Watcher that is present in the target environment.
  - **Yes**: Select this option to remove the File Watcher in the target environment.
  - **No**: Select this option to retain the File Watcher in the target environment.
- **File Watcher exists**: Select this option to take appropriate action when the File Watcher that is being

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

installed is present in the target environment.

- **Do not install File Watcher**: Select this option if you want to use the File Watcher that is present in the target environment.
- **Install to overwrite existing File Watcher**: Select this option if you want to overwrite the File Watcher that is present in the target environment with the one that present in the package.

- **File Watcher Is Modified**: Select this option to retain, roll back to original state, or delete the File Watcher in the target environment even after the Package is updated. This property is applicable only if **Delete File Watcher** is set to *Yes* and if the File Watcher is modified.
  - **Retain**: Select this option to retain the File Watcher along with the modifications in the target environment even after the Package is updated.
  - **Roll back to original State**: Select this option to roll back to the original state. All modifications that were done to the File Watcher after it was installed in the target environment will be reverted. However, the File Watcher will be retained in the target environment.
  - **Delete**: Select this option to delete the File Watcher from the target environment when updating the Package.

Uninstall

The settings in this tab are referred to when the Package is being uninstalled in the target environment.

- **Delete File Watcher**: Select this option to retain or delete the File Watcher in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to delete the File Watcher from the target environment when the Package is uninstalled.
  - **No**: Select this option to retain the File Watcher in the target environment even after the Package is uninstalled.
- **File Watcher Is Modified**: This option allows you to decide if you want to retain, roll back to original state or delete the File Watcher in the target environment even after the Package is uninstalled. This property is displayed only if **Delete File Watcher** is set to *Yes*. However, if **Delete File Watcher** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the File Watcher along with the modifications in the target environment even after the Package is uninstalled.
  - **Roll back to original State**: Select this option to roll back to the original state. All modifications that were done to the File Watcher after it was installed in the target environment will be reverted. However, the File Watcher will be retained in the target environment.
  - **Delete**: Select this option to delete the File Watcher from the target environment when uninstalling the Package.

Associated Artifacts Tab

File Watcher does not have any associated artifact.

Child Artifacts Tab

File Watcher does not have any child artifact.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Security Artifacts Tab

The Security Artifacts tab displays the security artifacts of a file watcher, if any.

You can enable the security settings of the file watcher that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

Resolve Physical Path of File Watcher

You can edit the physical path of the File Watcher in the target environment.

- If a File Watcher with the same name already exists in the target environment but with a different path, the File Watcher will not be installed and the following warning message appears:

  File Watcher already exists with a different physical path and cannot be updated.

  You can resolve the warning message by changing the path of the File Watcher in the package to the same path as in the target environment.
- If a File Watcher with a different name but same physical path as the one in the package is already present in the package, then the File Watcher will not be installed and the following warning message appears:

  A file watcher with the same physical path is already configured: <physical path>.

  You can resolve the warning message by changing the physical path in the package. Ensure that the new path is not configured for another File Watcher.

The **Physical Path** grid contains the following settings, which are required to resolve the physical path of the File Watcher.

- **Source**: Displays the physical path value of the File Watcher in the source environment.
- **Target**: Enter the physical path value of the File Watcher in the target environment. If you do not specify any value, then the value specified in the source is considered.

Folder

You can package and deploy **Folder** along with its associated and child artifacts from one environment or repository to another environment or repository.

Configuration Settings for Folder

The **Folder** has a unique set of configuration settings required to install, update, and uninstall the package. You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template > Edit**
- **Target Environment > Manage Package > Edit**

Settings Tab

You can define the configuration settings of the Folder artifact in the **Settings** tab. Set the options in this tab to take appropriate actions while installing, updating, or uninstalling the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Install

Use this tab to set the configurations for installing the package.

- **Parent folder is not present**: Select this property to take appropriate action when the parent folder is not present in the target environment.
  - **Install folder**: Select this option to install this folder in target environment even if the parent folder is not present.
  - **Do not install folder**: Select this option if you do not want to install the folder as the parent folder is not present in the target environment.

Update

Use the settings in this tab to update an installed package.

- **Parent folder is not present**: Select this property to take appropriate action when the parent folder is not present in the target environment.
  - **Update folder**: Select this option to update the folder in target environment even if the parent folder is not present.
  - **Do not update folder**: Select this option if you do not want to update the folder as the parent folder is not present in the target environment.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete folder** : Use this property to decide if you want to delete the folder or retain it when uninstalling the package.
  - **Yes**: Select this option to delete this folder after uninstalling the package.
  - **No**: Select this option to retain this folder after uninstalling the package.

    Note: The Folder in a package does not have the **Rollback** and **Overwrite** options like other artifacts.

Associated Artifacts of Folder

The **Associated Artifacts** tab displays all the associated artifacts of the selected folder.

Folder can include the following associated artifacts:

- Parent Folder
- Parent ListItem

Child Artifacts of Folder

The **Child Artifacts** tab displays all the child artifacts of the selected folder.

Folder can include the following child artifacts:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Child Folder
- Child ListItem

Include Child Item

For more information, see Include Child Item.

Security Artifacts of Folder

The **Security Artifacts** tab displays all the security artifacts of the selected folder.

You can enable the security settings of the folder that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

Child Folder

You can package and deploy a child folder along with its associated and child artifact from one environment or repository to another environment or repository.

Configuration Settings for Child Folder

The **Child Folder** has an unique set of configuration settings required to install, update, and uninstall the package. You can view and edit the configuration settings from one of the following windows:

- Source Environment > Package Template window
- Target Environment > Manage Package window

Settings tab

You can define the configuration settings of the **Child Folder** artifact in the **Settings** tab. Set the properties in this tab to take appropriate actions while installing, updating, or uninstalling the package.

Install and Update

Use this tab to set the configurations for installing and updating the package.

Parent item is not present: Use this property to take appropriate action when the parent item is not present in the target environment.

- **Install folder**: Select this option to install the child folder in target environment even if the parent item is not present.
- **Do not install folder**: Select this option to install the child folder as the parent item is not present in the target environment.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Delete folder : Use this property to decide if you want to delete the child folder or retain it when uninstalling the package.

- **Yes**: Select this option to delete the child folder after uninstalling the package.
- **No**: Select this option to retain the child folder after uninstalling the package.

Forms

You can package and deploy Forms with its associated and child artifacts from one environment or repository to another environment or repository.

You can also package custom js, CSS, images, and widgets as associated files to the Form and install them in the target environment. If you are packaging custom CSS, images, and widgets from the Default theme, you need to install it in the Default theme in the target environment to work as expected.

Forms

You can package and deploy Forms with its associated and child artifacts from one environment or repository to another environment or repository.

The Overwrite option available in the **Form Exists** property overwrites the existing Form Definition of the workflow in the target environment with the latest while installing or updating the package even if the latest Form version is in Draft state. When you overwrite the Form Definition for the first time, it gets backed up. But if you roll back, then the backed-up Form definition gets replaced with an existing Form definition.

The locked Forms will also get overwritten.

You can also package custom js, CSS, images, and widgets as associated files to the Form and install them in the target environment. If you are packaging custom CSS, images, and widgets from the Default theme, you need to install it in the Default theme in the target environment to work as expected.

The js file will not be copied to the target machine while installing the package. You need to manually copy it in **\BPMUITemplates\Default\NextGenForms\custom\js**

To add Forms to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, and then click **Forms**.

   The **Forms** window appears.
3. Select the form that you want to package.
4. Right-click on the form, and then click **Add to Package Template**.

   The **Add to Package Template** dialog box appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of Forms in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the Forms package in the Settings form on the right side of the Add to Package Template window.

The status of the forms or folders created through the Forms page will be in *Draft* status. To add a form or a

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

folder to a package template, you must first publish the form or the folder. Also ensure the form that you want to package is not disabled.

You can package and install a form to the target environment even if it is in a locked state, but in the target environment the locked state of the installed form is not retained.

Runtime Initialization Settings

You can configure runtime initialization settings for forms. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration Settings for Forms

Forms has a unique set of configuration settings required to install, update, and uninstall the package. You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

Settings

You can define the configuration settings of Forms artifact in the **Settings** tab. Set the properties in this tab to take appropriate actions while installing, updating, or uninstalling the package.

Note: The **Control Id** field displays the complete position of the control in the form. For example: If the control with Id "H3" is used within a panel form with Id "P1", then the **Control Id** will be displayed as "P1.H3".

In the package template, an orange icon will appear for the Forms prompting you to rebuild it when the form is published, unpublished, or edited after it is added to the package template. Similarly, in the target environment if the form installed by the package is published, unpublished or edited, then while uninstalling the package the forms settings will be retained.

When a form in the package template is modified, ensure to perform a rebuild on the particular form. You can right-click the form from the left hand side configuration tree view in the package template, and click **Complete Rebuild** or **Control Rebuild** to rebuild the form.

- **Complete Rebuild**: Click **Complete Rebuild** to rebuild the particular form artifact in the package template. During complete rebuild, the install, update, or uninstall settings do not change. However, the settings done to resolve any control properties in the target section of the **Settings** form get cleared.
- **Control Rebuild**: Click **Control Rebuild** to rebuild only a modified form artifact. All settings done to resolve any control property in the target section of the **Settings** form does not get changed. Only the new values of the modified control get updated in the package. However, if you want to rename a control in the form, use the **Complete Rebuild** option.

Install

Use this tab to set the configurations for installing the package.

- **Inherit Artifact Type Settings:** Use this property to define whether the settings inherited from the artifact type or the settings configured at the artifact-level must be considered.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Yes**: Select this option if the form must inherit the settings configured for the artifact type. If this option is selected, then the artifact type settings take precedence.
- **No**: Select this option if the form must consider the settings configured at the artifact-level.The following options are available:

Property	Description
Install Form as Draft	Use this property to install the form. • **Yes**: Select this option to install the form in the *Draft* status in the target environment. • **No**: Select this option to install the form in the *Published* status in the target environment.
Form Exists	Use this property to take appropriate action when the forms in the package is already present in the target environment. Existing forms will not be overwritten, hence you can choose to install it with a different version. • **Install as next major version**: Select this option to install the forms as a next major version. • **Install as next minor version**: Select this option to install the forms as a next minor version. • **Install as next version**: Select this option to install the form as a next version. • **Overwrite latest (published) version**: Select this option to overwrite the latest published version of form in the target environment. An existing Form Definition gets replaced with the Form Definition contained in this package. • **Do not install**: Select this option if you do not want to install the form, as it is already present in the target environment.

- **Associated Artifacts Not Present**: Use this property to take appropriate action when the associated artifacts of the forms are not present in the target environment.
  - **Install as draft**: Select this option to install the forms in the *Draft* status, even if the associated artifacts are not present in the target environment.
  - **Do not install**: Select this option if you do not want to install the forms as its associated artifacts are not present.
- **Workflow Version**: See Forms Association for information.
- **Custom Widgets**: Use this property to use the custom widgets as the associated artifact in the form in the target environment.

  In **<Install path>\BPMUITemplates\Default\NextGenForms\custom\Widgets**, the following folders get created in the target environment when you install the package having custom widgets in a form:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Progress
- BaseGrid
- Grid1

Note: Consider, the latest published form exists in the target environment, and the **Overwrite latest (published) version** option is selected. During package installation, the backup of the existing form is taken and stored in the database. And, the form in the target environment is overwritten with the packaged source definition.

During Overwrite, the form name and the description does not get updated in the artifact **Package Name (Version)** column.

Update

Use the settings in this tab to update an installed package.

- **Delete Form**: Use this property to delete the form that is present in the target environment.
  - **Yes**: Select this option to remove the form from the target environment.

    Form is modified: Use this option to retain or delete the form in the target environment even after the package is updated. This property is applicable only if **Delete Forms** is set to *Yes* and the form is modified.
    - **Retain**: Select this option to retain the form along with the modifications, in the target environment.
    - **Rollback to original state**: Select this option to roll back the artifact to the backup taken. All modifications that were done to the Form after it was installed in the target environment will be reverted. However, the Form will remain in the target environment
    - **Delete**: Select this option to delete the form from the target environment.
  - **No**: Select this option to retain the form in the target environment. On selection of this option, the **Install Forms as draft**, **Forms exist**, and **Associated artifacts not present** options appear.

    Only when the **Delete Form** option is set to **No**, the **Inherit Artifact Type Settings** option is enabled.
- **Inherit Artifact Type Settings:** Use this property to define whether the settings inherited from the artifact type or the settings configured at the artifact-level must be considered.
  - **Yes**: Select this option if the form must inherit the settings configured for the artifact type. If this option is selected, then the artifact type settings take precedence.
  - **No**: Select this option if the form must consider the settings configured at the artifact-level. The following options are available:

Property	Description
Install Form as Draft	Use this property to install the form. • **Yes**: Select this option to install the form in the *Draft* status in the target environment. • **No**: Select this option to install the form in the *Published* status in the target environment.
Form Exists	Use this property to take appropriate action when the forms in the package is already present in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	target environment. Existing forms will not be overwritten, hence you can choose to install it with a different version. • **Install as next major version**: Select this option to install the forms as a next major version. • **Install as next minor version**: Select this option to install the forms as a next minor version. You can install the form as a next minor version up to 12 times only. • **Install as next version**: Select this option to install the form as a next version. • **Overwrite latest (published) version**: Select this option to overwrite the latest published version of form in the target environment. An existing Form Definition gets replaced with the Form Definition contained in this package. • **Do not install**: Select this option if you do not want to install the form, as it is already present in the target environment.

- **Associated Artifacts Not Present**: Use this property to take appropriate action when the associated artifacts of the forms are not present in the target environment.
  - **Update as draft**: Select this option to update the forms in draft status, even if the associated artifacts are not present in the target environment.
  - **Do not update**: Select this option if you do not want to update the forms as its associated artifacts are not present.
- **Custom Widgets**: Use this property to use the custom widgets as the associated artifact in the form in the target environment.

  In **<Install path>\BPMUITemplates\Default\NextGenForms\custom\Widgets**, the following folders get updated in the target environment when you update the package having form with custom widgets:

  - Progress
  - BaseGrid
  - Grid1

  Note: Consider, the latest published form exists in the target environment, and the **Overwrite latest (published) version** option is selected. During package update, the backup of the existing form is not be taken if available; and taken if not available. And, the form in the target environment is overwritten with the packaged source definition.

  During Overwrite, the form name and the description does not get updated in the artifact **Package Name (Version)** column.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete Forms**: Use this property to retain or delete the forms in the target environment even after the package is uninstalled. If the forms is a dynamic form and it is associated to a list, then you cannot delete the dynamic form until the list is deleted.
  - **Yes**: Select this option to remove the form from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the form in the target environment even after the package is uninstalled.
- **Form is modified**: Use this property to retain or delete the forms in the target environment even after the package is uninstalled. This property is displayed only if **Delete Forms** is set to *Yes*. However, if **Delete Forms** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the form along with the modifications even after the package is uninstalled.
  - **Rollback to original state**: Select this option to roll back the artifact to the backup taken. All modifications that were done to the Form after it was installed in the target environment will be reverted. However, the Form will remain in the target environment
  - **Delete**: Select this option to delete the form from the target environment.

Note:
- During package uninstall, with the **Rollback to original state** option selected, the overwritten from definition is replaced with backup data, and the backup from the database gets deleted.
- During rollback, the form name and the description does not get updated in the artifact **Package Name (Version)** column.
- When you uninstall the package, the folders **Progress**, **BaseGrid**, and **Grid1** does not get deleted from the target machine.

Associated Artifacts of Forms

The **Associated Artifacts** tab displays all the associated artifacts of the selected forms.

The following artifacts can appear as **Associated Artifacts** if you are using any one of the following artifacts while designing the forms:

- Parent Folder
- Lookup
- SOA Assembly
- Data Grid Configuration
- Data Grid View
- Report Viewer with the Embed and Pop-up option
- Report Viewer with the Export option
- Invoke Workflow
- Associated Workflow
- Container Control

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Forms Tree Control
- MQTT Connection

When you set the **Include Associated Artifacts** option to **Yes**, while adding the form to a package template, then the associated artifacts are added to the package template, and it also allows you to view the associated artifacts of the form in the tree view of the package. If you set the **Include Associated Artifacts** option to **No**, then the associated artifacts are not added to the package template, and are also not visible in the tree view of the package.

Note: If a SOA Assembly is registered through a Web Service and a physical path (dll), by default the Web Service is a static method and the physical path can be a static or non- static method. While packaging forms that does not use SOA Assembly, by default the static objects of the registered SOA Assembly is considered as an associated artifact.

Child Artifacts of Forms

The **Child Artifacts** tab displays all the child forms or folders that are published.

Forms can include the following child artifacts:

- Folder
- Child Form

Include Child Item

For more information, see Include Child Item.

The child item can also be another folder. When you add the child item, all the associated artifacts, child artifacts, and security artifacts of the child item also gets added to the package template. You can add up to top three levels of child item. To know more about adding a folder as child item, see Child Folder.

Security Artifacts

The **Security Artifacts** tab displays all the security artifacts of Forms, if any. You can enable the security settings of the forms that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

Forms Settings Grid

This section is applicable only when you have Forms with **Hyperlink** or **Image** controls.

While installing, the package does not validate the images of the forms control in the target repository or environment. You can take appropriate action using the **Forms Settings Grid** property when forms include controls such as hyperlink or images. You can use the **Forms Settings Grid** to make any correction to the source property value by providing the correct information in the given target property value during the design or install time. If you do not specify the target property in the **Forms Settings Grid** during design time, you can specify the settings during the install time in the target properties or else it will take the values specified in the source property value during installation.

- For more information about control settings of a hyperlink, see Resolving Hyperlink.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- For more information about control settings of an image, see Resolving Image.

Resolving Version Details for Invoke Workflow

During package installation, you can resolve the version details of Invoke Workflow artifacts.

For more information about resolving version details of Invoke Workflow, see Resolving Invoke Workflow.

Rebuild Scenario

When you rebuild a form that was edited after adding it to the package template, the package template retains the old data in the settings form, but considers only the latest record. This does not affect the functionality of the forms that is rebuilt. For example, consider a form that contains a Hyperlink control which points to an employee URL. After adding the forms to the package template, consider that the Hyperlink control is modified to customer URL. When you rebuild forms, the Settings form will contain the employee URL but upon installation, will consider the customer URL.

Pop-Up control will always point to the latest version of the form.

Resolve Action Bar Settings

Action Bar Dependent Files

This section is applicable only when you have Forms with the **Action Bar** enabled.

When you package and install a Form with Action Bar, the Custom Style Sheet and images used in the Action Bar within the Form are identified by the package and added to the package template.

You can use the **Action Bar Dependent Files** grid to define the target path where you want to place the image files for the Action Bar. Ensure that the path mentioned in the **Target Value** field is relative.

Note: Ensure that you include the image files and the corresponding hover image files along with it.

Resolve Action Bar settings

During package installation, if the form contains an **Action Bar** control, you can define the target property values in the **Action Bar Settings** section of the **Settings** window.

The following table provides information about the settings available for Forms with an **Action Bar** item:

Field Name	Description
Action Bar Item	Name that identifies the Action Bar item.
Source Value	Path of the source property. The **Source Value** displays the source path of the image files and the corresponding hover image files.
Target Value	Path of the target property. The **Target Value** displays the target path of the image files and the corresponding hover image files.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Resolve Print Forms Details

While installing a package or while adding a form to a package template, if the form contains a Button control of Button Type "Print", you can edit the version details of the associated Print Forms. You can define the target property values in the **Associated Print Forms Details** section of the **Settings** window.

Field Name	Description
Forms Name	Name that identifies the form.
Source Version	The version of the form selected in the **Button** control of **Button Type "Print"**.
Target Version	The version of the form, which will be used in the **Button** control of **Button Type "Print",** during installation. Ensure that this version of the form is available in the target environment.

In the **Associated Print Forms Details** section, you can enter the version of the associated form to which the current control should be mapped.

- If you do not edit the **Target Version** field and retain the default value as "Latest", then the latest version of the associated form is mapped to the Print Form in the target environment.
- If the **Target version** field is left blank, then the source version of the associated form present in the target environment is mapped to the current form. If the source version is not present in the target environment, then the latest version of the associated form is mapped to the current version in the target environment.
- If the **Target version** of the form is not present in the target environment, then the latest version of the associated form is mapped to the Print Form in the target environment.
- You can use the **Associated Print Forms Details** property to specify the version of the associated form to which the current form should be mapped in the target environment.
- The **Associated Print Forms Details** property is only to specify the target version of the associated form to which the current form should be mapped. The configuration of the associated form is done separately from the package template as it is added as an individual artifact.

Resolve Navigate Form Details

While adding a form to a package template or while installing a package, if the form contains a Button control of Button Type "Navigate", you can edit the version details of the form to which you have navigate to. You can define the target property values in the **Associated Forms with Navigate Button** section of the **Settings** window.

Field Name	Description
Button Id	The ID of the button in the form.
Button Name	The name of the button in the form.
Forms Name	Name of the form that is selected in the **Navigate To Form** property of the **Button** control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description
Source Version	The version of the form selected in the **Button** control of **Button Type "Navigate"**.
Target Version	The version of the form, which will be used in the **Button** control of **Button Type "Navigate",** during installation. Ensure that this version of the form is available in the target environment.

- In the **Associated Forms with Navigate Button** section, you can enter the version of the associated form to which the current control should be mapped.
- If you do not edit the **Target Version** field and retain the default value as "Latest", then the latest version of the associated form is mapped to the form to be navigated to in the target environment.
- If the **Target version** field is left blank, then the source version of the associated form present in the target environment is mapped to the current form. If the source version is not present in the target environment, then the latest version of the associated form is mapped to the current version in the target environment.
- If the **Target version** of the form is not present in the target environment, then the latest version of the associated form is mapped to the form to be navigated to in the target environment.
- You can use the **Associated Forms with Navigate Button** property to specify the version of the associated form to which the current form should be mapped in the target environment.
- The **Associated Forms with Navigate Button** property is only to specify the target version of the associated form to which the current form should be mapped. The configuration of the associated form is done separately from the package template as it is added as an individual artifact.

Configure Artifact Type Settings for Forms

You can set the status in which all the Form artifacts in a package must be installed or updated in the target environment.

Configure artifact type settings for Forms in the following tabs:

- **Install**

  Use this tab to set the configurations for installing the package.
- **Update**

  Use the settings in this tab to update an installed package.

Artifact Type Settings

1. In the **Enterprise Console**, click menu icon, click **Package**, and then click **Package Template**.

   The **Package Template** page appears.
2. To edit an existing package, in the **Package Template** page, select the package template, and then click **Edit**. The **Edit Package Template** dialog box appears.
3. In the **Configuration** tree on the left pane, click **Forms** under **Artifacts**. The **Settings** form with the **Artifact**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Type Settings section appears.

4. For the **Install Forms as Draft** property, select one of the following:
   - **Yes**: Select this option to install all Form artifacts in the package in *Draft* status in the target environment.
   - **No**: Select this option to install all Form artifacts in the package in *Published* status in the target environment.

     The status of the Forms in a package is set based on the value defined.
5. Use the **Forms Exist** property to take appropriate action when the forms in the package is already present in the target environment. Existing forms will not be overwritten, hence you can choose to install it with a different version. Select one of the following:
   - **Install as next major version**: Select this option to install all Form artifacts in the package as next major version.
   - **Install as next minor version**: Select this option to install all Form artifacts in the package as next minor version.
   - **Install as next version**: Select this option to install all Form artifacts in the package as a next version.
   - **Overwrite latest (published) version**: Select this option to overwrite the latest published version of Form in the target environment. An existing Form Definition gets replaced with the Form Definition contained in this package.
   - **Do not install**: Select this option if you do not want to install the Form artifacts, as they are already present in the target environment.

     Note: You can also configure these settings while installing the package using the **Manage Package** page.

Resolve Custom Dependent Files for Forms

When you package Forms, the custom style sheet files are automatically identified and appear as custom dependent files. The Source Files field displays the physical path of this files in the source environment. When the package is installed, this file is copied to the installed path of the target environment.

This is a read-only file which cannot be edited or deleted.

Note: If the source environment is in a particular theme and the css is packaged in that theme, the css will be copied in the target environment to the path of the source theme. You have to manually copy the css from the source theme folder to the respective theme folder of the target environment .

For more information on configuring custom style sheet, see section *Customizing Forms Using Custom Styles* of *Forms and Lists User Guide.*

Resolve Invoke Workflow

During package installation or while adding forms to a package template, if the forms contain **Invoke Workflow** control, you can edit the version details of the workflow that is selected in this control. You can correct the source property values and define the target property values in the **Invoke Workflow details** section of the **Settings** window.

The following table provides information about the settings available for a Form with a Invoke Workflow control:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description
Workflow Name	Name that identifies the workflow.
Source Version	The version of the workflow selected in the **Invoke Workflow** control.
Target Version	The version of the workflow, which will be used in the **Invoke Workflow** control in the form, during installation. Ensure that this version of the workflow is available in the target environment.

- If you do not edit the **Target Version** field and retain the default value as "Latest", then the latest version of the associated workflow is mapped to the control in the form in the target environment.
- If the **Target version** field is left blank, then the source version of the associated workflow present in the target environment is mapped to the current control in the form. If the source version is not present in the target environment, then the latest version of the associated workflow is mapped to the current version in the target environment.
- If the **Target version** of the workflow is not present in the target environment, then the latest version of the associated workflow is mapped to the control in the form in the target environment.
- In **Invoke Workflow details** section, the **Workflow Name** property is not available in the configuration settings if the workflow name is configured using expression editor.
- The **Invoke Workflow details** property is only to specify the target version of the associated workflow to which the current control should be mapped. The configuration of the associated workflow is done separately from the package template as it is added as an individual artifact.

Resolve Image

During package installation, if the form contains an **Image** control, you can correct the source property values and define the target property values in the **Forms Settings Grid** section of the **Settings** window.

The following table provides information about the settings available for Forms with an **Image** control:

Field Name	Description
Control Id	Reference identification code for the control. This value is automatically generated and cannot be changed.
Control Name	Name that identifies the control. This value is also used for the label of the control. If a form contains a hyperlink, the **Control Name** field displays the tag name of the hyperlink.
Property Name	Name of the property. If a form contains an image, the **Property Name** field displays the image location.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description
Source Value	Path of the source property. If a form contains an image, the **Source Value** displays the source path of the image.
Target Value	Path of the target property. If a form contains an image, the **Target Value** displays the target path of the image.

Resolve Hyperlink

During package installation, if the forms contain a **Hyperlink** control, you can correct the source property values and define the target property values in the **Forms Settings Grid** section of the **Settings** window.

The following table provides information about the settings available for a Form with a Hyperlink control:

Field Name	Description
Control Id	Reference identification code for the control. This value is automatically generated and cannot be changed.
Control Name	Name that identifies the control. This value is also used for the label of the control. If a form contains a hyperlink, the **Control Name** field displays the tag name of the hyperlink.
Property Name	Name of the property. If a form contains a hyperlink, the **Property Name** field displays the property name as *NavigateURL*.
Source Value	Path of the source property. If a form contains a hyperlink, the **Source Value** field displays the source URL that you can navigate to.
Target Value	Path of the target property. If a form contains a hyperlink, the **Target Value** field displays the proposed URL that you want to navigate to.

Forms Association

You can package and deploy a form with its associated artifacts such as workflow (for Fill Form) from one environment or repository to another environment or repository using the **Package** feature in AVEVA Work Tasks.

When a form (associated to a workflow) is packaged, the following options are displayed in the **Associated**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Workflow details section under the **Settings** window. The **Associated Workflow details** section is displayed only in the **Install** and the **Update** tabs. For other options, see Forms.

Workflow Version

Field	Description
Bound to Workflow	Displays the name of the workflow to which the form is associated.
Source Version	Displays the version of workflow in the source environment to which the form is associated.
Target Version	Enter the version of the workflow: • Specify the version of the workflow to which you want to associate the form in target environment. • If you do not specify the version of the workflow, then the latest version of the workflow is considered. **Note:** If the version of the workflow that is specified while installing the package is not present in the target environment, then the next minor version of the latest version of the workflow in the target environment is considered.

Example

Consider the following scenario:

Source Environment

- Form1 is associated to Workflow1.
- Workflow1 has versions 1, 2, 3 and 4 with version 1 published and versions 2, 3, and 4 in draft status.

Package Template

- Form1 associated to version 4 of Workflow1.
- Target version is not specified.

Target Environment

- Workflow1.
- Workflow1 has versions 1, 2, 3 and 4 with version 1 published and versions 2, 3, and 4 in draft status.

When the package is installed in the target environment, Form1 is created and version 1.1 of Workflow1 is created in Draft status as only version 1 of Workflow1 is published. Form1 is associated to version 1 of Workflow1 and not version 1.1 of Workflow1 as the version is in Draft status. You must publish the desired version of the workflow if you want the form to automatically associate to the latest version (that is the last

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

published version) of the of workflow.

When the package is uninstalled from the target environment, the version 1.1 of Workflow1 is removed. However, Form1 is not removed as Form1 is associated to the next latest version of Workflow1.

Form Controls

You can package and deploy the Form Controls along with its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add Form Controls to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Form Controls**.
   The **Form Controls** window appears.
3. Select the Form Controls that you want to package.
4. Right-click the form controls and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Runtime Initialization Settings

You can configure runtime initialization settings for form controls registered through GAC. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration settings for Form Controls

The **Form Controls** has a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the **Settings** tab to define the configuration settings. The **Associated Artifacts** tab displays the associated artifacts of the **Form Controls** definition. The **Child Artifacts** tab displays child artifacts of the **Form Controls** definition.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Child Artifacts
- Security Artifacts

Settings Tab

You can use the properties in this tab to take appropriate actions to install, update, or uninstall the package.

Install

The settings in this tab are referred to when installing the Package.

- **Form Control exists**: Select this property to take appropriate action when the Form Controls which is installed is present in the target environment
  - **Do not install Form Control**: Select this option to use the Form Controls which is present in the target environment
  - **Install to overwrite existing Form Control**: Select this option to overwrite the Forms Control which is present in the target environment with the one that is currently being packaged
- **Associated artifacts not present**: Select this property to take appropriate action when the associated artifacts of the Form Control is not present in the target environment.
  - **Install Form Control**: Select this option to install the Forms Control, even if the associated artifacts are not present in the target environment.
  - **Do not install Form Control**: Select this option if you do not want to install the Forms Control as its associated artifacts are not present.

Update

The settings in this tab are referred to when the same Package is installed for the consecutive time.

- **Delete Form Control**: Select this option to delete the Forms Control which is present in the target environment
  - **Yes**: Select this option to remove the **Form Controls** in the target environment. On selection of this option, the **Form Control Is Modified** option appears.

    Form Control Is Modified: Select this option to retain, roll back to original state, or delete the Forms Control in the target environment even after the Package is updated. This property is applicable only if **Delete Forms Control** is set to *Yes* and if the Form Control is modified.
    - **Retain**: Select this option to retain the Forms Control along with the modifications, in the target environment even after the Package is updated.
    - **Rollback to original state**: Select this option to roll back to the original state. All modifications that were done to the Forms Control after it was installed in the target environment will be reverted. However, the Forms Control will be retained in the target environment.
    - **Delete**: Select this option to delete the Forms Control from the target environment when updating the Package.
  - **No**: Select this option to retain the Form Control in the target environment. On selection of this option, the **Form Control Exists** option appears.

    Form Control exists: Select this property to take appropriate action when the Form Control definition present in the package is already present in the target environment.
    - **Do not install Form Control**: Select this option to use the Forms Control which is present in the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Install to overwrite existing Form Control**: Select this option to overwrite the Forms Control, which is present in the target environment with the one that is currently being packaged.

  Associated artifacts not present: Select this property to take appropriate action when the associated artifacts of the Form Control are not present in the target environment.
- **Update Form Control**: Select this option to update the Forms Control.
- **Do not update Form Control**: Select this option if you do not want to update the Forms Control as the associated artifacts are not present.

Uninstall

The settings in this tab are referred to when the Package is being uninstalled in the target environment.

- **Delete Form Control**: Select this property to retain or delete the Forms Control in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to remove the Forms Control from the target environment when the Package is uninstalled.
  - **No**: Select this option to retain the Forms Control in the target environment even after the Package is uninstalled.
- **Forms Control Is Modified**: Select this option to decide if you want to retain, roll back to original state or delete the Forms Control in the target environment even after the Package is uninstalled. This property is displayed only if **Delete Form Control** is set to *Yes*. However, if **Delete Form Control** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the Forms Control along with the modifications, in the target environment even after the Package is uninstalled.
  - **Rollback to original state**: Select this option to roll back to the original state. All modifications that were done to the Forms Control after it was installed in the target environment will be reverted. However, the Forms Control will be retained in the target environment.
  - **Delete**: Select this option to delete the Forms Control from the target environment when uninstalling the Package.

Associated Artifacts for Form Controls

This tab displays all the associated artifacts of the selected Form Controls artifact.

The following artifact can appear as an Associated Artifact, if you are using any one of the following artifacts while designing the Forms Control:

- SOA Folder

Child Artifacts for Form Controls

Form Controls does not have any child artifact.

Security Artifacts for Form Controls

The Security Artifacts tab displays the security artifacts of form controls, if any.

You can enable the security settings of the form controls that you want to include in the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For more information about including security artifacts, see Include Security Artifacts.

Note: If a **Form Control** is consumed inside a form, you cannot modify or delete the **Form Control**. To modify or delete the **Form Control**, you have to first modify or delete all its associated artifacts.

Resolve Custom Dependent Files for Form Controls

The configuration settings for Form Controls differ based on how the Form Controls are registered. Form Controls can be registered through Global Assembly Cache (GAC) or physical path.

Register Form Controls from Global Assembly Cache

The **Source Version** field displays the version of the form control in the source environment. In the **Target** field, you can specify the version of the Form Controls that you want to use in target environment. The source version is considered while installing the package if you do not specify the target version.

You can also package the identified custom dependent files which can be installed while installing the package in the target environment. The **Source Files** field displays the physical assembly path of the Form Controls in the source environment.

Under **Install in Global Assembly Cache**, click **Yes** to install the Form Controls and custom dependent files in the GAC of the target environment. Click **Add Record** to add multiple rows to install additional files in the target environment.

Register Form Controls from a Physical Path

The **Custom assembly name source** field displays the physical path of the Form Controls in the source environment. You can also package the identified custom dependent files which can be installed while installing the package in the target environment. Click **Add Record** to add multiple rows to install additional files in the target environment. In the **Custom assembly name source** field, enter the source file path of the custom dependent files.

If the source folder is relative to the AVEVA Work Tasks installed path, and the folder does not exist in the target environment, then a new folder is created and the files are copied to it. For example, if the path of the file, rating.dll, is *C:\Program files\AVEVA\Work Tasks\\FormsCustom Control\rating.dll*, then a new folder named **FormsCustom Control** is created in the target machine and rating.dll is copied to it.

If the source folder is not relative to the AVEVA Work Tasks installed path, then the files are copied to the *C:\Program files\AVEVA\Work Tasks\\bin* folder. For example, if the path of the file, rating.dll, is **C:\FormsCustom Control\rating.dll**, then **rating.dll** is copied to *C:\Program files\AVEVA\Work Tasks\\bin* folder in the target machine.

The **Install in Global Assembly Cache** field is optional and is used to install the Form controls and custom dependent files in the GAC of the target environment.

Note: Click the icon in the row corresponding to a custom dependent file, which should not be added to the package template.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Global Work Item Layout

You can package and deploy both the Global Work Item Layout and Custom Work Item Layout from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add Global Work Item Layout to Package Template

1. Navigate to **Global Work Item layout**.
2. Select a Layout. You can either select a Global Layout or a Custom Layout which is saved.
3. Click **Add to Package Template**. The **Add to Package Template** page appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

In the Package Template the custom package template appears along with the name of the custom layout.

Associated Artifacts for Work Item Layout

Work Item Layout does not have any associated artifact.

Child Artifacts for Work Item Layout

Work Item Layout does not have any child artifact.

Configuration Settings for Work Item Layout

The Work Item Layout has a unique set of configuration settings required for installing, updating, and uninstalling the Package. You can edit and view the configuration settings from the Source Environment > Package Template screen or Target Environment > Manage Package screen. The Settings tab of the Work Item Layout artifact allows you to define the configuration settings. The Associated and Child Artifacts tab are not enabled as it cannot have Associated and Child Artifacts.

- Settings
- Associated Artifacts
- Child Artifacts

Settings Tab: You can use the properties in this tab to take appropriate actions while installing, updating, or uninstalling the Package.

Install: The settings in this tab is referred to when installing the Package.

- **Work Item Layout Exists**: Select this property to take appropriate action when the Work Item Layout which is being installed is present in the target environment
  - **Do not install Work Item Layout**: Select this option to use the Work Item Layout which is present in the target environment.
  - **Install to overwrite existing Work Item Layout**: Select this option to overwrite the Work Item Layout which is present in the target environment with the one that is currently being packaged.
- Work Item fields are not packaged when packaging Work item layout, hence if the work item fields are not available in the target environment, then on installing the package the following message appears 'Work

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Item field(s) is missing: <Field Name>'. This can be resolved by creating the required work item fields in the target environment.

- If the data type of the work item field in source and target environment do not match, then on installing the package the work item layout definition will be updated with the data type and values of the work item field as in the target environment. In this scenario a message 'Work item field(s) is of different type' appears.

Update: You can use the settings in this tab when the same package is installed for the consecutive time.

- **Delete Work Item Layout**: Select this option to delete the Work Item Layout which is present in the target environment
  - **Yes**: Select this option to remove the Work Item Layout in the target environment. On selection of this option, the Work Item Layout Modified property appears.
  - **Work Item Layout Is Modified**
    - **Retain**: Select this option to retain the Work Item Layout along with the modifications, in the target environment even after the Package is uninstalled
    - **Rollback**: Select this option to roll back to the original state. All modifications that were done to the Work Item Layout after it was installed in the target environment will be reverted. However, the Work Item Layout will be retained in the target environment.
    - **Delete**: Select this option to delete the Work Item Layout from the target environment when uninstalling the Package.
  - **No**: Select this option to retain the Work Item Layout in the target environment. On selection of this option, the Work Item Layout Exists property appears that you can configure.
  - **Work Item Layout Exists**
    - **Use Existing Work Item Layout**: Select this option to use the Work Item Layout which is present in the target environment
    - **Do not update Work Item Layout**: Select this option to overwrite the Work Item Layout which is present in the target environment with the one that is currently being packaged.

Uninstall: You can use the settings in this tab when the package is being uninstalled in the target environment.

- **Delete Work Item Layout**: Select this property to retain or delete the Work Item Layout in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to delete the Work Item Layout from the target environment when the Package is uninstalled.
  - **No**: Select this option to retain the Work Item Layout in the target environment even after the Package is uninstalled.
- **Work Item Layout Is Modified**: Select this option to decide if you want to retain, roll back to original state or delete the Work Item Layout in the target environment even after the Package is uninstalled. This property is displayed only if **Delete Work Item Layout** is set to *Yes*. However, if **Delete Work Item Layout** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the Work Item Layout along with the modifications in the target environment even after the Package is uninstalled.
  - **Roll back to original State**: Select this option to roll back to the original state. All modifications that were done to the Work Item Layout after it was installed in the target environment will be reverted. However, the Work Item Layout will be retained in the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Delete**: Select this option to delete the Work Item Layout from the target environment when uninstalling the Package.

Associated Artifacts

Work Item Layout does not have any associated artifacts.

Child Artifacts

Work Item Layout does not have any child artifacts.

Grid Configuration

You can package and deploy the Grid Configuration along with its associated artifacts and child artifacts from one environment or repository to another environment or repository.

To add Grid Configuration to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, click **Settings**, click **Data Grid**, and then click **Grid Configuration**.

   The **Grid Configuration** window appears.
3. Select the grid configuration that you want to package.
4. Right-click the grid configuration and then click **Add to Package Template**.

   The **Add to Package Template** window appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of the grid configuration in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the grid configuration package in the Settings form on the right side of the Add to Package Template window.
5. Configure the required settings.
6. Click **Save Template** to save the configured settings.

Configuration Settings for Grid Configuration

The **Grid Configuration** has an unique set of configuration settings required to install, update, and uninstall the package. You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Tab

You can define the configuration settings of the grid configuration artifact in the Settings tab. Set the properties in this tab to take appropriate actions while installing, updating, or uninstalling the package.

Install

Use this tab to set the configurations for installing the package.

- **Grid Configuration Exists**: Select this property to take appropriate action when the Grid Configuration to be installed is present in the target environment.
  - **Do not install Grid Configuration**: Select this option to use the Grid Configuration which is present in the target environment.
  - **Install to overwrite existing Grid Configuration**: Select this option to overwrite the Grid Configuration present in the target environment with the one that you packaged.
- **Associated Artifacts Not Present**: Select this property to take appropriate action when the associated artifacts of the Grid Configuration are not present in the target environment.
  - **Install Grid Configuration**: Select this option to install the Grid Configuration, even if the associated artifacts are not present in the target environment.
  - **Do not install Grid Configuration**:Select this option if you do not want to install the Grid Configuration as its associated artifacts are not present in the target environment.

Note: In case of a List Lookup, ensure that the Columns in the List Lookup matches the target environment before you install the Data Grid Configuration package to the target environment. If you try to install the Data Grid Configuration package in the target environment without matching the Columns, the package installs the Data Grid Configuration with the following warning message:

Either Tables or Columns are missing. Verify your Database or List

Besides this, you might also get the same warning message if you use a repository which is a case-sensitive database and the queries or columns in the target environment do not match as it is written in the source environment.

You can resolve this error by correcting the Column details in the target environment. After correction, you must reconfigure the Data Grid Configuration item including the Advance Settings and Relations again. To configure the Data Grid Configuration item, you can access the item from the Enterprise Console menu or run the package again.

Update

Use the settings in this tab to update an installed package.

- **Delete Grid Configuration**: Select this property to delete or retain an existing Grid Configuration which is present in the target environment.
  - **Yes**: Select this option to remove the Grid Configuration from the target environment.

    Grid Configuration is modified: Select this option to retain, roll back to original state, or delete the Grid

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration in the target environment even after the package is updated. This property is applicable only if **Delete Grid Configuration** is set to *Yes* and the grid configuration is modified.

- **Retain**: Select this option to retain the Grid Configuration along with the modifications in the target environment even after the package is updated.
- **Rollback to original state**: Select this option to roll back to the original state. All modifications that were done to the Grid Configuration after it was installed in the target environment will be reverted. However, the Grid Configuration will be retained in the target environment.
- **Delete**: Select this option to delete the Grid Configuration from the target environment when updating the package.

- **No**: Select this option to retain the Grid Configuration in the target environment.

  Grid Configuration exists: Select this option to take appropriate action when the Grid Configuration that is being installed is present in the target environment.

  - **Do not install Grid Configuration**: Select this option to use the existing Grid Configuration which is present in the target environment.
  - **Install to overwrite existing Grid Configuration**: Select this option to overwrite the existing Grid Configuration which is present in the target environment with the one that is currently being packaged.

    Associated Artifacts not present: Select this option to take appropriate action if associated artifact is not present in the target environment.
  - **Update Grid Configuration**: Select this option to update the Grid Configuration.
  - **Do not update Grid Configuration**: Select this option if you do not want to update the Grid Configuration as associated artifacts are not present.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete Grid Configuration**: Select this property to retain or delete the Grid Configuration in the target environment even after the package is uninstalled.
  - **Yes**: Select this option to remove the Grid Configuration from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the Grid Configuration in the target environment even after the package is uninstalled.
- **Grid Configuration is modified**: Select this property to retain, rollback, or delete the Grid Configuration from the target environment even after the package is uninstalled. This property is displayed only if **Delete Grid Configuration** is set to *Yes*. However, if **Delete Grid Configuration** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the Grid Configuration with the modifications in the target environment even after the package is uninstalled.
  - **Rollback to original state**: Select this option to rollback to the original state. All modifications that were done to the Grid Configuration after it was installed in the target environment will be reverted. However, the Grid Configuration will be retained in the target environment.
  - **Delete**: Select this option to delete the Grid Configuration from the target environment when uninstalling the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Associated Artifacts of Grid Configuration

The **Associated Artifacts** tab displays all the associated artifacts of the selected Grid Configuration artifact.

Grid configuration can include the following associated artifacts:

- Parent Folder
- Lookup

Child Artifacts of Grid Configuration

The **Child Artifacts** tab displays all the child artifacts of the selected Grid Configuration artifact.

Grid configuration can include the following child artifacts:

- Folder
- Child Grid Configuration

Include Child Item

For more information, see Include Child Item.

Security Artifacts of Grid Configuration

The **Security Artifacts** tab displays all the security artifacts of the selected Grid Configuration artifact. You can enable the security settings of the grid configuration that you want to include in the package.

For more information about including security artifacts, see Include Security Artifacts.

Logic App Connections

You can package and deploy a Logic App Connection from one environment or repository to another environment or repository using the package feature in AVEVA Work Tasks.

To add Logic App Connection to Package Template

1. Log on to the **Enterprise Console**.
2. On the menu, point to **Tools**, and then click **Logic App Connections**.

   The **Logic App Connection** List window appears.
3. Select the Logic App connection that you want to package.
4. Right-click the Logic App connection and then click **Add to Package Template**.

   The Add to Package Template dialog box appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of a Logic App Connection in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

configuration properties of the Logic App Connection package in the **Settings** form on the right side of the Add to Package Template window.

Runtime Initialization Settings

You can configure runtime initialization settings for Logic App connections. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration Settings for Logic App Connection

The **Logic App Connection** has an unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment** > **Package Template** window
- **Target Environment** > **Manage Package** window

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Child Artifacts

Settings Tab

You can define the properties in the **Settings** tab to take appropriate actions to install, update, or uninstall a package.

Install

Use this tab to set the configurations for installing the package.

- **Connection Exists**: Select this property to take appropriate action when the Logic App connection in the package is already present in the target environment.
  - **Do not install connection**: Select this option to use the Logic App connection, present in the target environment.
  - **Install to overwrite the existing connection**: Select this option to overwrite the Logic App connection in the target environment with the one that is present in the package. During the package installation, backup will be taken before overwriting the existing artifact.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Update

Use the settings in this tab to update an installed package.

- **Delete connection**: Use this property to delete the Logic App connection present in the target environment.
  - **Yes**: Select this option to remove the Logic App connection from the target environment.

    Connection is modified: Select this option to retain, roll back to original state, or delete the Logic App connection in the target environment even after the package is updated. This property is applicable only if **Delete connection** is set to *Yes* and the Logic App connection is modified.
    - **Retain**: Select this option to retain the Logic App connection with the modifications in the target environment even after the uninstalling the package.
    - **Rollback to original state**: Select this option to rollback the Logic App connection to the original state. All modifications done to the Logic App connection after installation in the target environment will be rolled back. However, the Logic App connection will be retained in the target environment.
    - **Delete**: Select this option to delete the Logic App connection from the target environment while updating the package.
  - **No**: Select this option to retain the Logic App connection in the target environment.

    Connection exists: Select this option to take appropriate action when the Logic App connection that is being installed is present in the target environment.
    - **Do not install connection**: Select this option if you do not want to update the Logic App connection as associated artifacts are not present.
    - **Install to overwrite existing connection**: Select this option to overwrite the Logic App connection present in the target environment with the one that is currently being packaged.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete connection**: Select this property to retain or delete the Logic App connection in the target environment after you uninstall the package.
  - **Yes**: Select this option to delete the Logic App connection from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the Logic App connection in the target environment even after the package is uninstalled.
- **Connection is modified**: Select this property to retain, rollback, or delete the Logic App connection in the target environment after uninstalling the package. This property is displayed only if **Delete connection** is set to *Yes*. However, if **Delete connection** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the Logic App connection with the modifications in the target environment even after the package is uninstalled.
  - **Rollback to original state**: Select this option to rollback the Logic App connection to the original state. All modifications that were done to the connection after it was installed in the target environment will be reverted. However, the connection will be retained in the target environment.
  - **Delete**: Select this option to delete the Logic App connection from the target environment when uninstalling the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Connection Details

Field	Description
Connection String	Connection string for Logic App connection.

Associated Artifacts of Logic App Connection

Folder can be the associated artifact.

Child Artifacts of Logic App Connection

Logic App connection can be the child artifact for the folder.

Security Artifacts of Logic App Connection

You can enable the security settings of the Logic App connection that you want to include in the package.

For more information about including security artifacts, see Include Security Artifacts.

Lookup Settings

You can package and deploy Database Lookup and List Lookup along with its associated and child artifact from one environment or repository to another environment or repository.

To add Lookup Settings to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Lookup Settings**.

   The **Lookup List** window appears.
3. Select the lookup item that you want to package.
4. Right-click the lookup, and then click **Add to Package Template**.

   The **Add to Package Template** window appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of lookup in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the lookup package in the **Settings** form on the right side of the Add to Package Template window.

   When you select the **Include Associated Artifacts** option as *Yes* while adding the lookup to a package template, the associated artifacts are added to the package template. You can then view the associated artifacts of a lookup in the tree view of a package. If you select the **Include Associated Artifacts** option as *No*, the associated artifacts are not added to the package template. Then, you cannot see the associated artifacts in the tree view of a package. This condition holds good for both the **Include Child Artifacts** and **Include Security Artifacts** options as well.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

When you select the **Include the automatically generated SQL Script** option as *Yes* while adding the lookup to a package template, the automatically generated SQL script is included in the package template. If you select the **Include the automatically generated SQL Script** option as *No*, the automatically generated SQL script is not included in the package template.

Configuration settings for Lookup

Lookup has a unique set of configuration settings required to install, update, and uninstall the package. You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

Settings Tab

You can define the configuration settings of a lookup artifact in the **Settings** tab. Set the properties in this tab to take appropriate actions while installing, updating, or uninstalling the package.

Install

You can use this tab to set the configurations for installing the package. When you install a package containing a lookup using stored procedure, you must create stored procedure with the same name.

- **Lookup exists**: Use this property to take appropriate action when the lookup which is installed is present in the target environment.
  - **Do not install Lookup**: Select this option to use the lookup which is present in the target environment.
  - **Install to overwrite existing Lookup**: Select this option to overwrite the lookup which is present in the target environment with the one that is currently being packaged.
- **Associated artifacts not present**: Use this property to take appropriate action when the associated artifacts of the lookup is not present in the target environment.
  - **Install Lookup**: Select this option to install the lookup, even if the associated artifacts are not present in the target environment.
  - **Do not install Lookup**: Select this option if you do not want to install the lookup as its associated artifacts are not present.
- **SQL script**: When a lookup contains a query or a procedure, while adding the lookup to the package template, a script is generated and populated in the SQL script text area. While installing the package in the target environment, the SQL script is run first and then the package is installed.

  If the lookup contains a *query*, the script will be generated for the tables in the query. Parent tables and child tables are not considered. Such a query will contain the following information:
  - Table structure of the tables in the query, with field name, data types, and their order
  - Keys - primary key and foreign key
  - Constraints - default or check, and so on
  - Index

    If the lookup contains a *procedure*, a script for the procedure is generated. However, objects such as

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

tables or views used by the procedure are not generated. All such objects must be available in the target environment before package installation.

The following points are not considered while generating the SQL script:

- User defined types
- Table partitioning
- Views
- Triggers
- Functions
- Synonyms
- Parent and child tables
- Query against views and functions

If you need any of the unsupported objects, then you can choose to do one of the following:

- Write and run the script manually in SQL Server
- Write and append the script to the SQL script text area

The auto-generated script contains a pre-condition statement which checks for the existing tables in the target environment. If the table exists in the target environment, the SQL script does not run. If the table does not exist in the target environment, the SQL script runs and creates the table, and adds the keys, index, constraints as defined in the source environment.

Ensure that the object names used in the source repository does not conflict with the destination repository. If any of the objects (name of a table, key, constraint, or index) exists in the target environment, then the script execution will fail. This will be recorded in **Package History** page after the package installation.

As the script is auto-generated, we recommend not to modify the script. If you want to add your own piece of code sample to the script, you can either add it at the start of the script or append it to the end of the script. Do not add or edit the existing script.

Generating scripts for Oracle database is not supported. The SQL script text box will not be visible when the database in source environment is Oracle.

Update

Use the settings in this tab to update an installed package.

- **Delete Lookup**: Select this property to delete the lookup which is present in the target environment.
  - **Yes**: Select this option to remove the lookup from the target environment.

    Lookup is modified: Use this option to retain or delete the lookup in the target environment even after the package is updated. This property is applicable only if **Delete Lookup** is set to *Yes* and the lookup is modified.
    - **Retain**: Select this option to retain the lookup along with the modifications, in the target environment even after the package is uninstalled.
    - **Rollback to original state**: Select this option to rollback the lookup to the original state. All modifications that were done to the lookup after it was installed in the target environment will be reverted. However, the lookup will be retained in the target environment.
    - **Delete**: Select this option to delete the Lookup from the target environment when uninstalling the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

package.

- **No**: Select this option to retain the lookup in the target environment.

  Lookup exists Select this option to take appropriate action when the lookup that is being installed is present in the target environment.

  - **Do not install Lookup**: Select this option to use the lookup which is present in the target environment.
  - **Install to overwrite existing Lookup**: Select this option to overwrite the lookup which is present in the target environment with the one that is currently being packaged.

    Associated artifacts not present: Select this option to take appropriate action when the associated artifacts of the lookup is not present in the target environment.
  - **Update Lookup:** Select this option to update the lookup.
  - **Do not update Lookup**: Select this option if do not want to update the lookup as associated artifacts are not present.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete Lookup**: Select this property to retain or delete the lookup in the target environment even after the package is uninstalled.
  - **Yes**: Select this option to remove the lookup from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the lookup in the target environment even after the package is uninstalled.
- **Lookup is modified**: Use this property to retain, rollback to original state, or delete the lookup in the target environment even after the package is uninstalled. This property is displayed only if **Delete Lookup** is set to *Yes*. However, if **Delete Lookup** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the lookup along with the modifications, in the target environment even after the package is uninstalled.
  - **Rollback to original state**: Select this option to rollback to the original state. All modifications that were done to the lookup after it was installed in the target environment will be reverted. However, the lookup will be retained in the target environment.
  - **Delete**: Select this option to delete the lookup from the target environment when uninstalling the package.

Note:

- The SQL script does not run when the package is installed in update mode.

- Changes made to the package through SQL scripts will not be reverted during package uninstallation.

- Database Lookup script will be executed against the target database specified in **Database connection** artifact which is an associated artifact for the Lookup.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Associated Artifacts of Lookup

The Associated Artifacts tab displays all the associated artifacts of the selected lookup.

The following artifacts can appear as an associated artifact, if you are using any one of the following artifacts while designing the lookup:

- Parent Folder
- Database Connection (Query, Stored Procedure)
- Manage List
- Web API List

Child Artifacts of Lookup

Lookup does not have any child artifact.

Include Child Item

For more information, see Include Child Item.

Security Artifacts of Lookup

The Security Artifacts tab displays all the security artifacts of lookup, if any.

You can enable the security settings of the lookup that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

Note:

- Along with a Manage List containing Lookup display value, the list items will be packaged and installed in the target environment. The lookup display value will be blank in case the lookup display value in target environment does not match the lookup in the source environment.

- If a stored procedure used within the product is already existing in the target machine, then the user has to handle the stored procedure before installing the package

Lookup Example Scenarios

Scenario 1: Defining the uninstall properties for the lookup where the lookup is modified after it is installed in the target environment

Consider the following:

- A database lookup, *TelematicsDBLookup*
- The database lookup contains a query, **Select * from SWExecute**
- In the target environment, the query **Select * from SWExecute** is modified to **Select * from SWExecute Where ID=1**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To define the uninstall properties

1. Add the database lookup, *TelematicsDBLookup* to the package template, *TelematicsDBKLookupPackage*.
2. In the Package Template window, select *TelematicsDBLookup***,** go to the **Settings** page, and then click the **Uninstall** tab.
3. Define the **Uninstall** properties as the following:

   Lookup is modified: Use this property to retain, rollback the database lookup to the original state, or delete the database lookup in the target environment after the package is uninstalled.

   - **Retain**: Select this option to retain *TelematicsDBLookup* along with the modifications, that is, retain the modified query **Select * from SWExecute Where ID=1** in the target environment even after the package is uninstalled.
   - **Rollback to original state**: Select this option to rollback the database lookup to the original state, that is, revert the modifications and update the query as **Select * from SWExecute,** and retain the database lookup in the target environment even after the package is uninstalled.
   - **Delete**: Select this option to delete *TelematicsDBLookup* from the target environment while uninstalling the package.

Scenario 2:Defining the uninstall properties for a lookup where the lookup was already present in the target environment before installing the package

Consider the following:

- A database lookup, *TelematicsDBLookup2*, is already present in the target environment
- Description of *TelematicsDBLookup2* is *Lookup for Telematics2*
- The same database lookup, *TelematicsDBLookup*, with description as *Database Lookup for Telematics2* is being added the package template

1. Add the database lookup, *TelematicsDBLookup2*, with the description as *Database Lookup for Telematics2* to the package template, *TelematicsDBKLookupPackage2*.
2. In the **Package Template** window, select *TelematicsDBLookup2* and go to the **Settings** page, and then click the **Uninstall** tab.
3. Define the uninstall properties as following:
   - **Retain**: Select this option to retain *TelematicsDBLookup* with the updated description *Database Lookup for Telematics2* in the target environment even after the package is uninstalled.
   - **Rollback to original state**: Select this option to rollback to the original state, that is, retain the original lookup *TelematicsDBLookup2* with description as *Lookup for Telematics2* which was present in the target environment before installing the package.
   - **Delete**: Select this option to delete *TelematicsDBLookup* from the target environment when uninstalling the package.

Manage List

You can package and deploy the Manage List along with its associated artifacts and child artifacts from one environment or repository to another environment or repository. After installation, ensure that you refresh the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Enterprise Console site to add list items to the installed list.

When you package a Manage List, the list items get added to the package. However, while packaging a folder containing list items, the **Description** of the folder does not get packaged.

To add Manage List to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**.

   The **Manage List** window appears.
3. Select the Manage List that you want to package.
4. Right-click the manage list artifact and then click **Add to Package Template**.

   The **Add to Package Template** dialog box appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of the Manage List in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the Manage List package in the Settings form on the right side of the Add to Package Template window.
5. Configure the required settings.
6. Click **Save Template** to save the configured settings.

Configuration Settings for Manage List

The **Manage List** has an unique set of configuration settings required to install, update, and uninstall the package. You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

Settings

You can define the configuration settings of the Manage List artifact in the Settings tab. Set the properties in this tab to take appropriate actions while installing, updating, or uninstalling the package.

Install

Use this tab to set the configurations for installing the package.

- **Manage List exists**: Use this property take appropriate action when the Manage List definition present in the package is already present in the target environment.
  - **Install to overwrite existing Manage List**: Select this option to overwrite the Manage List definition that is present in the target environment with the one in the package.
  - **Do not install Manage List**: Select this option to use the Manage List definition that is present in the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Associated artifacts not present**: Use this property to take appropriate action when the associated artifacts of the Manage List definitions are not present in the target environment.
  - **Install Manage List**: Select this option to install the Manage List definition, even if the associated artifacts are not present in the target environment.
  - **Do not install Manage List**: Select this option to install the Manage List definition as its associated artifacts are not present.
- **Install all List items**: Use this property to take appropriate action when list items are included in the package.
  - **Yes**: Select this option to install the list items of Manage List.
  - **No**: Select this option if you do not want to install the list items of Manage List.

  Note: The default value is **No**. However, if you want to package list items, select **Yes**, and then generate and export the package.
- **SQL script**: When a Manage List contains an existing custom list, while adding the Manage List to the package template, a script is generated and populated in the SQL script text area. While installing the package in the target environment, the SQL script is run first and then the package is installed.

  The auto-generated script contains the the following:
  - Table structure, with field name, data types, and their order
  - Keys - primary key and foreign key
  - Constraints - default or check, and so on
  - Index
  - Child list and all the previously mentioned related scripts

    The following points are not considered while generating the SQL script:
  - User defined types
  - Table partitioning
  - Parent list
  - Views
  - Triggers
  - Functions
  - Synonyms

    If you need any of the unsupported objects, then you can choose to do one of the following:
  - Write and run the script manually in SQL Server
  - Write and append the script to the SQL script text area

    The auto-generated script contains a pre-condition statement which checks for the existing tables in the target environment. If the table exists in the target environment, the SQL script does not run. If the table does not exist in the target environment, the SQL script runs and creates the table, and adds the keys, index, constraints as defined in the source environment.

    Ensure that the object names used in the source repository does not conflict with the destination repository. If any of the objects (name of a table, key, constraint, or index) exists in the target environment, then the script execution will fail. This will be recorded in **Package History** page after the package installation.

    Similar scripts are also generated for all the child list items of the existing custom list. However, if a child list is being packaged, the SQL script will be generated only for that list and its child items. No SQL script

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

will be generated for the parent of the current list item being packaged.

As the script is auto-generated, we recommend not to modify the script. If you want to add your own piece of code sample to the script, you can either add it at the start of the script or append it to the end of the script. Do not add or edit the existing script.

Note:

- If the list exists in the target environment, the list items will not be installed.

- In the case of List items, ensure that the Table schema in the List matches in the target environment before you install the Manage List package to the target environment. If you install the Manage List package in the target environment without matching the schema, the package will install the List with the following warning:

You cannot create the list items. Either the Table schema is missing or not matching. Verify your Database.

Besides this, you might also get the same warning if you use a repository which is a case-sensitive database and the queries or columns in the target environment do not match as it is written in the source environment. You can resolve this issue by correcting the Table Schema details in the target environment.

- When you select the **Include the automatically generated SQL Script** option as *Yes* while adding the Manage List to a package template, the automatically generated SQL script is included in the package template. If you select the **Include the automatically generated SQL Script** option as *No*, the automatically generated SQL script is not included in the package template.

See Also: SQL Script Generation and Execution

Update

Use the settings in this tab to update an installed package.

- **Delete Manage List**: Use this property to delete or retain an existing Manage List definition that is present in the target environment.
  - **Yes**: Select this option to remove the Manage List definition in the target environment.

    Manage List is modified: Use this option to retain, rollback to original state, or delete the Manage List in the target environment even after the package is updated. This property is applicable only if **Delete Manage List** is set to *Yes* and the Manage List is modified.
    - **Retain**: Select this option to retain the Manage List definition along with the modifications, in the target environment even after the Package is updated.
    - **Rollback to original state**: Select this option to rollback to the original state. All modifications that were done to the Manage List definition after it was installed in the target environment will be reverted. However, the Manage List definition will be retained in the target environment.
    - **Delete**: Select this option to delete the Manage List definition from the target environment when updating the package.
  - **No**: Select this option if you want to retain the Manage List definition in the target environment.

    Manage List exists: Use this option to take appropriate action when the Manage List that is being installed is present in the target environment.
    - **Use existing Manage List**: Select this option to use the Manage List definition that is present in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

target environment.

- **Do not update Manage List**: Select this option to overwrite the Manage List definition that is present in the target environment with the one present in the package.

  Associated artifacts not present: Use this option to take appropriate action if associated artifact is not present in the target environment.
- **Update Manage List**: Select this option to update the Manage List definition.
- **Do not update Manage List**: Select this option if you do not want to update the Manage List definition as associated artifacts are not present.

Note:

- Update and uninstall of list items is not supported.

- While updating a list item through package updation, if you overwrite the schema of the existing list in the target environment through package, the existing list items will get corrupted.

- The SQL script does not run when the package is installed in update mode.

- Changes made to the package through SQL scripts will not be reverted during package uninstallation.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete Manage List**: Use this property to decide if you want to retain or delete the Manage List definition in the target environment even after the package is uninstalled.
  - **Yes**: Select this option to remove the Manage List definition from the target environment when the package is uninstalled
  - **No**: Select this option to retain the Manage List definition in the target environment even after the package is uninstalled
- **Manage List is modified**: Use this property to retain, rollback to original state, or delete the Manage List definition in the target environment even after the package is uninstalled. This property is displayed only if **Delete Manage List** is set to *Yes*. However, if **Delete Manage List** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the Manage List definition along with the modifications in the target environment even after the package is uninstalled.
  - **Rollback to original state**: Select this option to rollback to the original state. All modifications that were done to the Manage List definition after it was installed in the target environment will be reverted. However, the Manage List definition will be retained in the target environment.
  - **Delete**: Select this option to delete the Manage List definition from the target environment when uninstalling the package.

Associated Artifacts of Manage List

The **Associated Artifacts** tab displays all the associated artifacts of the selected Manage List definition.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Manage List can include the following Associated Artifacts:

- Parent Folder
- Lookup
- Form
- Attachments. You can have *n* number of attachments.

Child Artifacts of Manage List

The **Child Artifacts** tab displays all the child artifacts of the Manage List definition.

Manage List can include the following child artifacts:

- Folder
- Child Manage List

Include Child Item

For more information, see Include Child Item.

Security Artifacts of Manage List

The Security Artifacts tab displays the security artifacts of a manage list definition, if any. You can enable the security settings of the manage list that you want to include in the package.

For more information about including security artifacts, see Include Security Artifacts.

Note:

- Manual uninstallation of forms may be required when uninstalling packages in certain scenarios. For example, a package with a form associated to a generic list is created in the source repository and then installed, reinstalled, and uninstalled from the target repository as follows:

Installation: Package is imported and installed. The associated form (version 1) and the generic list (version 1) are created.

Reinstallation: Same package is installed again with the default settings. The next minor version of the associated form (version 1.1) is created but the list (version 1) is not overwritten.

Uninstallation: Package is uninstalled. The minor version of the associated form (version 1.1) and the generic list (version 1) are removed but the previously associated version of the form (version 1) is retained.

- For a custom Manage List definition packaged with particular Manage List item types and installed, while uninstalling only the list items installed through the package will be uninstalled. Any list item created for the Manage List after installing the package will not be uninstalled during package uninstallation.

- Along with a Manage List containing User Lookup, the list items will be packaged and installed in the target environment. The user lookup field will be blank in case the user provider in target environment does not match the user provider in the source environment. For example, if you select a user in user lookup from active

---

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

directory provider and install in the target repository with AVEVA Work Tasks list provider, the List items will be installed , but the user lookup field will remain blank. In this case, if the user lookup field is mandatory, the list item will not be installed as it cannot be left blank.

- Update and uninstall for List items are not supported through package. During uninstall, the Manage List even though marked for deletion will not be deleted as List items cannot be deleted through package.

Manage View

You can package and deploy the Manage View definition from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks. The Manage View artifacts installed will be listed in the tree view and displays the name of the List View along with the artifact name in the parenthesis.

You can package the available views for the following artifacts:

- Workflows
- SOA Assembly
- SOA Folder
- Manage Forms
- Users
- Form Controls
- Report Connection
- Report Templates
- Grid Configuration
- Grid View
- Reports
- File Watcher
- Manage Package
- Package Template
- Lookup Settings
- Manage View
- Database Connections
- Deleted List Items
- Impersonation User
- External Connections

To add Manage View to Package Template

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, point to **Tools**, and then click **Manage Views**.
   The **Manage Views** window appears.
3. Select the List View that you want to package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Right-click the List View and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Note: Ensure that the List Views are in Published state in order to add the selected List Views to a package template.

Configuration Settings for Manage View

The **Manage View** has a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template > Edit**
- **Target Environment > Manage Package > Edit**

You can use the **Settings** tab to define the configuration settings. The **Associated Artifacts** and the **Child Artifacts** tabs are not available as Manage View does not contain any associated or child artifacts.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Security Artifacts

Settings

The properties in this tab allow you to take appropriate actions while installing, updating, or uninstalling the Package.

Note: In the case of List view, ensure that the Table is present in the target environment before you install the List View package in the target environment. When you are installing the List view package in the target environment, if the table is not present in the target environment, the following message appears: **You cannot view the list items**. **The table is missing**. **Verify your Database**. Besides this, you may also get the above error message if you are using a repository which is a case-sensitive database and the queries/columns in the target environment do not match as it is written in the source environment. You can resolve this issue by correcting the Table Name or creating the table in the target environment.

Install

The settings in this tab are referred to when installing the Package.

- **List View Exists**: This property allows you to take appropriate action when the List View definition present in the package is already present in the target environment.
  - **Do not install List view**: Select this option if you want to use the List View definition that is present in the target environment.
  - **Install to overwrite existing List View**: Select this option if you want to overwrite the List View definition that is present in the target environment with the one in the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Default List View**: This property allows you to specify default list view in the target environment. This will create a new default list view or overwrite an existing default list view.
  - **Yes**: Select this option to create the List View as the default List View in the target environment. If an existing List View is already available in the target environment, the selected List View will replace the existing one as default in the target environment.
  - **No**: Select this option if you do not want to specify the default List View definition in the target environment.
- **Associated Artifacts Not Present**: Use this property to take appropriate action when the associated artifacts of the List View are not present in the target environment.
  - **Install List View**: Select this option to install the List View in the Draft status, even if the associated artifacts are not present in the target environment.
  - **Do not install List View**: Select this option if you do not want to install the List View as its associated artifacts are not present.

Update

The settings in this tab are referred to when the same Package is installed for the consecutive time.

- **Delete List View**: Select this option if you want to delete the List View definition that is present in the target environment.
  - **Yes**: Select this option to remove the List View definition in the target environment. On selection of this option, the **List View Is Modified** option appears.

    List View Is Modified: Use this option to retain or delete the List View in the target environment even after the package is updated. This property is applicable only if **Delete List View** is set to *Yes* and the List View is modified.
    - **Retain**: Select this option if you want to retain the List View definition along with the modifications, in the target environment even after the Package is updated.
    - **Rollback**: Select this option if you want to roll back to the original state. All modifications that were done to the List View definition after it was installed in the target environment will be reverted. However, the List View definition will be retained in the target environment.
    - **Delete**: Select this option if you want to delete the List View definition from the target environment when updating the Package.
  - **No**: Select this option if you want to retain the List View definition in the target environment. On selection of this option, the **List View Exists** option appears.

    List View Exists: This property allows you to take appropriate action when the List View definition present in the package is already present in the target environment.
    - **Do not install List View**: Select this option if you want to use the List View definition that is present in the target environment
    - **Install to overwrite existing List View**: Select this option if you want to overwrite the List View definition that is present in the target environment with the one present in the Package.
- **Default List View**: This property allows you to specify default list view in the target environment. This will create a new default list view or overwrite an existing default list view.
  - **Yes**: Select this option to create the List View as default in the target environment. If an existing list view is already available in the target environment, the selected List View will replace the existing one as default in the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **No**: Select this option if you do not want to specify the Package List View as default List View definition in the target environment.

- **Associated Artifacts Not Present**: Use this property to take appropriate action when the associated artifacts of the List View are not present in the target environment.
  - **Install List View**: Select this option to update the List View in the Draft status, even if the associated artifacts are not present in the target environment.
  - **Do not install List View**: Select this option if you do not want to update the List View as its associated artifacts are not present.

Uninstall

The settings in this tab are referred to when the Package is being uninstalled in the target environment.

- **Delete List View**: This property allows you to decide if you want to retain or delete the List View definition in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option if you want to remove the List View definition from the target environment when the Package is uninstalled.
  - **No**: Select this option if you want to retain the List View definition in the target environment even after the Package is uninstalled.
- **List View Is Modified**: This option allows you to decide if you want to retain, roll back to original state or delete the List View definition in the target environment even after the Package is uninstalled. This property is displayed only if **Delete List View** is set to *Yes.* However, if **Delete List View** is set to *No*, this property is not displayed.
  - **Retain**: Select this option if you want to retain the List View definition along with the modifications, in the target environment even after the Package is uninstalled.
  - **Roll back to original State**: Select this option if you want to roll back to the original state. All modifications that were done to the List View definition after it was installed in the target environment will be reverted. However, the List View definition will be retained in the target environment.
  - **Delete**: Select this option if you want to delete the List View definition from the target environment when uninstalling the Package.

Note: Consider a scenario when a "Default list view" of Manage Package List that is installed in the target environment is packaged, and the same List View is selected to view the Manage Package List items. In this scenario, if you uninstall the package that had installed the Default List View, then the package will be uninstalled successfully. However, an exception occurs when the **Package History** page is closed. This exception occurs because the List View that you are using to view the list items, gets deleted when the package is uninstalled.

Associated Artifacts tab

This tab is not available as the List View does not have any associated artifact.

Child Artifacts tab

This tab is not available as the List View does not have any child artifact.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Security Artifacts tab

The Security Artifacts tab displays the security artifacts of a List View, if any.

You can enable the security settings of the manage view that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

Message Queue Connections

You can package and deploy a Message Queue Connection from one environment or repository to another environment or repository using the package feature in AVEVA Work Tasks.

To add Message Queue Connection to Package Template

1. Log on to the **Enterprise Console**.
2. On the menu, point to **Tools**, and then click **Message Queue Connections**.

   The **Message Queue** Connection List window appears.
3. Select the Message Queue Connection that you want to package.
4. Right-click the Message Queue Connection and then click **Add to Package Template**.

   The Add to Package Template dialog box appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of a Message Queue Connection in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the Message Queue Connection package in the **Settings** form on the right side of the Add to Package Template window.

Runtime Initialization Settings

You can configure runtime initialization settings for Message Queue Connections. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration Settings for Message Queue Connection

The **Message Queue Connection** has an unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment** > **Package Template** window
- **Target Environment** > **Manage Package** window

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

    - Uninstall
- Associated Artifacts
- Child Artifacts

Settings Tab

You can define the properties in the **Settings** tab to take appropriate actions to install, update, or uninstall a package.

Install

Use this tab to set the configurations for installing the package.

- **Connection Exists**: Select this property to take appropriate action when the Message Queue Connection in the package is already present in the target environment.
    - **Do not install connection**: Select this option to use the Message Queue Connection, present in the target environment.
    - **Install to overwrite the existing connection**: Select this option to overwrite the Message Queue Connection in the target environment with the one that is present in the package. During the package installation, backup will be taken before overwriting the existing artifact.

Update

Use the settings in this tab to update an installed package.

- **Delete connection**: Use this property to delete the Message Queue Connection present in the target environment.
    - **Yes**: Select this option to remove the Message Queue Connection from the target environment.

      Connection is modified: Select this option to retain, roll back to original state, or delete the Message Queue Connection in the target environment even after the package is updated. This property is applicable only if **Delete connection** is set to *Yes* and the Message Queue Connection is modified.
        - **Retain**: Select this option to retain the Message Queue Connection with the modifications in the target environment even after the uninstalling the package.
        - **Rollback to original state**: Select this option to rollback the Message Queue Connection to the original state. All modifications done to the Message Queue Connection after installation in the target environment will be rolled back. However, the Message Queue Connection will be retained in the target environment.
        - **Delete**: Select this option to delete the Message Queue Connection from the target environment while updating the package.
    - **No**: Select this option to retain the Message Queue Connection in the target environment.

      Connection exists: Select this option to take appropriate action when the Message Queue Connection that is being installed is present in the target environment.
        - **Do not install connection**: Select this option if you do not want to update the Message Queue Connection as associated artifacts are not present.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Install to overwrite existing connection**: Select this option to overwrite the Message Queue Connection present in the target environment with the one that is currently being packaged.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete connection**: Select this property to retain or delete the Message Queue Connection in the target environment after you uninstall the package.
  - **Yes**: Select this option to delete the Message Queue Connection from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the Message Queue Connection in the target environment even after the package is uninstalled.
- **Connection is modified**: Select this property to retain, rollback, or delete the Message Queue Connection in the target environment after uninstalling the package. This property is displayed only if **Delete connection** is set to *Yes*. However, if **Delete connection** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the Message Queue Connection with the modifications in the target environment even after the package is uninstalled.
  - **Rollback to original state**: Select this option to rollback the Message Queue Connection to the original state. All modifications that were done to the connection after it was installed in the target environment will be reverted. However, the connection will be retained in the target environment.
  - **Delete**: Select this option to delete the Message Queue Connection from the target environment when uninstalling the package.

Connection Details

Use the **Connect to Repository DataSource** settings to choose between an existing repository connection string or a different connection string.

Source:

Field	Description
User Name	Displays the use name with which the connection is configured .
Password	Displays the password in encrypted format.
Additional Settings	Displays the Message Queue connection additional settings.
Queue Name	Displays the name of the Queue.

Target:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
User Name	Enter the user name with which you want to configure connection.
Password	Enter the password.
Additional Settings	Enter the Message Queue connection additional settings.
Queue Name	Enter the name of the Queue.

Associated Artifacts of Message Queue Connection

Message Queue Connection does not have any associated artifact.

Child Artifacts of Message Queue Connection

Message Queue Connection does not have any child artifact.

Security Artifacts of Message Queue Connection

You can enable the security settings of the Message Queue Connection that you want to include in the package.

For more information about including security artifacts, see Include Security Artifacts.

Note: Restart the Workflow Engine service to utilize the Message Queue Connection installed through package.

MQTT Connections

You can package and deploy a MQTT Connection from one environment or repository to another environment or repository using the package feature in AVEVA Work Tasks.

To add MQTT Connection to Package Template

1. Log on to the **Enterprise Console**.
2. On the menu, point to **Tools**, and then click **MQTT Connections**.

   The **MQTT** Connection List window appears.
3. Select the MQTT Connection that you want to package.
4. Right-click the MQTT Connection and then click **Add to Package Template**.

   The Add to Package Template dialog box appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of a MQTT Connection in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the MQTT Connection package in the **Settings** form on the right side of the Add to Package Template window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Runtime Initialization Settings

You can configure runtime initialization settings for MQTT Connections. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration Settings for MQTT Connection

The **MQTT Connection** has an unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment** > **Package Template** window
- **Target Environment** > **Manage Package** window

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Child Artifacts

Settings Tab

You can define the properties in the **Settings** tab to take appropriate actions to install, update, or uninstall a package.

Install

Use this tab to set the configurations for installing the package.

- **Connection Exists**: Select this property to take appropriate action when the MQTT Connection in the package is already present in the target environment.
  - **Do not install connection**: Select this option to use the MQTT Connection, present in the target environment.
  - **Install to overwrite the existing connection**: Select this option to overwrite the MQTT Connection in the target environment with the one that is present in the package. During the package installation, backup will be taken before overwriting the existing artifact.

Update

Use the settings in this tab to update an installed package.

- **Delete connection**: Use this property to delete the MQTT Connection present in the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Yes**: Select this option to remove the MQTT Connection from the target environment.

  Connection is modified: Select this option to retain, roll back to original state, or delete the MQTT Connection in the target environment even after the package is updated. This property is applicable only if **Delete connection** is set to *Yes* and the MQTT Connection is modified.

  - **Retain**: Select this option to retain the MQTT Connection with the modifications in the target environment even after the uninstalling the package.
  - **Rollback to original state**: Select this option to rollback the MQTT Connection to the original state. All modifications done to the MQTT Connection after installation in the target environment will be rolled back. However, the MQTT Connection will be retained in the target environment.
  - **Delete**: Select this option to delete the MQTT Connection from the target environment while updating the package.
- **No**: Select this option to retain the MQTT Connection in the target environment.

  Connection exists: Select this option to take appropriate action when the MQTT Connection that is being installed is present in the target environment.

  - **Do not install connection**: Select this option if you do not want to update the MQTT Connection as associated artifacts are not present.
  - **Install to overwrite existing connection**: Select this option to overwrite the MQTT Connection present in the target environment with the one that is currently being packaged.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete connection**: Select this property to retain or delete the MQTT Connection in the target environment after you uninstall the package.
  - **Yes**: Select this option to delete the MQTT Connection from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the MQTT Connection in the target environment even after the package is uninstalled.
- **Connection is modified**: Select this property to retain, rollback, or delete the MQTT Connection in the target environment after uninstalling the package. This property is displayed only if **Delete connection** is set to *Yes*. However, if **Delete connection** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the MQTT Connection with the modifications in the target environment even after the package is uninstalled.
  - **Rollback to original state**: Select this option to rollback the MQTT Connection to the original state. All modifications that were done to the connection after it was installed in the target environment will be reverted. However, the connection will be retained in the target environment.
  - **Delete**: Select this option to delete the MQTT Connection from the target environment when uninstalling the package.

Connection Details

Use the **Connect to Repository DataSource** settings to choose between an existing repository connection string or a different connection string.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Source:

Field	Description
Server Name	Displays the name of the MQTT Server.
Port	Displays the Port number of the Web Socket protocol in the MQTT server.
User Name	Displays the User Name for the MQTT Connection. **Note:** User Name is not available if **Anonymous** is set to **Yes** while configuring MQTT Connection.
Password	Displays the Password for the MQTT Connection. **Note:** Password is not available if **Anonymous** is set to **Yes** while configuring MQTT Connection.

Target:

Field	**Description**
Server Name	Enter the name of the MQTT Server.
Port	Enter the Port number of the Web Socket protocol in the MQTT server.
User Name	Enter the User Name for the MQTT Connection. **Note:** User Name is not available if **Anonymous** is set to **Yes** while configuring MQTT Connection.
Password	Enter the Password for the MQTT Connection. **Note:** Password is not available if **Anonymous** is set to **Yes** while configuring MQTT Connection.

Associated Artifacts of MQTT Connection

MQTT Connection does not have any associated artifact.

Child Artifacts of MQTT Connection

MQTT Connection does not have any child artifact.

Security Artifacts of MQTT Connection

You can enable the security settings of the MQTT Connection that you want to include in the package.

For more information about including security artifacts, see Include Security Artifacts.

Note: Restart the Workflow Engine service to utilize the MQTT Connection installed through package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Navigation List

You can package and deploy the Navigation List item along with its associated and child artifact from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

After install, ensure that you refresh the Enterprise Console site to view the installed links.

You can package Navigation lists of type Navigation Link, List Link, and Form Link. For all these three types of links, you can update the source path of the icon. Additionally, for the Navigation Link, you can also update the name of the page to which you want to navigate to.

You can also package folders. For more information about adding folders to package template, see Folders.

To add Navigation List to Package Template

1. Log on to the **Enterprise Console**.
2. Click the <**User**> icon, and then click **Repository Settings**. The **Repository Settings** page appears.
3. Click the **Advanced Settings** tab, and then click **Manage Navigation List**. The **Navigation List** page appears.
4. Select the Navigation List that you want to package.
5. Right-click the Navigation List and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Configuration Settings for Navigation List

The **Navigation List** has a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template > Edit**
- **Target Environment > Manage Package List > Edit**

You can use the **Settings** tab to define the configuration settings. The **Associated Artifacts** tab displays the associated artifacts of the **Navigation List** item. The **Child Artifacts** tab displays child artifacts of the **Navigation List** item.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Child Artifacts
- Security Artifacts

Settings tab

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The properties in this tab allow you to take appropriate actions while installing, updating, or uninstalling the Package.

Install

The settings in this tab are referred to when installing the Package.

- **Navigation Link Exists**: This property allows you to take appropriate action when the Navigation List item present in the package is already present in the target environment
  - **Do Not Install Navigation Link**: Select this option if you want to use the Navigation List item that is present in the target environment.
  - **Install to overwrite existing Navigation Link**: Select this option if you want to overwrite the Navigation List item that is present in the target environment with the one in the package.
- **Associated Artifacts Not Present**: This property allows you to take appropriate action when the associated artifacts of the Navigation List items are not present in the target environment.
  - **Install Navigation Link**: Select this option if you want to install the Navigation List item, even if the associated artifacts are not present in the target environment.
  - **Do Not Install Navigation Link**: Select this option if you do not want to install the Navigation List item as its associated artifacts are not present.

Update

The settings in this tab are referred to when the same Package is installed for the consecutive time.

- **Delete Navigation Link**: Select this option if you want to delete the Navigation List item that is present in the target environment.
  - **Yes**: Select this option to remove the Navigation List item in the target environment. On selection of this option, the **Navigation Link Is Modified** option appears as shown in the following table.

    Navigation Link Is Modified: Use this option to retain or delete the Navigation Link in the target environment even after the package is updated. This property is applicable only if **Delete Navigation Link** is set to *Yes* and the Navigation Link is modified.
    - **Retain**: Select this option if you want to retain the Navigation List item along with the modifications, in the target environment even after the Package is updated.
    - **Rollback to original state**: Select this option if you want to roll back to the original state. All modifications that were done to the Navigation List item after it was installed in the target environment will be reverted. However, the Navigation List item will be retained in the target environment.
    - **Delete**: Select this option if you want to delete the Navigation List item from the target environment when updating the Package.
  - **No**: Select this option if you want to retain the Navigation List item in the target environment. On selection of this option, the Navigation Link Exists and the Associated Artifacts not present options appear.

    Navigation Link Exists: This property allows you to take appropriate action when the Navigation Link definition present in the package is already present in the target environment.
    - **Do Not Install Navigation Link**: Select this option if you want to use the Navigation Link definition that is present in the target environment.
    - **Install to overwrite existing Navigation Link**: Select this option if you want to overwrite the Navigation Link definition that is present in the target environment with the one present in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Package.

Associated Artifacts Not Present: Use this property to take appropriate action when the associated artifacts of the Navigation Link are not present in the target environment.

- **Update Navigation Link**: Select this option to update the Navigation Link in the Draft status, even if the associated artifacts are not present in the target environment.
- **Do not update Navigation Link**: Select this option if you do not want to update the Navigation List items as its associated artifacts are not present.

Uninstall

The settings in this tab are referred to when the Package is being uninstalled in the target environment.

- **Delete Navigation Link**: This property allows you to decide if you want to retain or delete the Navigation List in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option if you want to remove the Navigation List from the target environment when the Package is uninstalled.
  - **No**: Select this option if you want to retain the Navigation List in the target environment even after the Package is uninstalled.
- **Navigation Link Is Modified**: This option allows you to decide if you want to retain, roll back to original state or delete the Navigation List item in the target environment even after the Package is uninstalled. This property is displayed only if **Delete Navigation Link** is set to *Yes*. However, if **Delete Navigation Link** is set to *No*, this property is not displayed.
  - **Retain**: Select this option if you want to retain the Navigation List item along with the modifications, in the target environment even after the Package is uninstalled.
  - **Roll back to original State**: Select this option if you want to roll back to the original state. All modifications that were done to the Navigation List item after it was installed in the target environment will be reverted. However, the Navigation List item will be retained in the target environment.
  - **Delete**: Select this option if you want to delete the Navigation List item from the target environment when uninstalling the Package.

Scenario

Consider a scenario where a package with many artifacts, including Navigation List items, is installed.

- While installing a package which has customized menu structure, ensure that the **Install to overwrite existing Navigation Link** option is selected.
- If you want to revert the customized menu structure to the original menu structure, ensure that **Rollback to original state** option in the **Uninstall** tab is set for all the artifacts in the Navigation List items.

Associated Artifacts for Navigation List Item

This tab displays all the associated artifacts of the selected Navigation List item.

Navigation List can include the following Associated Artifacts:

- Parent Folder
- List

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Form
- Parent Navigation List item

Child Artifacts for Navigation List Item

This tab displays all the child artifacts of the Navigation List item.

Navigation List Item can include the following child artifacts:

- Folder
- Child Navigation List Item

Security Artifacts for Navigation List Item

The Security Artifacts tab displays the security artifacts of a Navigation List, if any.

You can enable the security settings of the navigation list item that you want to include in the package.

For more information about including security artifacts, see Include Security Artifacts.

Navigation List Folders

For a Navigation List, you can package and deploy **Folder** along with its associated and child artifacts from one environment or repository to another environment or repository.

Configuration Settings for Folder

The **Folder** has a unique set of configuration settings required to install, update, and uninstall the package. You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template > Edit**
- **Target Environment > Manage Package List > Edit**

Settings Tab

You can define the configuration settings of the Folder artifact in the **Settings** tab. Set the options in this tab to take appropriate actions while installing, updating, or uninstalling the package.

Install

Use this tab to set the configurations for installing the package.

- **Parent item is not present**: Select this property to take appropriate action when the parent item is not present in the target environment.
  - **Install folder**: Select this option to install this folder in target environment even if the parent folder is not present.
  - **Do not install folder**: Select this option if you do not want to install the folder, as the parent folder is not present in the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Folder exists**: Select this property to take appropriate action when the folder in the package is present in the target environment.
  - **Do not install folder**: Select this option if you do not want to overwrite the existing folder in the target environment with the folder in the package.
  - **Install to overwrite existing folder**: Select this option to overwrite the existing folder in the target environment with the folder in the package. A backup of the existing folder is taken before overwriting the folder.

    If this option is selected and if the folder in the package is present in the target environment, only then the **Parent item is not present** option is considered.

Update

Use the settings in this tab to update an installed package.

- **Parent folder is not present**: Select this property to take appropriate action when the parent folder is not present in the target environment.
  - **Update folder**: Select this option to update the folder in target environment even if the parent folder is not present.
  - **Do not update folder**: Select this option if you do not want to update the folder, as the parent folder is not present in the target environment.
- **Folder exists**: Use this property to take appropriate action when the folder being packaged is present in the target environment.
  - **Do not install folder**: Select this option if you do not want to overwrite the existing folder in the target environment with the folder in the package.
  - **Install to overwrite existing folder**: Select this option to overwrite the existing folder in the target environment with the folder in the package. A backup of the existing folder is taken before overwriting the folder.

    If this option is selected and if the folder in the package is present in the target environment, only then the **Parent item is not present** option is considered.

Installation and Update Scenarios

Consider that the folder to be packaged is "Folder X", which is available under "Folder A" in the source environment. The target environment contains "Folder X", which is available under "Folder B". However, "Folder A" does not exist in the target environment.

- *Scenario 1*: While adding "Folder X" to the package template, if the value for the **Include Associated Artifact** option is set as **Yes**, then "Folder X" will be added along with "Folder A", to the package template. If the **Install to overwrite existing folder** option is selected, then the following changes take place in the target environment, after the package is installed or updated:
  - "Folder A" is installed
  - "Folder X" is overwritten and moved from "Folder B" to "Folder A".
- *Scenario 2*: While adding "Folder X" to the package template, if the value for the **Include Associated Artifact** option is set as **No**, then only "Folder X" will be added to the package template. If the **Install to overwrite existing folder** option and the **Install folder** option under **Parent item is not present** option are selected,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

then after the package is installed or updated, "Folder X" is overwritten and moved from "Folder B" to the root level.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete folder**: Select this property to decide if you want to delete the folder or retain it when uninstalling the package.
  - **Yes**: Select this option to delete this folder after uninstalling the package.
  - **No**: Select this option to retain this folder after uninstalling the package.
- **If folder modified after installation**: Select this option to retain, rollback to original state, or delete the Navigation List Folder in the target environment. This property is displayed only if **Delete folder** is set to *Yes*. However, if **Delete Folder** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the Navigation List Folder definition along with the modifications in the target environment, even after the Package is uninstalled.
  - **Rollback to original state**: Select this option to roll back to the original state. All modifications that were done to the Navigation List Folder definition after it was installed in the target environment will be reverted. However, the Navigation List Folder definition will be retained in the target environment.

    If this option is selected, then the following changes take place in the target environment:
    - The folder definition is rolled back to the definition that was available before the installation.
    - The folder will not be rolled back to its original location. However, if you want to roll back the folder to its original location, use the **Organize** option from the Ribbon bar.
  - **Delete**: Select this option to delete the Navigation List Folder definition from the target environment.

Scenario

Consider a scenario where a package with many artifacts, including Navigation List folders, is installed.

- While installing a package which has customized menu structure, ensure that the **Install to overwrite existing folder** option is selected.
- If you want to revert the customized menu structure to the original menu structure, ensure that **Rollback to original state** option in the **Uninstall** tab is set for all the artifacts in the Navigation List folders.

Associated Artifacts of Folder

The **Associated Artifacts** tab displays all the associated artifacts of the selected folder.

Folder can include the following associated artifacts:

- Parent Folder
- Parent ListItem

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Child Artifacts of Folder

The **Child Artifacts** tab displays all the child artifacts of the selected folder.

Folder can include the following child artifacts:

- Child Folder
- Child ListItem

Include Child Item

For more information, see Include Child Item.

Security Artifacts of Folder

The **Security Artifacts** tab displays all the security artifacts of the selected folder.

You can enable the security settings of the folder that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

Resolve Custom Dependent Files for Navigation Links

Configure Navigation Links

When you package Navigation Links the image, aspx page, and aspx.cs files are automatically identified and appear as custom dependent files. The **Source Files** field displays the physical path of these files in the source environment. When the package is installed, these files are copied to the installed path of the target environment.

Click **Add row** to add multiple rows to install additional files in the target environment. In the **Source Files** field, enter the source file path of the custom dependent files.

If the source folder is relative to the AVEVA Work Tasks installed path, and the folder does not exist in the target environment then a new folder is created and the files are copied to it. If the source folder is not relative to the AVEVA Work Tasks installed path then the files are copied to the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\bin* folder.

The **Install in Global Assembly Cache** field is optional and is used to install the custom dependent files in the GAC of the target environment.

Queue Management

You can package and deploy a queue along with its associated and child artifacts from one environment or repository to another environment or repository.

Users are not be added to the queue. When the queue is installed, the user installing the queue will become the owner of the queue. When the queue is installed, the role will be added to the queue only if the role name matches in any of the target providers.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To add Queue Management to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Queue Management**.

   The **Queue Management** window appears.
3. Right-click on a queue, and then click **Add to Package Template**.

   The **Add to Package Template** window appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of queue in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the queue package in the **Settings** form on the right side of the Add to Package Template window.

   When you select the **Include Associated Artifacts** option as *Yes* while adding the queue to a package template, the associated artifacts are added to the package template. You can then view the associated artifacts of a queue in the tree view of a package. If you select the **Include Associated Artifacts** option as *No*, the associated artifacts are not added to the package template. Then, you cannot see the associated artifacts in the tree view of a package. This condition holds good for both the **Include Child Artifacts** and **Include Security Artifacts** options as well.

   When you select the **Include the automatically generated SQL Script** option as *Yes* while adding the queue to a package template, the automatically generated SQL script is included in the package template. If you select the **Include the automatically generated SQL Script** option as *No*, the automatically generated SQL script is not included in the package template.

Runtime Initialization Settings

You can configure runtime initialization settings for queues. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration settings for Queue

Queue has a unique set of configuration settings required to install, update, and uninstall the package. You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template > Edit**
- **Target Environment > Manage Package > Edit**

Settings

You can define the configuration settings of a queue artifact in the **Settings** tab. Set the properties in this tab to take appropriate actions while installing, updating, or uninstalling the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Install

Use this tab to set the configurations for installing the package.

- **Queue exists**: Use this property to take appropriate action when the Queue to be installed is present in the target environment.
  - **Overwrite existing Queue**: Select this option to overwrite the Queue present in the target environment with the one that you packaged.
  - **Use existing Queue**: Select this option to use the Queue which is present in the target environment.
- **SQL script**: When a queue contains a custom dispatch pattern, while adding the queue to the package template, a script is generated and populated in the SQL script text area. While installing the script in the target environment, the SQL script is run first to make necessary database entry related to the custom dispatch provider, so that the queue with the custom dispatch provider works without any manual intervention.

  If a default dispatch pattern is configured, then the SQL scripts are not generated. However, if a custom dispatch pattern is configured, then the SQL scripts are generated based on the PUSH and/or PULL patterns.

  See Also: SQL Script Generation and Execution

Update

Use the settings in this tab to update an installed package.

- **Delete Queue**: Use this property to delete the Queue present in the target environment.
  - **Yes**: Select this option to remove the Queue from the target environment.

    If Queue is modified after installation: Use this option to retain, delete, or rollback to original state the queue in the target environment even after the package is updated. This property is applicable only if **Delete Queue** is set to *Yes* and the queue is modified.
    - **Retain**: Select this option to retain the Queue along with the modifications, in the target environment even after the package is updated.
    - **Rollback to original state**: Select this option if you want to rollback to the original state. All modifications that were done to the Queue after it was installed in the target environment will be reverted. However, the Queue is retained in the target environment.
    - **Delete**: Select this option if you want to delete the Queue from the target environment when updating the package.
  - **No**: Select this option to retain the Queue in the target environment.

    Queue exists: Use this option to take appropriate action when the queue that is being installed is present in the target environment.
    - **Overwrite existing Queue**: Select this option to overwrite the Queue that is present in the target environment with the one present in the package.
    - **Use existing Queue**: Select this option to use the Queue that is present in the target environment.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Delete Queue**: Use this property to decide if you want to retain or delete the Queue in the target environment after you uninstall the package.
  - **Yes**: Select this option to delete the Queue from the target environment while uninstalling the package.
  - **No**: Select this option to retain the Queue in the target environment after uninstalling the package.
- **If Queue is modified after installation**: Use this property to decide if you want to retain, rollback, or delete the Queue in the target environment after uninstalling the package. This property is displayed only if **Delete Queue** is set to *Yes.* However, However, if **Delete Queue** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the Queue with the modifications in the target environment after uninstalling the package.
  - **Rollback to original state**: Select this option to rollback to the original state. All modifications that were done to the Queue after the installation in the target environment is. However, the Queue is retained in the target environment.
  - **Delete**: Select this option to delete the Queue from the target environment when uninstalling the package.

Note:

- The SQL script does not run when the package is installed in update mode.

- Changes made to the package through SQL scripts will not be reverted during package uninstallation.

Associated Artifacts of Queue

The Associated Artifacts tab displays all the associated artifacts of the selected Queue.

The following artifacts can appear as Associated Artifacts, if you are using any one of the following artifacts while designing the queue:

- Workflow
- Work Item Field
- Work Item Layout
- Queue

Child Artifacts

Queue does not have any child artifact.

Note: Packaging of **Users** (that is, participants of type Users) are not supported.

Resolve Associated Workflows for Queue

You can package and deploy a queue with its associated workflows from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

When a queue (associated to a workflow) is packaged, the following options are displayed in the **Associated Workflow Details** section under the **Settings** window. The **Associated Workflow Details** section is displayed only in the **Install** and the **Update** tabs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Workflow Name	Displays the name of the workflow to which the queue is associated.
Source Version	Displays the version of workflow in the source environment to which the queue is associated.
Target Version	Enter the version of the workflow: • Specify the version of the workflow to which you want to associate the queue in the target environment. • If you do not specify the version of the workflow, then the latest version of the workflow is considered. **Note:** If the version of the workflow that is specified while installing the package is not present in the target environment, then the next minor version of the latest version of the workflow in the target environment is considered.

Resolve Custom Dependent Files for Queue

You can package Queue with custom dispatch pattern provider and the dependent files from the source environment to the target. The package will identify dispatch provider assembly registered in the SKAddInProviders table in the repository database.

The **Source Files** field displays the physical assembly path of the queue custom dispatch provider files in the source environment. Click **Add row** to add multiple rows to install additional files in the target environment. In the **Source Files** field, enter the source file path of the custom dependent files.

You must ensure the target repository database contains an entry for custom dispatch provider in the SKAddinProviders table, before you install a package containing the custom dispatch provider.

The **Install in Global Assembly Cache** field is optional and is used to install the custom dependent files in the Global Assembly Cache of the target environment. Check the **Install in Global Assembly Cache** box, if you want to install the files in the Global Assembly Cache of the target environment.

Reports

You can package and deploy the Report artifact along with its associated and child artifacts from one environment or repository to another environment or repository.

While installing the Report artifacts, it is possible that you can find n number of different reports with same title in the Reports list. The package module identifies a unique Report based on the **UniqueReportValue** column value of the repository database and installs it to the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Limitations of the Form Reports

The Form Reports artifact has some behavior limitations. Make sure that you adhere to the following guidelines while installing a Form Reports to a package:

- The Report you want to package must not contain any semicolon (;) in its name.
- Report does not exceed 10 MB of file size.

  You cannot import a package file size exceeding 10 MB to **Manage Package** list. Hence, you must not add any single report file or more than one report file that exceeds 10 MB file size in a single package. In such cases, we recommend to package each report file in a separate package template.

To add Reports to Package Template

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, point to **Settings**, point to **Reporting Service**, and then click **Reports**.

   The Reports page appears.
3. Select the report that you want to package.
4. Right-click the report, and then click **Add to Package Template**.

   The **Add to Package Template** dialog box appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Configuration Settings for Reports

Reports has an unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the **Settings** tab to define the configuration settings. The **Associated Artifacts** tab displays the associated artifacts of the **Reports** definition. The **Child Artifacts** tab displays child artifacts of the **Reports** definition.

Settings Tab

The properties in this tab allows you to take appropriate actions while installing, updating, or uninstalling the Package.

Install

The settings in this tab is referred to when installing the Package.

- **Report Exists**: This property allows you to take appropriate action when the Report that you are installing is already present in the target environment.
  - **Do not install Report**: Select this option if you want to use the existing report which is present in the target environment.
  - **Install to overwrite existing Report**: Select this option if you want to overwrite the Report which is present in the target environment with the one that is currently being packaged.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Source**: This property allows you to specify the source type of the Report or update the source type in the target environment if it is not present in the target environment.
- **Source**: This option displays the source type of the Report that you want to package.
- **Target**: This option allows you to modify or add the source type of the Report that you want to use in the target environment.

Update

The settings in this tab is referred to when the same Package is installed for the consecutive time.

- **Delete Report**: Select this option if you want to delete an existing Report which is present in the target environment.
  - **Yes**: Select this option to remove the Report in the target environment and the **Report Is Modified** option appears.

    Report Is Modified

    - **Retain**: Select this option if you want to retain the Report along with the modifications, in the target environment even after the Package is updated.
    - **Rollback to original state**: Select this option if you want to roll back to the original state. All modifications that were done to the Report after it was installed in the target environment will be reverted. However, the Report will be retained in the target environment.
    - **Delete**: Select this option if you want to delete the Report from the target environment when updating the Package.
  - **No**: Select this option if you want to retain the Report in the target environment and the **Report Exists** option appears.

    Report Exists

    - **Do not install Report**: Select this option if you want to use the existing Report which is present in the target environment. By default, the Do not install Report option is selected.
    - **Install to overwrite existing Report**: Select this option if you want to overwrite the existing Report which is present in the target environment with the one that is currently being packaged.

      Source

    - **Source**: Displays the source of the Report.
    - **Target**: Click to update the source of the Report in the Target environment.

Uninstall

This settings in this tab allows you to define the Package behavior when the Package is being uninstalled in the target environment.

- **Delete Report**: This property allows you to define whether you want to retain, rollback, or delete the Report in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to delete the Report from the target environment when the Package is uninstalled.
  - **No**: Select this option to retain the Report in the target environment even after the Package is uninstalled.
- **Report Is Modified**: Select this option to retain, rollback, or delete the Report in the target environment even after the Package is uninstalled. This property is displayed only if **Delete Report** is set to *Yes*. However, if

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Delete Report is set to *No*, this property is not displayed.

- **Retain**: Select this option to retain the Report along with the modifications, in the target environment even after the Package is uninstalled.
- **Roll back to original State**: Select this option to roll back to the original state. All modifications that were done to the Report after it was installed in the target environment will be reverted. However, the Report will be retained in the target environment.
- **Delete**: Select this option to delete the Report from the target environment when uninstalling the Package.

Associated Artifacts for Reports

This tab displays all the available associated artifacts of the selected Reports artifact.

Report can include the following associated artifacts:

- Parent Report
- Parent Folder: You cannot install a Folder which is disabled but it will allow you to install, update, or uninstall the Reports within it by validating the **UniqueReportValue**.

Child Artifacts for Reports

If a Report contains a child artifact, the Child Artifacts tab displays all the child artifacts of the selected Report artifact. Otherwise, the Child Artifacts tab is disabled.

Report can include the following Child Artifacts:

- Folder
- Child Reports

Include Child Item

For more information, see Include Child Item.

Security Artifacts for Reports

The Security Artifacts tab displays the security artifacts of a Report, if any.

You can enable the security settings of the reports that you want to include in the package.

For more information about including security artifacts, see Include Security Artifacts.

Note: When a report reference variable is assigned to report viewer artifacts then the report artifact will not be added automatically. The report artifact should be added manually.

Report Connections

You can package and deploy the Report Connection artifact from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To add Report Connections to Package Template

1. Log on to the **Enterprise Console**.
2. On the **Enterprise Console** menu, point to **Settings**, point to **Reporting Service**, and then click **Report Connections**. The **Report Connections** window appears.
3. Select the report connection that you want to package.
4. Right-click the report connection and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Runtime Initialization Settings

You can configure runtime initialization settings for Report Connections. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration Settings for Report Connections

The **Report Connections** has a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template > Edit**
- **Target Environment > Manage Package > Edit**

You can use the **Settings** tab to define the configuration settings. The **Associated Artifacts** and **Child Artifacts** tab are not available as **Report Connections** does not contain any associated or child artifacts.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Security Artifacts

Settings

The properties in this tab allow you to take appropriate actions while installing, updating, or uninstalling the Package.

Install

The settings in this tab are referred to when installing the Package.

- **Report Connection Exists**: This property allows you to take appropriate action when the Report Connection that you are installing is already present in the target environment:
  - **Do not install Report Connection**: Select this option if you want to use the report connection which is present in the target environment

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Install to overwrite existing Report Connection**: Select this option if you want to overwrite the report connection, which is present in the target environment with the one that is currently being packaged

- **Report Connection Details**:
  - **Source**: This section allows you to view the following connection details information that you have used in the source environment.
    - **URI**: Displays the URI of the report server.
    - **Authentication Type > Anonymous Access**: Displays whether the URI mentioned-above is configured anonymously. If this option is selected, it means that SQL Server Reporting Service is being accessed as anonymous user.
    - **User Name**: Displays the name of the user to access the report server URI.
    - **Password**: Displays the password in encrypted format. This password is used to authenticate the user credentials to access the report server.
    - **Domain**: Displays the name of the domain in which the above-mentioned user exists.
    - **Default Path**: Displays the default report template path. This information is required to locate a specific group of report templates that have been deployed in the reporting server. For example: /ArchestrA Reports/Historian.
  - **Target**: This section allows you to specify the following connection details information that you may use in the target environment.
    - **URI**: Enter the URI of the target report server.
    - **Authentication Type > Anonymous Access**: Specify whether the URI mentioned-above is configured anonymously. If this option is selected, it means that SQL Server Reporting Service is being accessed as anonymous user.
    - **User Name**: Enter the name of the user to access the report server URI.
    - **Password**: Enter the password. This password is used to authenticate the user credentials to access the report server.
    - **Domain**: Enter the name of the domain in which the above-mentioned user exists.
    - **Default Path**: Enter the default report template path. This information is required to locate a specific group of report templates that have been deployed in the reporting server. For example: /ArchestrA Reports/Historian.

      Note:

      You can specify the **Report Connection Details > Target** either during design-time or during installation. In case you do not specify the target details during design time, you can specify the target details during the install time in the **Report Connection Details > Target** section or else it will take the values specified in the **Report Connection Details > Source** section value during installation.

Update

The settings in this tab are referred to when the same Package is installed for the consecutive time.

- **Delete Report Connection**: Select this option if you want to delete an existing report connection which is present in the target environment.
  - **Yes**: Select this option to remove the report connection in the target environment. On selection of this option, the **Report Connection Is Modified** option appears as shown in the following table:

    Report Connection is Modified: Use this option to retain, delete, or rollback to original state the Report

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Connection in the target environment even after the package is updated. This property is applicable only if **Delete Report Connection** is set to *Yes* and the report connection is modified.

- **Retain**: Select this option to retain the Report Connection along with the modifications, in the target environment even after the package is updated.
- **Rollback to original state**: Select this option if you want to rollback to the original state. All modifications that were done to the Report Connection after it was installed in the target environment will be reverted. However, the Report Connection will be retained in the target environment.
- **Delete**: Select this option if you want to delete the Report Connection from the target environment when updating the package.

- **No**: Select this option if you want to retain the Report Connection in the target environment. On selection of this option, the **Report Connection Exists** option appears.

  Report Connection Exists: Use this option to take appropriate action when the Report Connection is present in the target environment.

  - **Do Not Install Report Connection**: Select this option if you want to use the existing Report Connection which is present in the target environment.
  - **Install to overwrite existing Report Connection**: Select this option if you want to overwrite the existing Report Connection which is present in the target environment with the one that is currently being packaged.

- **Report Connection Details**:
  - **Source**: This section allows you to view the following connection details information that you have used in the source environment.
    - **URI**: Displays the URI of the report server.
    - **Authentication Type > Anonymous Access**: Displays whether the URI mentioned-above is configured anonymously. If this option is selected, it means that SQL Server Reporting Service is being accessed as anonymous user.
    - **User Name**: Displays the name of the user to access the report server URI.
    - **Password**: Displays the password in encrypted format. This password is used to authenticate the user credentials to access the report server.
    - **Domain**: Displays the name of the domain in which the above-mentioned user exists.
    - **Default Path**: Displays the default report template path. This information is required to locate a specific group of report templates that have been deployed in the reporting server. For example: /ArchestrA Reports/Historian.
  - **Target**: This section allows you to specify the following connection details information that you may use in the target environment.
    - **URI**: Enter the URI of the target report server.
    - **Authentication Type > Anonymous Access**: Specify whether the URI mentioned-above is configured anonymously. If this option is selected, it means that SQL Server Reporting Service is being accessed as anonymous user.
    - **User Name**: Enter the name of the user to access the report server URI.
    - **Password**: Enter the password. This password is used to authenticate the user credentials to access the report server.
    - **Domain**: Enter the name of the domain in which the above-mentioned user exists.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Default Path**: Enter the default report template path. This information is required to locate a specific group of report templates that have been deployed in the reporting server. For example: /ArchestrA Reports/Historian.

Uninstall: The settings in this tab allow you to define the Package behavior when the Package is being uninstalled in the target environment.

- **Delete Report Connection**: This property allows you to define whether you want to retain or delete the report connection in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option if you want to delete the report connection from the target environment when the Package is uninstalled.
  - **No**: Select this option if you want to retain the report connection in the target environment even after the Package is uninstalled.
- **Report Connection Is Modified**: This option allows you to decide if you want to retain, roll back to original state or delete the report connection in the target environment even after the Package is uninstalled. This property is displayed only if **Delete Report Connection** is set to *Yes*. However, if **Delete Report Connection** is set to *No*, this property is not displayed.
  - **Retain**: Select this option if you want to retain the report connection along with the modifications in the target environment even after the Package is uninstalled.
  - **Roll back to original State**: Select this option if you want to roll back to the original state. All modifications that were done to the connection after it was installed in the target environment will be reverted. However, the report connection will be retained in the target environment.
  - **Delete**: Select this option if you want to delete the report connection from the target environment when uninstalling the Package.

Associated Artifacts

Report Connection does not have any associated artifacts.

Child Artifacts

Report Connection does not have any child artifacts.

Security Artifacts

The Security Artifacts tab displays the security artifacts of report connections, if any.

You can enable the security settings of the report connections that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

Report Templates

You can package and deploy the Report Templates artifact from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add Report Templates to Package Template

Complete the following tasks to add a report template to a package template:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, point to **Settings**, point to **Reporting Service**, and then click **Report Templates**. The **Report Templates** window appears.
3. Select the report template that you want to package.
4. Right-click the report template and then click **Add to Package Template**. The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Runtime Initialization Settings

You can configure runtime initialization settings for Report Templates. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration settings for Report Templates

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template > Edit**
- **Target Environment > Manage Package > Edit**

Settings

You can use the **Settings** tab to define the configuration settings. Set the properties in this tab to take appropriate actions while installing, updating, or uninstalling the package.

Install

Use this tab to set the configurations for installing the package.

- **Report Template Exists**: This property allows you to take appropriate action when the Report Templates that you are installing is already present in the target environment:
  - **Do Not Install Report Template**: Select this option if you want to use the report template which is present in the target environment.
  - **Install to overwrite existing Report Template**: Select this option if you want to overwrite the report template which is present in the target environment with the one that is currently being packaged.
- **Associated Artifacts Not Present**: This property allows you to take appropriate action when the associated artifacts of the Report Template is not present in the target environment.
  - **Install Report Template**: Select this option if you want to install the Report Template, even if the associated artifacts are not present in the target environment.
  - **Do Not Install Report Template**: Select this option if you do not want to install the Report Template as its associated artifacts are not present.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Update

Use the settings in this tab to update an installed package.

- **Delete Report Template**: Select this option if you want to delete the Report Template which exists in the target environment.
  - **Yes**: Select this option to remove the Report Template in the target environment. On selection of this option, the **Report Template Is Modified** option appears.

    Report Template is Modified: Use this option to retain, delete, or rollback to original state the Report Template in the target environment even after the package is updated. This property is applicable only if **Delete Report Template** is set to *Yes* and the report template is modified.
    - **Retain**: Select this option to retain the Report Template along with the modifications, in the target environment even after the package is updated.
    - **Rollback to original state**: Select this option if you want to rollback to the original state. All modifications that were done to the Report Template after it was installed in the target environment will be reverted. However, the Report Template will be retained in the target environment.
    - **Delete**: Select this option if you want to delete the Report Template from the target environment when updating the package.
  - **No**: Select this option if you want to retain the Report Template in the target environment. On selection of this option, the **Report Template Exists** and the **Associated Artifacts not present** options appear.

    Report Template Exists: Use this option to take appropriate action when the Report Template is present in the target environment.
    - **Do Not Install Report Template**: Select this option if you want to use the existing Report Template which is present in the target environment.
    - **Install to overwrite existing Report Template**: Select this option if you want to overwrite the existing Report Template which is present in the target environment with the one that is currently being packaged.

  Associated Artifacts not present: Use this property to take appropriate action when the associated artifacts of the Report Template are not present in the target environment.
- **Install Report Template**: Select this option if you want to update the Report Template.
- **Do Not Install Report Template**: Select this option if you do not want to update the Report Template as associated artifacts are not present.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete Report Template**: This property allows you to define whether you want to retain or delete the Report Template in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option if you want to delete the Report Template from the target environment when the Package is uninstalled.
  - **No**: Select this option if you want to retain the Report Template in the target environment even after the Package is uninstalled.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Report Template Is Modified**: This option allows you to decide if you want to retain, roll back to original state or delete the Report Template in the target environment even after the Package is uninstalled. This property is displayed only if **Delete Report Template** is set to *Yes*. However, if **Delete Report Template** is set to *No*, this property is not displayed.
  - **Retain**: Select this option if you want to retain the Report Template along with the modifications in the target environment even after the Package is uninstalled.
  - **Roll back to original State**: Select this option if you want to roll back to the original state. All modifications that were done to the template after it was installed in the target environment will be reverted. However, the Report Template will be retained in the target environment.
  - **Delete**: Select this option if you want to delete the Report Template from the target environment when uninstalling the Package.

Associated Artifacts

This tab displays all the associated artifacts of the selected Report Template artifact.

Report Templates can include "Report Connection" as an associated artifact.

Note: When you select the **Include Associated Artifacts** option as **Yes** while adding the Report Template to a Package Template, the associated artifacts are added to the package template and it also allows you to view the associated artifacts of a Report Template in the tree view of a Package. If you select the **Include Associated Artifacts** option as **No**, the associated artifacts are not added to the package template and are also not visible in the tree view of a Package.

Child Artifacts

This tab is not available as the Report Template artifact does not have any child artifacts.

Security Artifacts

The **Security Artifacts** tab displays the security artifacts of report templates, if any.

You can enable the security settings of the report templates that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

Resolve Report Template Path

The report template path is required to locate a specific group of report templates that have been deployed in the reporting server. You can edit the path of the template file in the target environment.

For example: */ArchestrA Reports/Historian*.

The **Report template path** grid contains the following settings, which are required to resolve the report template path:

Field Name	Description
Source	Displays the report template path value in the source environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Target	Enter the report template path value in the target environment. If you do not specify any value, then the value specified in the source is considered. Ensure to provide a valid report template path that contains the valid files.

Security Group

You can package and deploy a Security Group definition and its associated artifacts from one environment or repository to another environment or repository using the package feature in AVEVA Work Tasks.

You can package the security group from the following three levels:

- Repository level
- List level
- List item level

Repository Level

In repository level, you can only overwrite the standard security group. You cannot create or delete the standard security group.

You can create, overwrite, or delete a custom security group at the repository level. Deleting a custom security group at repository level also deletes the security group at all lists and list items level.

List Level

At list level, you cannot create or delete security groups. You can only overwrite the security groups at list level.

- Security Group Customization - At list level, security groups are not available by default. Enable the security group customization first, which makes a copy of all the security groups and makes available at list level. After you enable the security group customization, you can overwrite the security groups at list level.
- Remove Security Group Customization - If there is no need of a separate definition for any of the security groups at list level, then you can remove the security group customization. That is, the definition of all the security groups are removed from the list level, but the security group exists at repository level, which is effective in the current list as well.

List Item Level

At List Item Level, you can select and package individual security group. When installing or updating the package in the target environment, the package checks whether the security group customization is enabled at the list item level.

Notes:

- You cannot create or delete a security group at List Level or List Item level. Thus, the Resource Mapping is identified option is not available in both the List Level and List Item Level Security Group Configuration Settings.
- You cannot package a standard Security Right as an associated artifact.
- Web part page will not be packaged with the security group as an associated artifact. It will just provide a reference to the web part page in the target environment. After the security group is installed, if the web

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

part page is present in the target environment then the security group is linked to it.

- A list item from a dynamic list, such as Employee list can be edited or viewed using different forms or form versions based on the security rights given to the logged in user.
- For an existing Security Group, you can get a backup of the artifact that is available in the target environment.

Security Group - Repository Level

You can package and deploy the repository level Security Group definition along with its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add Repository Level Security Group to Package Template

Complete the following tasks to add security group definition to package template:

1. Log on to the **Enterprise Console**.
2. Click **Repository Settings**.
   The **Repository Settings** window appears.
3. On the **Advanced Settings** tab, click **Manage Security Groups**.
4. Right-click the Security Group that you want to package, and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

Associated Artifacts for Security Group Definition

The Security Group at repository level can have the following associated artifacts:

- Custom Security Right
- Workflow
- Form Link

Child Artifacts for Security Group Definition

The Security Group at repository level does not have any child artifact.

Configuration Settings for Security Group

The Manage Security Group definition has an unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the Settings tab of the Manage Security Group definition artifact to define the configuration settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Associated Artifacts tab displays the associated artifacts of the Manage Security Group definition.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Child Artifacts

Settings tab

You can use the properties in the Settings tab to take appropriate actions to install, update, or uninstall the package.

Install

You can refer to the configuration settings in the **Install** tab to install the package.

- **Security Group Exists**: Select this property to take appropriate action when the Security Group definition specified in the package is already present in the target environment.
  - **Do not install Security Group**: Select this option to use the Security Group definition that is present in the target environment.
  - **Install to overwrite existing Security Group**: Select this option to overwrite the Security Group definition that is present in the target environment with the one in the package.

    By default, Install to overwrite existing Security Group is selected, since the security group will be available in the target environment and on installing the package it will be overwritten. If the security group is not available in the target environment then prior to installing the package the security group will be enabled, hence the security group will be overwritten.
- **Associated Artifacts Not Present**: This property allows you to take appropriate action when the associated artifacts of the Security Group is not present in the target environment.
  - **Install Security Group**: Select this option to install the Security Group, even if the associated artifacts are not present in the target environment.
  - **Do not install Security Group**: Select this option if you do not want to install the Security Group artifact as its associated artifacts are not present.

Update

You can refer to the settings in this tab to install the same package for the consecutive time.

Delete Security Group: Select this option to delete the Security Group definition that is present in the target environment.

- **Yes**: Select Yes to remove the Security Group definition in the target environment. When you select Yes, the **Security Group is Modified** and **Resource Mapping is Identified** options appear.

  Security Group is Modified

  - **Retain**: Select this option to retain the Security Group definition along with the modifications, in the target environment even after the Package is updated.
  - **Rollback to Original State**: Select this option to roll back to the original state. All modifications that were

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

done to the Security Group definition after it was installed in the target environment will be reverted. However, the Security Group definition will be retained in the target environment.

- **Delete**: Select this option to delete the Security Group definition from the target environment when updating the Package.

Resource Mapping is Identified

- **Retain**: Select this option to retain the Resource Mapping definition in the target environment when updating the Package.
- **Delete**: Select this option to delete the Resource Mapping definition from the target environment when updating the Package.

- **No**: Select No if you want to retain the Security Group definition in the target environment.
When you select No, the **Security Group Exists** option appears.

  Security Group Exists

  - **Do not install Security Group**: Select this option to use the Security Group definition that is present in the target environment.
  - **Install to overwrite existing Security Group**: Select this option to overwrite the Security Group that is present in the target environment with the one present in the package.

  Associated Artifacts not present

  - **Update Security Group**: Select this option to update the Security Group even though no associated artifacts are present.
  - **Do not update Security Group**: Select this option if you do not want to update the Security Group when the associated artifacts are not present.

Uninstall

You can refer to the following settings in the Uninstall tab to uninstall a package from a target environment:

- **Delete Security Group**: Select this property to retain or delete the Security Group definition in the target environment even after the Package is uninstalled.
  - **Yes**: Select Yes to remove the Security Group definition from the target environment when the Package is uninstalled. When you select Yes, the **Security Group is Modified** and **Resource Mapping is Identified** options appear:

    Security Group is Modified

    - **Retain**: Select this option to retain the Security Group definition along with the modifications, in the target environment even after the Package is uninstalled.
    - **Rollback to Original State**: Select this option to roll back to the original state. All modifications that were done to the Security Group definition after it was installed in the target environment will be reverted. However, the Security Group definition will be retained in the target environment.
    - **Delete**: Select this option to delete the Security Group definition from the target environment when uninstalling the Package.

      Resource Mapping is Identified

    - **Retain**: Select this option to retain the Resource Mapping definition in the target environment when uninstalling the Package.
    - **Delete**: Select this option to delete the Resource Mapping definition from the target environment when uninstalling the Package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **No**: Select No to retain the Security Group definition in the target environment even after the Package is uninstalled.

When you select Yes in the Delete Security Group option while doing the Update or Uninstall, the following scenarios are possible:

Installation Type	Security Group is Modified Status	Security Group is Modified Option	Resource Mapping is Identified	Result
New Security Group	If Security Group is not modified.	Select the Retain option.	If no Resource Mapping is identified. Or If you select Delete option for Resource Mapping is Identified.	Security Group artifact is deleted along with the Resource Mapping from the target environment.
New Security Group	If Security Group is modified.	Select the Retain option.	If the Resource Mapping is identified and you select Retain option for Resource Mapping is Identified.	Security Group artifact is retained along with the Resource Mapping in the target environment.
New Security Group	If Security Group is modified.	Select the Retain option.	Resource Mapping is either identified or not identified.	Security Group artifact is retained in the target environment.
New Security Group	If Security Group is not modified.	Select the Rollback to Original State option.	No Resource Mapping is identified. Or If you select Delete option for Resource Mapping is Identified.	Security Group artifact is deleted along with the Resource Mapping from the target environment.
New Security Group	In case, the Security Group is modified.	Select the Rollback to Original State option.	In case, either the Resource Mapping is identified or not.	Security Group artifact will not be rolled back as there is no backup available.
New Security Group	In case, either the Security Group is	Select the Rollback to Original	In case, the Resource Mapping is	Security Group artifact is retained

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation Type	Security Group is Modified Status	Security Group is Modified Option	Resource Mapping is Identified	Result
	modified or not.	State option	identified and user has selected "Retain" option for "Resource Mapping is Identified".	along with the Resource Mapping in the target environment.
New Security Group	In case, Security Group is not modified.	Select the Delete option	In case, no Resource Mapping is identified. Or User has selected Delete option for "Resource Mapping is Identified".	Security Group artifact is deleted along with the Resource Mapping from the target environment.
New Security Group	In case, either the Security Group is modified or not.	Select the Delete option	In case, the Resource Mapping is identified and user has selected "Retain" option for "Resource Mapping is Identified".	Security Group artifact is retained along with the Resource Mapping in the target environment.
Existing Security Group	In case, either the Security Group is modified or not.	Select the Retain option	In case, either the Resource Mapping is identified or not.	Security Group artifact is retained in the target environment.
Existing Security Group	In case, either the Security Group is modified or not.	Select the Rollback to Original State option	In case, either the Resource Mapping is identified or not.	Security Group artifact is Rolled back to its Original state in the target environment.
Existing Security Group	In case, either the Security Group is modified or not.	Select the Delete option	In case, the Resource Mapping is identified and user has selected "Retain" option for "Resource Mapping is Identified".	Security Group artifact is retained along with the Resource Mapping in the target environment.
Existing Security Group	In case, either the Security Group is modified or not.	Select the Delete option	In case, no Resource Mapping is identified. Or	Security Group artifact is Rolled back to its Original state in the target

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Installation Type	Security Group is Modified Status	Security Group is Modified Option	Resource Mapping is Identified	Result
			User has selected Delete option for "Resource Mapping is Identified".	environment.

Associated Artifacts Tab

The Associated Artifacts tab displays all the associated artifacts of the selected Security Group definition.

If a form link with the same name is present in the target environment and it is not overwritten, then the security right might not work as expected if the form link in the target environment is of a different type and different mode.

Child Artifacts Tab

The Security Group at repository level does not have any child artifact.

Security Group - List Level

You can package and deploy the list level Security Group definition and its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add List Level Security Group to Package Template

Complete the following tasks to add security group definition to package template:

1. Go to the List which contains the security group that you want to package. For example, Workflow List.
2. Click **Security** on the ribbon bar.
   The security settings window opens.
3. Click **Advanced Settings** tab.
4. Click **Customize Security Group** for this List.
   The security group list at list level window appears.
5. Select the **Security Group** that you want to package.
6. Right-click and then select **Add to Package Template**.
   The **Add to Package Template** dialog box appears with the List Level Security Group.
   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

The following shows the tree view format of the list level security group, for example, **Add to Package Template**: <**Security Group Name - List - List name**>:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Associated Artifacts for Security Group Definition

A List Level Security Group can have the following Associated Artifacts:

- Repository Level Security Group
- List Specific Custom Security Right
- Workflow
- Form Link
- Dynamic Lists
- Form

Child Artifacts for Security Group Definition

Security Group does not have any child artifacts.

Configuration Settings for Security Group

The Manage Security Group definition has an unique set of configuration settings required for installing, updating, and uninstalling the Package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the Settings tab of the **Manage Security Group** definition artifact to define the configuration settings. The **Associated Artifacts** tab displays the associated artifacts of the **Manage Security Group** definition.

The Package Template configuration window contains the following tabs:

- Settings
- Associated Artifacts
- Child Artifacts

Settings Tab

You can use the properties in the Settings tab to take appropriate actions to install, update, or uninstall the package.

Install:

While installing the security group, the package checks whether the security group customization is enabled the list level. If the security group customization is not enabled, then the package first enables the security group customization. Then the package overwrites the security group definition. If the security group customization is already enabled, then the package overwrites the security group.

The configuration settings in the Install tab is referred to when installing the package.

- **Security Group Exists**: This property allows you to take appropriate action when the Security Group definition specified in the package is already present in the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Do not install Security Group**: Select this option to use the Security Group definition that is present in the target environment.
- **Install to overwrite existing Security Group**: Select this option to overwrite the Security Group definition that is present in the target environment with the one in the package.

  By default, Install to overwrite existing Security Group is selected, since the security group will be available in the target environment and on installing the package it will be overwritten. If the security group is not available in the target environment then prior to installing the package the security group will be enabled, hence the security group will be overwritten.

- **Associated Artifacts Not Present**: This property allows you to take appropriate action when the associated artifacts of the Security Group is not present in the target environment.
  - **Install Security Group**: Select this option to install the Security Group, even if the associated artifacts are not present in the target environment.
  - **Do not install Security Group**: Select this option if you do not want to install the Security Group artifact as its associated artifacts are not present.

Update

The settings in this tab is referred to when the same Package is installed for the consecutive time.

- **Delete Security Group**: Select this option to delete the Security Group definition that is present in the target environment.
  - **Yes**: Select this option to remove the Security Group definition in the target environment. On selection of this option, the Security Group is Modified option appears.

    Security Group is Modified

    - **Retain**: Select this option to retain the Security Group definition along with the modifications, in the target environment even after the Package is updated.
    - **Rollback to Original State**: Select this option to roll back to the original state. All modifications that were done to the Security Group definition after it was installed in the target environment will be reverted. However, the Security Group definition will be retained in the target environment.
    - **Delete**: Select this option to delete the Security Group definition from the target environment when updating the Package.
  - **No**: Select this option to retain the Security Group definition in the target environment. On selection of this option, the Security Group Exists option appears as shown in the following table:

    Security Group Exists

    - **Do not install Security Group**: Select this option to use the Security Group definition that is present in the target environment.
    - **Install to overwrite existing Security Group**: Select this option to overwrite the Security Group that is present in the target environment with the one present in the package.

      Associated Artifacts not present

    - **Update Security Group**: Select this option to update the Security Group even though no associated artifacts are present.
    - **Do not update Security Group**: Select this option if you do not want to update the Security Group when the associated artifacts are not present.

Uninstall

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

You can refer to the following settings in the Uninstall tab to uninstall a package from a target environment:

- **Delete Security Group**: Select this property to retain or delete the Security Group definition in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option to remove the Security Group definition from the target environment when the Package is uninstalled.

    When you select Yes, the Security Group is Modified option appears as shown in the following table:

    Security Group is Modified

    - **Retain**: Select this option to retain the Security Group definition along with the modifications in the target environment even after the Package is uninstalled.
    - **Rollback to Original State**: Select this option to roll back the Security Group to the original state. All modifications that were done to the Security Group definition after it was installed in the target environment will be reverted. However, the Security Group definition will be retained in the target environment.
    - **Delete**: Select this option to delete the Security Group definition from the target environment while uninstalling the package.
  - **No**: Select this option if you want to retain the Security Group definition in the target environment even after the Package is uninstalled.

Remove Security Group Customization at List Level

While uninstalling the security group at list level, the security group artifact checks whether the security group customization is enabled by the current security group artifact.

- If not, then uninstall rolls back the security group definition to the original state, but does not remove the security group customization as it was not enabled by it.
- Otherwise, the security group artifact checks whether any other security group is modified at the list level.
  - If yes, then uninstall does not remove the security group customization, but rolls back the security group definition to the original state.
  - Otherwise, uninstall removes the security group customization.
- Consider that a custom security group is packaged at list level and a repository level security group is an associated artifact. While installing, both the security groups are installed. While uninstalling, select the "Delete" option for the repository level security group and "Retain" option for the list level security group. Then, the list level security group artifact is uninstalled or retained first, followed by the repository level security group artifact which is deleted while uninstalling. As the repository level security group is deleted, it removes the complete definition of the security group from all the levels.
  - Thus, the list level security group gets uninstalled, though the delete option is to "Retain" the security group. However, the security group customization is retained.
  - The security group customization cannot be removed by the package, even if you uninstall the package again. So it is advisable to select the "Delete" option for list level security group. If you select the "Delete" option for repository level security group, irrespective of the option that is selected for list level security group, the security group as a whole is deleted.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Associated Artifacts

This tab displays all the associated artifacts of the selected Security Group definition.

If a Form link with the same name is already present in the target environment and it is not overwritten, then the security right might not work as expected if the form link in the target environment is of a different type and different mode, and if the mode is Pop-Up.

If the Security Group at repository level is not available in the target environment, then the security group at list level will not be installed.

Child Artifacts

Security Group does not have any child artifacts.

Security Artifacts

The Security Artifacts tab displays the security artifacts of a database connection, if any.

Security Group - List Item Level

You can package and deploy the list item level Security Group definition and its associated artifacts from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add List Item Level Security Group to Package Template

Complete the following tasks to add list item level security group definition to package template:

1. Go to the List Item which contains the security group that you want to package. For example, Workflow List.
2. Right-click a list item, and then click **Security Settings** in the context-menu.
   The security settings window opens.
3. Click **Advanced Settings** tab.
4. Click **Customize Security Group** for this List Item.
   The security group list at list item level window appears.
5. Right-click the Security Group that you want to package, and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears with the List Item Level Security Group.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

The following shows the tree view format of the list item level security group, for example, **Add to Package Template**: <**Security Group Name - List Item - List Name/List Item Name**>:

Associated Artifacts for Security Group Definition

A List Item Level Security Group can have the following Associated Artifacts:

- Repository Level Security Group
- List Specific Custom Security Right
- Workflow

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Form Link
- Dynamic Lists
- Form
- List Item

For information about whether a list item can be packaged or not, see Packageable Artifacts.

If the Security Group at repository level is not available in the target environment, then the security group at list item level will not be installed.

In a dynamic list, you cannot package a list item, but you can package security group of the list item. Thus, when you package the security group of the list item, the dynamic list is identified as the associated artifact of the security group. The list item, however, does not become an associated artifact of the security group. It is recommended that you package the dynamic list and the security group of the list item in two different packages. In this case, when you install the package in target environment, install the package that contains the dynamic list first. Then manually create the list item of the dynamic list, and run the package that contains the security group of the list item.

On the other hand, if you package both the dynamic list and the security group of the dynamic list item in one package, you have to install the package twice. When you install the package for the first time, the dynamic list gets installed, while it cannot install the security group as the list item is not available in the dynamic list. Create the required list item in the dynamic list manually and install the package again.

Child Artifacts for Security Group Definition

A List Item Level Security Group does not have any child artifact.

Security Artifacts for Security Group Definition

You can enable the security settings of the Security Group that you want to include in the package.
For more information about including security artifacts, see Include Security Artifacts.

Configuration Settings for Security Group

The Manage Security Group definition has an unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the Settings tab of the Manage Security Group definition artifact to define the configuration settings. The Associated Artifacts tab displays the associated artifacts of the Manage Security Group definition.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Uninstall
- Associated Artifacts
- Child Artifacts

Settings Tab

You can use the properties in the Settings tab to take appropriate actions to install, update, or uninstall a package.

Install

During List Item Level security group installation, the package checks whether the security group customization is enabled at the list item level. If the security group customization is not enabled, then the package first enables the security group customization. Then the package overwrites the security group definition. If the security group customization is already enabled, then the package overwrites the security group.

The configuration settings in the **Install** tab is referred to when installing the package.

- **Security Group Exists**: Select this property to take appropriate action when the Security Group definition specified in the package is already present in the target environment.
  - **Do not install Security Group**: Select this option to use the Security Group definition that is present in the target environment.
  - **Install to overwrite existing Security Group**: Select this option to overwrite the Security Group definition that is present in the target environment with the one in the package.

    By default, Install to overwrite existing Security Group is selected, since the security group will be available in the target environment and on installing the package it will be overwritten. If the security group is not available in the target environment then prior to installing the package the security group will be enabled, hence the security group will be overwritten.
- **Associated Artifacts Not Present**: Select this property to take appropriate action when the associated artifacts of the Security Group is not present in the target environment.
  - **Install Security Group**: Select this option to install the Security Group, even if the associated artifacts are not present in the target environment.
  - **Do not install Security Group**: Select this option if you do not want to install the Security Group artifact as its associated artifacts are not present.

Update

The settings in this tab is referred to when the same package is installed for the consecutive time.

- **Delete Security Group**: Select this option to delete the Security Group definition that is present in the target environment.
  - **Yes**: Select this option to remove the Security Group definition in the target environment. On selection of this option, the Security Group is Modified option appears:

    Security Group is Modified

    - **Retain**: Select this option to retain the Security Group definition along with the modifications, in the target environment even after the Package is updated.
    - **Rollback to Original State**: Select this option to roll back to the original state. All modifications that were done to the Security Group definition after it was installed in the target environment will be reverted. However, the Security Group definition will be retained in the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Delete**: Select this option to delete the Security Group definition from the target environment when updating the Package.

- **No**: Select this option to retain the Security Group definition in the target environment. On selection of this option, the **Security Group Exists** option appears.

  Security Group Exists

  - **Do not install Security Group**: Select this option to use the Security Group definition that is present in the target environment.
  - **Install to overwrite existing Security Group**: Select this option to overwrite the Security Group definition that is present in the target environment with the one present in the package.

    Associated Artifacts not present

  - **Update Security Group**: Select this option to update the Security Group definition even though no associated artifacts are present.
  - **Do not update Security Group**: Select this option if you do not want to update the Security Group definition when the associated artifacts are not present.

Uninstall

You can refer to the following settings in the Uninstall tab to uninstall a package from a target environment:

- **Delete Security Group**: This property allows you to decide if you want to retain or delete the Security Group definition in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option if you want to remove the Security Group definition from the target environment when the Package is uninstalled.

    When you select **Yes**, the **Security Group is Modified** option appears.

    Security Group is Modified

    - **Retain**: Select this option to retain the Security Group definition along with the modifications in the target environment even after the Package is uninstalled.
    - **Rollback to Original State**: Select this option to roll back the Security Group to the original state. All modifications that were done to the Security Group definition after it was installed in the target environment will be reverted. However, the Security Group definition will be retained in the target environment.
    - **Delete**: Select this option to delete the Security Group definition from the target environment while uninstalling the package.
  - **No**: Select this option to retain the Security Group definition in the target environment even after the Package is uninstalled.

Remove Security Group Customization at List Item Level

While uninstalling the security group at list item level, the security group artifact checks whether the security group customization is enabled by the current security group artifact.

- If not, then uninstall rolls back the security group definition to the original state, but does not remove the security group customization as it was not enabled by it.
- Otherwise, the security group artifact checks whether any other security group is modified at the list item level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

    - If yes, then uninstall does not remove the security group customization, but rolls back the security group definition to the original state.
    - Otherwise, uninstall removes the security group customization.
- Consider that a custom security group is packaged at list item level and a repository level security group is an associated artifact. While installing, both the security groups are installed. While uninstalling, select the "Delete" option for the repository level security group and "Retain" option for the list item level security group. Then, the list item level security group artifact is uninstalled or retained first, followed by the repository level security group artifact which is deleted while uninstalling. As the repository level security group is deleted, it removes the complete definition of the security group from all the levels.
    - Thus, the list item level security group gets uninstalled, though the delete option is to "Retain" the security group. However, the security group customization is retained.
    - The security group customization cannot be removed by the package, even if you uninstall the package again. So it is advisable to select the "Delete" option for list item level security group. If you select the "Delete" option for repository level security group, irrespective of the option that is selected for list item level security group, the security group as a whole is deleted.

You can package the list item level security group from the following lists:

- Dynamic list with default field
- Dynamic list with unique field
- Reports with duplicate names
- Manage ListViews
- Security Rights with different scopes

Associated Artifacts tab

The Associated Artifacts tab displays all the associated artifacts of the selected Security Group definition.

Child Artifacts tab

The Security Group at List Item level does not have any child artifact.

Security Artifacts

The Security Artifacts tab displays the security artifacts of a database connection, if any.

Security Group with Forms as Associated Artifacts

When you package a Security Group which is associated to a form when the options **Can Edit** or **Can View Item Details** selected, the associated form gets packaged.

In the **Mapping of Security Right with Form** grid, you must specify the version of the form in target environment to which you want to map the security right.

- If you do not edit the Target version field and retain the default value as "Latest", then the latest version of the form will be mapped to security group in the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- If you do not specify the target version, the security group will be mapped to the source version of the form in the target environment. If the source version of the form is not present, then the latest version of the form will be mapped to the security group.
- If the target version of the form is not present in the target environment, then the latest version of the form will be mapped to the security group in the target environment.

Security Group with Workflow as Associated Artifacts

When you package a Security Group which is associated to a workflow when the options **requires approval** or **requires approval with prompt** selected, the associated workflow also gets packaged.

In the **Mapping of Security Right with Workflow** grid, you must specify the version of the workflow in target environment to which you want to map the security right.

- If you do not edit the **Target version** field and retain the default value as "Latest", then the latest version of the workflow will be mapped to security group in the target environment.
- If you do not specify the target version, the security group will be mapped to the source version of the workflow in the target environment. If the source version of the workflow is not present, then the latest version of the workflow will be mapped to the security group.
- If the target version of the workflow is not present in the target environment, then the latest version of the workflow will be mapped to the security group in the target environment.

Security Right

You can package and deploy the Security Right definition along with its associated and child artifact from one environment or repository to another environment or repository using the Package feature in AVEVA Work Tasks.

To add Security Right to Package Template

To add security right definition to package template:

1. Navigate to the **Repository Settings** > **Advance Settings** > **Manage Security Rights**.
2. Select the Security Rights that you want to package.
3. Right-click and then select **Add to Package Template** from the shortcut menu. The **Add to Package Template** screen appears. For more information on adding artifacts to Package Template, see **Adding Artifacts to a Package Template**.

   Note: You cannot add standard Security Rights to the Package Template.

Associated Artifacts for Security Right definition

Security Right can have the following Associated Artifacts:

- Dynamic Lists
- Parent Custom Security Right

  Note: If Parent Security Right is a standard Security Right then it cannot be packaged as an associated artifact.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Child Artifacts for Security Right definition

Security Right can have the following Child Artifacts:

- Custom Security Right
- Configuration settings for Security Right

The Manage Security Rights definition has an unique set of configuration settings required for installing, updating, and uninstalling the Package. The configuration settings can be edited and viewed from the **Source Environment** > **Package Template** screen or **Target Environment** > **Manage Package** screen. The **Settings** tab of the Manage Security Rights definition artifact allows you to define the configuration settings. The Associated and Child Artifacts tab displays the associated and child artifacts of the Manage Security Rights definition respectively.

- Settings
- Associated Artifacts
- Child Artifacts

Settings tab: The properties in this tab allows you to take appropriate actions while installing, updating, or uninstalling the Package.

Install: The settings in this tab is referred to when installing the Package.

- **Security Rights Exists**: This property allows you to take appropriate action when the Security Rights definition specified in the package is already present in the target environment.
  - **Do not install Security Right**: Select this option if you want to use the Security Right definition that is present in the target environment.
  - **Install to overwrite existing Security Right**: Select this option if you want to overwrite the Security Right definition that is present in the target environment with the one in the package.
- **Associated Artifacts Not Present**: This property allows you to take appropriate action when the associated artifacts of the Security Right is not present in the target environment.
  - **Install Security Right**: Select this option if you want to install the Security Rights artifact, even if the associated artifacts are not present in the target environment.
  - **Do not install Security Right**: Select this option if you do not want to install the Security Rights artifact as its associated artifacts are not present.
- **Update**: The settings in this tab is referred to when the same Package is installed for the consecutive time.
- **Delete Security Right**: Select this option if you want to delete the Security Right definition that is present in the target environment.
  - **Yes**: Select this option to remove the Security Right definition in the target environment. On selection of this option, the Security Right Is Modified option appears.

    Security Right is Modified

    - **Retain**: Select this option to retain the Security Right definition along with the modifications, in the target environment even after the Package is updated.
    - **Rollback**: Select this option to roll back to the original state. All modifications that were done to the Security Right definition after it was installed in the target environment will be reverted. However, the Security Right definition will be retained in the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Delete**: Select this option to delete the Security Right definition from the target environment when updating the Package.

- **No**: Select this option to retain the Security Right definition in the target environment. On selection of this option, the Security Right Exists option appears.

  Security Right Exists

  - **Do not install Security Right**: Select this option to use the Security Right definition that is present in the target environment.
  - **Install to overwrite existing Security Right**: Select this option to overwrite the Security Right that is present in the target environment with the one present in the package.

    Associated Artifacts not present

  - **Update Security Right**: Select this option to update the Security Rights even though no associated artifacts are present.
  - **Do not update Security Right**: Select this option if you do not want to update the Security Right artifact as associated artifacts are not present.

    Note: If the Security Right exists in target repository while installing or updating the package then you cannot overwrite the security right definition if any one of the following items values are modified:

- Title
- Right Code
- Scope of the Right
- Value for the Right
- Follower-specific value for the Right
- Is this Right applicable to its Follower items?

Changing the type of any of the above-mentioned rights may result in any of the following:

1. Loss of data
2. Compromise of security
3. Enforcement of wrong privileges.

For example: Consider a right called "CanDoTask" is of "AllowDisallow,RequiresApproval" type of right, and the "Security Workflow" is selected for a particular security group (Say xyz), then when a user, who belongs to the "xyz" security group, logs on and tries to perform the particular task (CanDoTask) then the "Security Workflow" will be triggered and it will be available in approver's (say manager's) Inbox. Once the approver approves the Work Item the particular task will be done based on its approval; if the approver rejects then the task will not be allowed to be completed as it is rejected by the approver.

In this scenario, if you try to modify the type of "CanDoTask" right to "AllowDisallow" then the privilege and grant or deny the right to the users is enforced, which is not right.

Uninstall: This settings in this tab is referred to when the Package is being uninstalled in the target environment.

- **Delete Security Right**: This property allows you to decide if you want to retain or delete the Security Right definition in the target environment even after the Package is uninstalled.
  - **Yes**: Select this option if you want to remove the Security Right definition from the target environment

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

when the Package is uninstalled.

- **No**: Select this option if you want to retain the Security Right definition in the target environment even after the Package is uninstalled.

- **Security Right is Modified**: This option allows you to decide if you want to retain, roll back to original state or delete the Security Right definition in the target environment even after the Package is uninstalled. This property is displayed only if **Delete Security Right** is set to *Yes*. However, if **Delete Security Right** is set to *No*, this property is not displayed.
  - **Retain**: Select this option if you want to retain the Security Right definition along with the modifications, in the target environment even after the Package is uninstalled.
  - **Roll back to original State**: Select this option if you want to roll back to the original state. All modifications that were done to the Security Right definition after it was installed in the target environment will be reverted. However, the Security Right definition will be retained in the target environment.
  - **Delete**: Select this option if you want to delete the Security Right definition from the target environment when uninstalling the Package.

Associated Artifacts

This tab displays all the associated artifacts of the selected Security Right definition.

Child Artifacts

This tab displays all the child artifacts of the Security Right definition.

SOA Assembly

You can package and deploy the SOA Folder along with its associated and child artifact from one environment or repository to another environment or repository.

To add SOA Folder to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **SOA Assembly**.

   The Assembly List window appears.
3. Select the assembly list that you want to package.
4. Right-click the assembly list item and then click **Add to Package Template**.

   The **Add to Package Template** window appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of the SOA Assembly in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the SOA Assembly package in the Settings form on the right side of the Add to Package Template window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Configure the required settings.
6. Click **Save Template** to save the configured settings.

Note: If the SOA Assembly is registered in Framework 4.0, ensure that Client-Service is also configured in Framework 4.0.

Runtime Initialization Settings

You can configure runtime initialization settings for SOA Assembly. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration settings for SOA Assembly

The SOA Assembly has an unique set of configuration settings required for installing, updating, and uninstalling the package. You can edit and view the configuration settings in one of the following two ways:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

The configuration settings for the SOA Assembly differs based on how the SOA Assemblies are registered. SOA Assemblies can be registered from GAC, Web Service, or Physical Path. For more information, see Resolve Custom Dependent Files for SOA Assembly.

Settings tab

You can define the configuration settings of the SOA Assembly artifact in the **Settings** tab. Set the properties in this tab to take appropriate actions while installing, updating, or uninstalling the package.

Install

Use this tab to set the configurations for installing the package.

- **SOA Assembly Exists**: Use this property to take appropriate action when the SOA Assembly that is being installed is present in the target environment.
  - **Do not install SOA Assembly**: Select this option to use the SOA Assembly that is present in the target environment.
  - **Install to overwrite existing SOA Assembly**: Select this option to overwrite the SOA Assembly that is present in the target environment with the one that present in the package.

Update

Use the settings in this tab to update an installed package.

- **Delete SOA Assembly**: Use this property to delete or retain the SOA Assembly that is present in the target environment.
  - **Yes**: Select this option to remove the SOA Assembly from the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If SOA Assembly modified after installation: Use this option to retain, roll back to original state, or delete the SOA Assembly in the target environment even after the Package is updated. This property is applicable only if **Delete SOA Assembly** is set to *Yes* and the SOA Assembly is modified.

- **Retain**: Select this option if you want to retain the SOA Assembly along with the modifications in the target environment even after the package is updated.
- **Rollback to original state**: Select this option to roll back to the original state. All modifications that were done to the SOA Assembly after it was installed in the target environment will be reverted. However, the SOA Assembly will be retained in the target environment.
- **Delete**: Select this option to delete the SOA Assembly from the target environment when updating the package.

- **No**: Select this option to retain the SOA Assembly in the target environment.

  SOA Assembly exists: Use this option to take appropriate action when the SOA Assembly that is being installed is present in the target environment.

  - **Do not install SOA Assembly**: Select this option to use the SOA Assembly that is present in the target environment.
  - **Install to overwrite existing SOA Assembly**: Select this option to overwrite the SOA Assembly that is present in the target environment with the one that present in the package.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete SOA Assembly**: Use this property to retain or delete the SOA Assembly in the target environment even after the package is uninstalled.
  - **Yes**: Select this option to delete the SOA Assembly from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the SOA Assembly in the target environment even after the package is uninstalled.
- **If SOA Assembly modified after installation**: Use this property to retain, roll back to original state, or delete the SOA Assembly in the target environment even after the package is uninstalled. This property is displayed only if **Delete SOA Assembly** is set to *Yes*. However, if **Delete SOA Assembly** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the SOA Assembly along with the modifications in the target environment even after the package is uninstalled.
  - **Rollback to original state**: Select this option to rollback to the original state. All modifications that were done to the SOA Assembly after it was installed in the target environment will be reverted. However, the SOA Assembly will be retained in the target environment.
  - **Delete**: Select this option to delete the SOA Assembly from the target environment when uninstalling the package

Resolving Associated Workflow of SOA Assembly

The **Associated Workflow Details** grid in the Settings Form contains the following fields:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field Name	Description
Workflow Name	Displays the name of the associated workflow.
Source Version	Displays the source version of the associated workflow. You can use the same values for the target, or change the values as required.
Target Version	Displays the target version of the associated workflow. You can use the same values as that in the source, or change the values as required.

Associated Artifacts of SOA Assembly

The **Associated Artifacts** tab displays all the associated artifacts of the selected SOA Assembly artifact.

SOA Assembly can include the following associated Artifacts:

- Parent Folder
- SOA Folder

When you select the **Include Associated Artifacts** option as **Yes** while adding the SOA Assembly to a Package Template, the associated artifacts are added to the package template. You can then view the associated artifacts of the SOA Assembly in the tree view of a package. If you select the **Include Associated Artifacts** option as **No**, the associated artifacts are not added to the package template and are also not visible in the tree view of a package.

Child Artifacts of SOA Assembly

The **Child Artifacts** tab displays all the child artifacts of the selected SOA Assembly artifact.

SOA Assembly can include the following child artifacts:

- Folder
- Child SOA Assembly

When you select the **Include Child Artifacts** option as **Yes** while adding the SOA Assembly to a Package Template, the child artifacts are added to the package template. You can then view the child artifacts of the SOA Assembly in the tree view of a package. If you select the **Include Child Artifacts** option as **No**, the child artifacts are not added to the package template and are also not visible in the tree view of a package.

Include Child Item

For more information, see Include Child Item.

Security Artifacts of SOA Assembly

The **Security Artifacts** tab displays all the security artifacts of the selected SOA Assembly artifact. You can enable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the security settings of the SOA assembly that you want to include in the package.

For more information about including security artifacts, see Include Security Artifacts.

Resolve Custom Dependent Files for SOA Assembly

The configuration settings for the AVEVA Work Tasks Object Access (SOA) assembly differs based on how the SOA assemblies are registered. SOA assemblies can be registered through Global Assembly Cache (GAC), Web Service, or physical path.

Register SOA Assembly from Global Assembly Cache

The **Source Version** field displays the version of SOA Assembly in the source environment. In the **Target Version** field, you can specify the version of SOA Assembly that you want to use in the target environment. If you do not specify the target version, the source version is considered while installing the package. The SOA Assembly from Global Assembly Cache gets installed in the GAC folder.

You can also package the identified custom dependent files which can be installed while installing the package in the target environment. In the **Source Files** box, in the **Custom Dependent File** section, enter the source file path of the files. Click **Add row** to add multiple rows to install additional files in the target environment.

Register SOA Assembly from a Physical Path

The **Source Physical Path** field displays the physical path of SOA Assembly in the source environment. You can also package the identified custom dependent files which can be installed while installing the package in the target environment. Click **Add row** to add multiple rows to install additional files in the target environment. Enter the source physical path of the dependent files.

If the source folder is relative to the AVEVA Work Tasks installed path, and the folder does not exist in the target environment, then a new folder is created and the files are copied to it. For example, if the path of the file, *rating.dll*, is *C:\Program files\AVEVA\Work Tasks\\Web\BPMUITemplates\CustomFolder\rating.dll***,** then a new folder named **CustomFolder** is created in the target machine and *rating.dll* is copied to it.

If the source folder is not relative to the AVEVA Work Tasks installed path, then the files are copied to the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\bin* folder. For example, if the path of the file, *rating.dll*, is **E:\SOA99\ rating.dll**, then **rating.dll** is copied to *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\bin* folder in the target machine.

The **Install in Global Assembly Cache** field is optional and is used to install the SOA assembly and custom dependent files in the GAC of the target environment. Check the **Install in Global Assembly Cache** box, if you want to install the files in the GAC of the target environment.

Click **Delete Custom File** icon to delete the selected row. The dependent files will not be added to the package template.

Register SOA Assembly from Web Service

When SOA Assembly is registered from Web Service, you must specify the **Web Target Details.** The **Web Source Details** field displays the Web Service values in the source environment.

If you select **Authentication Type** as **Anonymous** in the Source environment, then enter only the Web service URL in the **Web Target Details** field. In the target environment, if the web service requires Windows

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

authentication, then you can change the **Authentication Type** to **Windows** and enter the user name, password, and domain information.

Note:

- You must clear the cache if you have changed the user name and password after registering a SOA Assembly from Web. Otherwise, SOA Assembly will consider the old credentials for authentication and it will not validate.

- Consider that in any repository, you have created a SOA Web Service for any activity. When you configure a workflow for the activity, and add the workflow to the package template, the SOA web service also gets packaged as
an associated artifact. Even though the SOA web service is not configured for the workflow, it still gets added to the package as it is configured for the activity.

SOA Folder

You can package and deploy the SOA Folder along with its associated and child artifact from one environment or repository to another environment or repository.

To add SOA Folder to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **SOA Folder**.
   The **Folder List** window appears.
3. Select the folder list that you want to package.
4. Right-click the folder list item and then click **Add to Package Template**.
   The **Add to Package Template** window appears.
   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.
   You can set the configuration properties of the SOA Folder in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the SOA Folder package in the Settings form on the right side of the Add to Package Template window.
5. Configure the required settings.
6. Click **Save Template** to save the configured settings.

Configuration Settings for SOA Folder

The SOA Folder has an unique set of configuration settings required for installing, updating, and uninstalling the package. You can edit and view the configuration settings in one of the following two ways:

- Source Environment > Package Template screen
- Target Environment > Manage Package screen

You can define the configuration settings of the SOA Folder artifact in the **Settings** tab. Set the properties in this

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

tab to take appropriate actions while installing, updating, or uninstalling the package.

Install

Use this tab to set the configurations for installing the package.

- **SOA folder exists**: Use this property to take appropriate action when the SOA folder in the package is already present in the target environment.
  - **Do not install SOA folder**: Select this option to use the SOA folder that is present in the target environment.
  - **Install to overwrite the existing SOA folder**: Select this option to overwrite the SOA folder that is present in the target environment with the one in the package.
- **Folder Path Configuration**: Use this property to view the path of SOA folder in the target environment.
  - **Source Physical Path**: This is a read-only field that displays the physical path of the SOA folder in the source environment.
  - **Target Physical Path**: This is a read-only field that displays the physical path of SOA folder in the target environment. If this field is blank, then the physical path in the target environment will be the same as the path specified in source environment.

Update

Use the settings in this tab to update an installed package.

- **Delete SOA folder**: Use this property to delete or retain the SOA folder that is present in the target environment.
  - **Yes**: Select this option to remove the SOA folder from the target environment.

    SOA folder is modified: Use this option to retain, roll back the SOA folder to original state, or delete the SOA folder from the target environment even after the package is updated. This property is applicable only if **Delete SOA folder** is set to *Yes* and the SOA folder is modified.
    - **Retain**: Select this option to retain the SOA folder along with the modifications in the target environment.
    - **Rollback to original state**: Select this option to roll back to the original state. All modifications that were done to the SOA folder after it was installed in the target environment will be reverted. However, the SOA folder will be retained in the target environment.
    - **Delete**: Select this option to delete the SOA Folder from the target environment.
  - **No**: Select this option to retain the SOA folder in the target environment.

    SOA folder exists: Use this property to take appropriate action when the SOA folder that is being installed is present in the target environment.
    - **Do not install SOA Folder**: Select this option to use the SOA folder that is present in the target environment.
    - **Install to overwrite existing SOA Folder**: Select this option to overwrite the SOA folder that is present in the target environment with the one that is in the package.
- **Folder Path Configuration**: Use this property to set the path of SOA folder in the target environment.
  - **Source Physical Path**: This is a read-only field that displays the physical path of SOA folder in the source

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

environment.

- **Target Physical Path**: This is a read-only field that displays the physical path of SOA folder in the target environment.

  If the source folder is relative to the AVEVA Work Tasks installed path, then the **Target Physical Path** points to the same relative path in the Target environment. If the folder does not exist, ensure to create the folder before installing the package. If the source folder is not relative to the AVEVA Work Tasks installed path, then the **Target Physical Path** points to the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\bin* folder.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete SOA folder**: Use this property to retain or delete the SOA folder from the target environment even after the package is uninstalled.
  - **Yes**: Select this option to delete the SOA folder from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the SOA folder in the target environment even after the package is uninstalled.
- **SOA folder is modified**: Use this option to retain, roll back the SOA folder to original state, or delete the SOA folder from the target environment even after the package is uninstalled. This property is displayed only if **Delete SOA folder** is set to *Yes*. However, if **Delete SOA folder** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the SOA folder along with the modifications in the target environment.
  - **Rollback to original state**: Select this option to roll back to the original state. All modifications that were done to the SOA folder after it was installed in the target environment will be reverted. However, the SOA folder will be retained in the target environment.
  - **Delete**: Select this option to delete the SOA folder from the target environment.

Associated Artifacts for SOA Folder

SOA Folder does not have any associated artifacts.

Child Artifacts for SOA Folder

SOA Folder does not have any child artifacts.

Security Artifacts for SOA Folder

You can enable the security settings of the SOA folder that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Web API

You can package and deploy a Web API along with its associated artifacts from one environment or repository to another environment or repository.

Adding Web API to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, click **Settings** and then click **Web API**.

   The Web API window appears.
3. Right-click on a Web API, and then click **Add to Package Template**.

   The **Add to Package Template** dialog box appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of the Web API in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the Web API package in the **Settings** form on the right side of the **Add to Package Template** window.
4. Configure the required settings.
5. Click **Save Template** to save the configured settings.

Runtime Initialization Settings

You can configure runtime initialization settings for Web APIs. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration Settings for Web API

The **Web API** has a unique set of configuration settings required to install, update, and uninstall the package. You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

Settings

You can define the configuration settings of the Web API artifact in the Settings tab. Set the properties in this tab to take appropriate actions while installing, updating, or uninstalling the package.

Install

Use this tab to set the configurations for installing the package.

- **Web API exists**: Use this property to take appropriate action when the Web API that is being installed is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

present in the target environment.

- **Do not install Web API**: Select this option to use the Web API that is present in the target environment.
- **Install to overwrite existing Web API**: Select this option to overwrite the Web API that is present in the target environment with the one that present in the package.

- **Web API Configuration**: Displays the URL of the Web API in the source environment and it also allows you to specify the URL of the Web API in the target environment:
  - **Source Base Address**: Displays the URL of the Web API in the source environment
  - **Target Base Address**: Specify the URL of the Web API in the target environment. If it is left blank, then the source base address will be considered.
  - **Source Authentication Mode**: Displays one of the following authentication mode options in the source environment:
    - **No Authentication**: If the Web API does not require authentication.
    - **Windows**: If the Web API is Windows authenticated.
    - **User Access Token**: If the Web API requires the currently logged-in user's access token.
    - **Basic Authentication**: If the Web API requires basic authentication. Displays the user name with domain and the password.
    - **Certificate**: If the Web API requires certificate authentication. Displays the certificate store location and the thumbprint.
    - **Service to Service Access Token**: If the Web API requires Service to Service Access Token.
  - **Target Authentication Mode**: Select one of the following authentication mode options in the target environment:
    - **No Authentication**: If the Web API does not require authentication.
    - **Windows**: If the Web API is Windows authenticated.
    - **User Access Token**: If the Web API requires the currently logged-in user's access token.
    - **Basic Authentication**: If the Web API requires basic authentication. Provide the user name with domain and the password.
    - **Certificate**: If the Web API requires certificate authentication. Provide the certificate store location and the thumbprint. For more information on certificate-based authentication, see Web API Certificate-based Authentication.

      Note: The thumbprint should not contain any spaces. In some machines, the characters in thumbprint should be in upper case.
    - **Service to Service Access Token**: If the Web API requires Service to Service Access Token. Provide **Authority URL**, **Audience**, **Client ID**, **Client Secret**, and **Scope** to get the access token.

      In the **Authority URL**, provide the token end-point of the respective repository provider. For example,
      AVEVA Connect: `{authority url}/oauth/token`
      Azure Active Directory: `{authority url}/oauth2/v2.0/token`
      AVEVA Identity Manager: `{authority url}/connect/token`

Update

Use the settings in this tab to update an installed package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Delete Web API**: Use this property to delete or retain an existing Web API that is present in the target environment.
  - **Yes**: Select this option to remove the Web API from the target environment.

    Web API is modified: Use this option to retain, rollback to original state, or delete the Web API in the target environment even after the package is updated. This property is applicable only if the **Delete Web API** is set to *Yes* and the Web API is modified.
    - **Retain**: Select this option to retain the Web API along with the modifications in the target environment even after the package is updated.
    - **Rollback to original state**: Select this option to rollback to the original state. All modifications that were done to the Web API after it was installed in the target environment will be reverted. However, the Web API will be retained in the target environment.
    - **Delete**: Select this option to delete the Web API from the target environment when updating the package.
  - **No**: Select this option if you want to retain the Web API in the target environment.

    Web API exists: Use this property to take appropriate action when the Web API that is being installed is present in the target environment.
  - **Do not install Web API**: Select this option to use the Web API that is present in the target environment.
  - **Install to overwrite existing Web API**: Select this option to overwrite the Web API that is present in the target environment with the one that present in the package.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete Web API**: Use this property to decide if you want to retain or delete the Web API in the target environment even after the package is uninstalled.
  - **Yes**: Select this option to delete the Web API from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the Web API in the target environment even after the package is uninstalled.
- **Web API is modified**: Use this property to retain, rollback to original state, or delete the Web API in the target environment even after the package is uninstalled. This property is displayed only if **Delete Web API** is set to *Yes*. However, if **Delete Web API** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the Web API along with the modifications in the target environment even after the package is uninstalled.
  - **Rollback to original state**: Select this option to rollback to the original state. All modifications that were done to the Web API after it was installed in the target environment will be reverted. However, the Web API will be retained in the target environment.
  - **Delete**: Select this option to delete the Web API from the target environment when uninstalling the package

Associated Artifacts of Web API

Web API can have Web API Source as the associated artifact.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Child Artifacts of Web API

Web API does not have any child artifacts.

Security Artifacts of Web API

You can enable the security settings of the Web APIs that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

Web API Source

You can package and deploy a Web API Source from one environment or repository to another environment or repository.

To add Web API Source to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, click **Settings** and then click **Web API Source**.
3. Right-click on a Web API Source, and then click **Add to Package Template**.

   The **Add to Package Template** dialog box appears.

   For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

   You can set the configuration properties of the Web API Source in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the Web API Source package in the **Settings** form on the right side of the **Add to Package Template** window.
4. Configure the required settings.
5. Click **Save Template** to save the configured settings.

Configuration Settings for Web API Source

The **Web API Source** has a unique set of configuration settings required to install, update, and uninstall the package. You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

Settings

You can define the configuration settings of the Web API Source artifact in the Settings tab. Set the properties in this tab to take appropriate actions while installing, updating, or uninstalling the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Install

Use this tab to set the configurations for installing the package.

- **Web API Source listitem exists**: Use this property to take appropriate action when the Web API Source that is being installed is present in the target environment.
  - **Do not install Web API Source listitem**: Select this option to use the Web API Source listitem that is present in the target environment.
  - **Install to overwrite the existing Web API Source listitem**: Select this option to overwrite the Web API Source listitem that is present in the target environment with the one that is present in the package.
- **Web API Configuration**: Displays the Base Address and Swagger URL configured in the source environment. It also allows you to specify the Base Address and Swagger URL in the target environment.

  Note: Web API Source is the associated artifact of Web API. If you change the Swagger URL of Web API Source in the target environment, you need to reselect the **Source** in Web API List Item to get the updated Swagger URL in the target environment.

Update

Use the settings in this tab to update an installed package.

- **Delete Web API Source listitem**: Use this property to delete or retain an existing Web API Source listitem that is present in the target environment.
  - **Yes**: Select this option to remove the Web API Source listitem from the target environment.
  - **No**: Select this option if you want to retain the Web API Source listitem in the target environment.
- **Web API Source listitem exists**: Use this property to take appropriate action when the Web API Source listitem that is being installed is present in the target environment.
  - **Do not install Web API Source listitem**: Select this option to use the Web API Source listitem that is present in the target environment.
  - **Install to overwrite existing Web API Source listitem**: Select this option to overwrite the Web API Source listitem that is present in the target environment with the one that is present in the package.
- **Web API Configuration**: Displays the Base Address and Swagger URL configured in the source environment. It also allows you to specify the Base Address and Swagger URL in the target environment.

  Note: Web API Source is the associated artifact of Web API. If you change the Swagger URL of Web API Source in the target environment, you need to reselect the **Source** in Web API List Item to get the updated Swagger URL in the target environment.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete Web API Source listitem**: Use this property to decide if you want to retain or delete the Web API Source listitem in the target environment even after the package is uninstalled.
  - **Yes**: Select this option to delete the Web API Source listitem from the target environment when the package is uninstalled.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **No**: Select this option to retain the Web API Source listitem in the target environment even after the package is uninstalled.

- **Web API Source listitem is modified**: Use this property to retain, rollback to original state, or delete the Web API Source listitem in the target environment even after the package is uninstalled. This property is displayed only if **Delete Web API Source listitem** is set to *Yes*. However, if **Delete Web API Source listitem** is set to *No*, this property is not displayed.
  - **Retain**: Select this option to retain the Web API Source listitem along with the modifications in the target environment even after the package is uninstalled.
  - **Rollback to original state**: Select this option to rollback to the original state. All modifications that were done to the Web API Source listitem after it was installed in the target environment will be reverted. However, the Web API Source will be retained in the target environment.
  - **Delete**: Select this option to delete the Web API Source listitem from the target environment when uninstalling the package

Associated Artifacts of Web API Source

Web API Source does not have any associated artifacts.

Child Artifacts of Web API Source

Web API Source does not have any child artifacts.

Security Artifacts of Web API Source

You can enable the security settings of the Web API Sources that you want to include in the package. For more information about including security artifacts, see Include Security Artifacts.

Workflows

You can package and deploy the workflow definition with its associated and child artifact from one environment or repository to another environment or repository.

To add a workflow to a package template, you must first publish the workflow. Ensure that the workflow that you want to package is not disabled.

You can resolve workflow activities by editing the database and also identifying the associated artifacts while installing the package. This is done to ensure that the workflow is run without any errors. For more information, see Resolving Workflow Activities.

To add Workflows to Package Template

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, and then click **Workflows**.
   The **Workflows** page appears.
3. Select the workflow that you want to package.
4. Right-click the workflow, and then click **Add to Package Template**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Add to Package Template window appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

You can set the configuration properties of the workflow in the Add to Package Template window. The left side of the configuration window shows the package template and its configurations. You can set the configuration properties of the workflow package in the **Settings** form on the right side of the Add to Package Template window.

Runtime Initialization Settings

You can configure runtime initialization settings for workflows. For more information about configuring runtime initialization settings, see Runtime Initialization Settings.

Configuration Settings for Workflows

Workflows have a unique set of configuration settings required to install, update, and uninstall the package. You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

Settings

You can define the configuration settings of Workflows artifact in the **Settings** tab. Set the properties in this tab to take appropriate actions while installing, updating, or uninstalling the package.

Install

Use this tab to set the configurations for installing the package.

- **Inherit Artifact Type Settings:** Use this property to define whether the settings inherited from the artifact type or the settings configured at the artifact-level must be considered.
  - **Yes**: Select this option if the workflow must inherit the settings configured for the artifact type. If this option is selected, then the artifact type settings take precedence.

    No: Select this option if the workflow must consider the settings configured at the artifact-level. The following options are available:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Install Workflow as Draft	Use this property to install the Workflow. • **Yes**: Select this option to install the Workflow in the Draft status in the target environment. • **No**: Select this option to install the Workflow in the Published status in the target environment.
Workflow Exists	Use this property to take appropriate action when the Workflow definition specified in the package is already present in the target environment. • **Install as next major version**: Select this option to install the Workflow as a next major version. • **Install as next minor version**: Select this option to install the Workflow as a next minor version. You can install the Workflow as a next minor version up to 12 times only. • **Install as next version**: Select this option to install the Workflow as a next version. • **Overwrite latest (published) version**: Select this option to overwrite the latest published version of workflow in the target environment. An existing Workflow Definition gets replaced with the Workflow Definition contained in this package. • **Do not install**: Select this option if you do not want to install the Workflow, as it is already present in the target environment.

- **Running Workflow Instance**: Use this property if you want to kill the running workflow instance or do not want to overwrite the workflow. You can also pause and resume the running workflow instance. This property is available only when you select the **Overwrite latest (published) version** option for **Workflow Exists** property.
  - **Kill**: Select this option to kill the running workflow instance.
  - **Pause and Resume**: Select this option to pause, overwrite, and resume the running workflow instance in the target environment.
- **Associated Artifacts Not Present**: Use this property to take appropriate action when the associated artifacts of the Workflow are not present in the target environment.

  This property is not applicable for the associated artifact of the type Event Association. The artifact will be installed even if the Event Association is unavailable in the target environment.
  - **Install as draft**: Select this option to install the Workflow in the *Draft* status, even if the associated artifacts are not present in the target environment.
  - **Do not install**: Select this option if you do not want to install the Workflow as its associated artifacts are

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

not present.

Note: Consider, the latest published workflow exists in the target environment, and the **Overwrite latest (published) version** option is selected. During package installation, the backup of the existing workflow is taken and stored in the database. And, the workflow in the target environment is overwritten with the packaged source definition.

During Overwrite, the package name and the description does not get updated in the artifact **Package Name (Version)** column.

For Pause, Resume, and Kill scenarios, the workflow needs to be published manually to get the latest workflow definition.

Consider the running instances of the workflow waiting for the user activity. If you rename those activities and overwrite the workflow while installing the running workflow instance through the package with the **Pause and Resume** option selected, the workflow will not get resumed.

Update

Use the settings in this to update an installed package.

- **Delete Workflow**: Use this property to delete the Workflow that is present in the target environment.
  - **Yes**: Select this option to remove the Workflow from the target environment.

    Workflow is modified: Use this option to retain or delete the workflow in the target environment even after the package is updated. This property is applicable only if **Delete Workflow** is set to *Yes* and the workflow is modified.
    - **Retain**: Select this option to retain the Workflow along with the modifications, in the target environment.
    - **Rollback to original state**: Select this option to rollback the Workflow to the original state if you have made any modifications in the target environment. You can kill, pause, or resume the running workflow instance.

      - **Kill**: Select this option to kill the running workflow instance.

      - **Pause and Resume**: Select this option to pause, rollback, and resume the running workflow instance in the target environment.
    - **Delete**: Select this option to delete the Workflow from the target environment.
  - **No**: Select this option to retain the Workflow in the target environment. When you select *No*, the **Install Workflow as draft**, **Workflow exists** and **Associated artifacts not present** options appear.

    Only when the **Delete Workflow** option is set to **No**, the **Inherit Artifact Type Settings** option is enabled.
- **Inherit Artifact Type Settings:** Use this property to define whether the settings inherited from the artifact type or the settings configured at the artifact-level must be considered.
  - **Yes**: Select this option if the workflow must inherit the settings configured for the artifact type. If this option is selected, then the artifact type settings take precedence.
  - **No**: Select this option if the workflow must consider the settings configured at the artifact-level. The following options are available:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Install Workflow as Draft	Use this property to install the Workflow. • **Yes**: Select this option to install the Workflow in the Draft status in the target environment. • **No**: Select this option to install the Workflow in the Published status in the target environment.
Workflow Exists	Use this property to take appropriate action when the Workflow definition specified in the package is already present in the target environment. • **Install as next major version**: Select this option to install the Workflow as a next major version. • **Install as next minor version**: Select this option to install the Workflow as a next minor version. You can install the Workflow as a next minor version up to 12 times only. • **Install as next version**: Select this option to install the Workflow as a next version. • **Overwrite latest (published) version**: Select this option to overwrite the latest published version of workflow in the target environment. An existing Workflow Definition gets replaced with the Workflow Definition contained in this package. • **Do not install**: Select this option if you do not want to install the Workflow, as it is already present in the target environment.

- **Running Workflow Instance**: Use this property if you want to kill the running workflow instance or do not want to overwrite the workflow. You can also pause and resume the running workflow instance. This property is available when you select the **Overwrite latest (published) version** option for **Workflow Exists** property.
  - **Kill**: Select this option to kill the running workflow instance.
  - **Pause and Resume**: Select this option to pause, overwrite, and resume the running workflow instance in the target environment.
- **Associated artifacts not present:** Use this property to take appropriate action when the associated artifacts of the Workflow are not present in the target environment.

  This property is not applicable for the associated artifact of the type Event Association. The artifact will be installed even if the Event Association is unavailable in the target environment.
  - **Install as draft:** Select this option to install the Workflow in *Draft* status, even if the associated artifacts are not present in the target environment.
  - **Do not install:** Select this option if you do not want to install the Workflow as its associated artifacts are not present.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: Consider, the latest published workflow exists in the target environment, and the **Overwrite latest (published) version** option is selected. During package update, the backup of the existing workflow is not be taken if available; and taken if not available. And, the workflow in the target environment is overwritten with the packaged source definition.

During Overwrite, the package name and the description does not get updated in the artifact **Package Name (Version)** column.

Uninstall

Use this tab to configure the settings when the package is being uninstalled from the target environment.

- **Delete Workflow**: Use this property to retain or delete the Workflow in the target environment even after the package is uninstalled..
  - **Yes**: Select this option to remove the Workflow from the target environment when the package is uninstalled.
  - **No**: Select this option to retain the Workflow in the target environment even after the package is uninstalled.
- **Workflow is modified**: Use this property to retain or delete the Workflow in the target environment even after the package is uninstalled. This property is displayed only if **Delete Workflow** is set to **Yes**. However, if **Delete Workflow** is set to **No**, this property is not displayed.
  - **Retain**: Select this option to retain the Workflow along with the modifications even after the package is uninstalled.
  - **Rollback to original state**: Select this option to rollback the Workflow to the original state if any modifications are made in the target environment.
  - **Delete**: Select this option to delete the Workflow from the target environment.
- **Running Workflow Instance**: Use this property if you want to kill the running workflow instance or do not want to overwrite the workflow. You can also pause and resume the running workflow instance. This property is available when you select the **Rollback to original state** option for **Workflow is modified** property.
  - **Kill**: Select this option to kill the running workflow instance.
  - **Pause and Resume**: Select this option to pause, rollback, and resume the running workflow instance in the target environment.

    Note:
    - During package uninstall, with the **Rollback to original state** option selected, the overwritten workflow definition is replaced with backup data, and the backup from the database gets deleted.
    - During rollback, the package name and the description does not get updated in the artifact **Package Name (Version)** column.

Associated Artifacts of Workflows Definition

The **Associated Artifacts** tab displays all the associated artifacts of a selected Workflow.

When you set the **Include Associated Artifacts** option to **Yes**, while adding the workflow to a package template, then the associated artifacts are added to the package template, and it also allows you to view the associated artifacts of the workflow in the tree view of the package. If you set the **Include Associated Artifacts** option to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

No, then the associated artifacts are not added to the package template, and are also not visible in the tree view of the package. This condition is also applicable for the **Include Child Artifacts** and **Include Security Artifacts** options.

A workflow can have the following associated artifacts:

- Parent Folder
- Workflows
- Calendar
- SOA Assembly
- SOA Folder
- Database Connection
- Forms
- Event Associations
  - Scheduler
  - Enterprise Console
  - File Watcher
- Web API
- Queue
- Analyst's Process Model
- Work Item Field
- Logic App Connection
- Message Queue Connection

You can import and export a workflow with event associations from one repository to another repository. During install, when the associated artifact is of type **Event Association**, workflow artifact does not validate whether **Event Association** is available or not, as Event Association cannot exist without workflow. While installing the package, you must ensure that the event association is available in the same package or in a different package, so that it is installed and associated to the workflow.

Note:

- The **Analyst's Process Model** will not appear as an associated artifact when the value for the **Analyst's Process Model Filename for Execution View** is set through the **Expression Editor** as the value is resolved only during run-time.

- If a SOA Assembly is registered through a Web Service and a physical path (dll), by default the Web Service is a static method and the physical path can be a static or non- static method. While packaging workflows that does not use SOA Assembly, by default the static objects of the registered SOA Assembly is considered as an associated artifact.

- When you package a workflow that is associated from Form, the Form does not appear as associated artifact of the workflow.

- When you package a Workflow, the Work Item Fields mentioned below are also packaged, and the Work Item Fields are displayed in the tree view of the package and the **Associated Artifacts** tab of the selected workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- All the Work Item Fields at the workflow level
- The application level Work Item Fields, which are used in that specific workflow

Identifying Event Association as Associated Artifact

Consider that you want to package a workflow, *Telematics* of version 2. While adding to package template, the package identifies **Event Association** as associated artifact based on the following conditions:

- Event Association mapped to version 2 of Telematics workflow.
- Event Association mapped to latest version of Telematics workflow (version 2 of the workflow "Telematics" is the latest version).

Child Artifacts for Workflows

The **Child Artifacts** tab displays all the child artifacts of the Workflow.

A workflow can have another workflow as its child artifact.

- Workflow

Include Child Item

For more information, see Include Child Item.

Security Artifacts

The **Security Artifacts** tab displays the security artifacts of a workflow, if any. For more information about including security artifacts, see Include Security Artifacts.

Activity Rebuild and Complete Rebuild

In the package template, an orange icon, , appears for the workflow which denotes that the workflow needs to be rebuilt as the workflow is published, unpublished, or edited after it is added to the package template. Similarly, in the target environment if the workflow installed by the package is published, unpublished, or edited, then while uninstalling the package, the workflow settings will be retained.

When a workflow in the package template is modified, ensure to perform a rebuild on the particular workflow. You can right-click the workflow from the left hand side configuration tree view in the package template, and click **Complete Rebuild** or **Activity Rebuild** to rebuild the workflow.

- **Complete Rebuild**: Click **Complete Rebuild** to rebuild the particular workflow artifact in the package template. During complete rebuild, the install, update, or uninstall settings do not change. However, the settings done to resolve any activity properties in the target section of the **Settings** form get cleared.
- **Activity Rebuild**: Click **Activity Rebuild** to rebuild only a modified workflow artifact. All settings done to resolve any activity property in the target section of the **Settings** form does not get changed. Only the new values of the modified activity get updated in the package. However, if you want to rename an activity in the workflow, use the **Complete Rebuild** option. Any changes in the property of a workflow do not get updated in the package unless the activity name is modified.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: Consider a scenario where a workflow 'Workflow1' having associated artifact 'List 1' is added to package template. If this workflow is rebuilt after modifying and changing the associated artifact to 'List 2', then 'List 2' will be added to the package template as the associated artifact. However, 'List 1' will not be deleted form the package template. If required, you can manually delete 'List1' from the package template.

The same behavior is applicable for other artifacts.

When an artifact, for example, a workflow is added to a package template, its associated artifacts are also added to the package template. If you delete an artifact from the workflow, and then perform a complete rebuild on the same package template, the deleted artifact is removed from the target environment. However, the associated artifacts of the deleted artifact are retained in the target environment. For example, consider that a workflow contains a form, and this workflow is added to the package template. While installing the package template, the associated artifacts of the form, such as the associated database lookup also get installed in the target environment. Now, if the form artifact is deleted from the workflow, and if the package template is rebuilt again, though the form is removed, its associated artifacts, such as the database lookup is retained in the target environment.

For information about resolving workflow activities, see Resolve Workflow Activities.

Configure Artifact Type Settings for Workflow

You can set the status in which all the workflow artifacts in a package must be installed or updated in the target environment.

To configure artifact type settings for Workflow in the following tabs:

- **Install**

  Use this tab to set the configurations for installing the package.
- **Update**

  Use the settings in this tab to update an installed package.

Artifact Type Settings

1. In the **Enterprise Console**, click menu icon, click **Package**, and then click **Package Template**.

   The **Package Template** page appears.
2. To edit an existing package, in the **Package Template** page, select the package template, and then click **Edit**. The **Edit Package Template** dialog box appears.
3. In the **Configuration** tree on the left pane, click **Workflows** under **Artifacts**. The **Settings** form with the **Artifact Type Settings** section appears.
4. For the **Install Workflows as Draft** property, select one of the following:
   - **Yes**: Select this option to install all Workflow artifacts in the package in *Draft* status in the target environment.
   - **No**: Select this option to install all Workflow artifacts in the package in *Published* status in the target environment.

     The status of the Workflow in a package is set based on the value defined.
5. Use the **Workflows Exist** property to take appropriate action when the Workflow definition specified in the package is already present in the target environment. Select one of the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Install as next major version**: Select this option to install all Workflow artifacts in the package as next major version.
- **Install as next minor version**: Select this option to install all Workflow artifacts in the package as next minor version.
- **Install as next version**: Select this option to install all Workflow artifacts in the package as next version.
- **Do not install**: Select this option if you do not want to install the Workflow artifacts, as they are already present in the target environment.

Note: You can also configure these settings while installing the package using the **Manage Package** page.

Resolve Workflow Activities

You can resolve workflow activities while installing the package. This is done to ensure that the workflow is executed without any errors.

During workflow installation, the values entered in the target fields of the **Settings** form are considered for installation. If values are not entered for the target fields, the values of the source fields are considered.

Note that a package does not resolve the values of those properties for which expression editor is used. You might have to manually define the value in the target environment or copy the required artifacts.

Rebuild Scenario

When you rebuild a workflow that was edited after adding it to the package template, the package template retains the old data in the settings form, but considers only the latest record. This does not affect the functionality of the workflow that is rebuilt. For example, consider a workflow that contains an Invoke Form activity which points to an employee list. After adding the workflow to the package template, consider that the invoke form activity is modified to customer list. When you rebuild workflow, the Settings form will contain both the employee list and customer list, but will consider only the customer list.

Workflow Activities Resolved

The following table shows the properties of the activities that are resolved:

Categories	Activity	Resolved Properties	Comments
-	Start Activity	SOA Object Instances	SOA Assembly and SOA Folder are identified as associated artifacts. If the SOA Object is used in any of the expression editor across workflow, then the same will be resolved.
		Exception Template	For more information, see Resolving User or Role Based Exception Template.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Categories	Activity	Resolved Properties	Comments
		XML Variables	For more information, see Resolving XML Variables.
Human Activities	Task Activity	Task Type	-
	All Human Activities	Custom Document View URL	Resolve URLs or paths specified in the workflow. For more information, see Resolving Custom Document View URL.
		Redirect URL	Resolve the redirect URLs or redirect paths specified in the workflow. For more information, see Resolving Redirect URL.
		Delivery Channels	For more information, see Resolving Delivery Channels.
		Set Calendar for Timeout	The calendar of type Organization is identified as an associated artifact. The Resource Calendar is not considered as an associated artifact. You must ensure that if the workflow uses any resource calendar, the resource calendar is available in the target environment.
	Invoke Form Activity	Form	Identified as an associated Form.
		Version	In the target environment, if multiple versions of the form are present, then the latest published version of the associated form is considered. For more information, see Mapping the Version

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Categories	Activity	Resolved Properties	Comments
			of the Associated Form.
		Template Path	For more information, see Resolving Template Path.
Engine Activities	Update Parent Variables	Parent Workflow Application Name	-
		Parent Workflow Name	Identified as an associated workflow. In the target environment, if multiple versions of the workflow are present, then the latest published version of the workflow is considered.
	Workflow Activity	Repository	-
		Workflow	Identified as an associated workflow.
		Version	In the target environment, if multiple versions of the workflow are present, then the latest published version of the workflow is considered. For more information, see Mapping the Version of the Associated Workflow to the Parent Workflow.
	Database Activity	Database Connection	Database Connection is identified as associated artifact. For all the Command Types of a Database Connection, the target database must have all the required tables and stored procedures. While installing, the package references only the tables

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Categories	Activity	Resolved Properties	Comments
			and stored procedures. Ensure that you have created all the required tables and stored procedures in the target environment.
Communication Activities	Wait for Email Activity	Attachment Save Location	For more information, see Resolving File Details.
Integration Activities	InvokeWebService Activity	WebService Details	For more information, see Resolving Web Service Configurations.
	File Operations Activity	• Source Details • Destination Details	For more information, see Resolving File Details.
	Create-Open-Office Document Activity	Destination Details	For more information, see Resolving File Details.
	Script Activity	References property	Resolve the assembly file path in the target environment. For more information, see Resolving Script Activity References.
List Activities	List Navigator	List Name or ID	Identified as an associated List.
	Wait For Event	List ID	Identified as an associated List.
Report Activities	Report Generator	Report Template	Identified as an associated Report Template. The corresponding Report Connection will be added as an associated artifact for the report template.
MES Activities	MES XML Command Activity	• Unique Environment Name • Middleware Machine Name	For more information, see Resolve MES XML Command Activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Categories	Activity	Resolved Properties	Comments
Logic App	Wait for Logic App Alert Activity	• Enterprise Console • URL • Authorization Token	For more information, see Resolve Wait For Logic App Alert
Logic App	Call Logic App Activity	Logic App Connection	Logic App Connection is identified as associated artifact. For more information, see Resolve Call Logic App Connection
Message Queue	SendMQData Activity WaitForMQData Activity	Message Queue Connection	Message Queue Connection is identified as associated artifact.

Map the Version of the Associated Workflow to the Parent Workflow

When you package a workflow containing a Workflow activity which is in turn associated to other workflows, you can change the configuration settings of the current workflow in the **Associated Workflow Details** grid that appears towards the lower section of the settings form.

In the **Associated Workflow Details** grid, you can specify the version of the associated workflow to which the current workflow should be mapped. In this grid, you can specify the version of the associated workflow to which the current workflow should be mapped in the target environment. However, the Associated Workflow Details grid will not be available in the configuration settings of the current workflow, if Workflow Activity expression editor is used to define the Workflow property.

You use the Associated Workflow Details grid to specify only the target version of the associated workflow to which the current workflow should be mapped. The configuration of the associated workflow is done separately from the package temple as it is added as an individual artifact.

The **Associated Workflow Details** grid contains the following fields:

- **Workflow Name**: Displays the name of the associated workflow.
- **Source Version**: Displays the source version of the associated workflow. You can use the same values for the target, or change the values as required.
- **Target Version**: Displays the target version of the associated workflow. You can use the same values as that in the source, or change the values as required.

  In case you do not edit the **Target version** field and retain the default value 'Latest', then the latest version of the associated workflow will be mapped to the current version in the target environment.

  If the **Target version** is left blank, then the source version of the associated workflow present in the target environment will mapped to the current workflow. If the source version is not present in the target environment, then the latest version of the associated workflow will be mapped to the current version in the target environment.

  If the **Target version** of the workflow is not present in the target environment, then the latest version of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

associated workflow will be mapped to the current version in the target environment.

Scenario where duplicate messages are logged in history for an artifact:

While installing or uninstalling a workflow containing a workflow activity, you might see duplicate log messages in history. This occurs if the workflow in the workflow activity is also a child workflow of the workflow being packaged.

In case of installation, the child workflow gets installed first and then the parent workflow is installed, and the following messages are logged in history:

- When the child workflow is installed, the following message is logged:

  Associated artifacts not present and installed as draft: <Name of the parent workflow>
- Later when the parent workflow is installed, the following message is logged:

  Workflow installed successfully.

  While uninstalling, the parent workflow gets uninstalled first and then the child workflow is uninstalled.
- When the parent workflow is uninstalled, the following message is logged:

  Workflow not uninstalled as child artifact(s) are present.
- Later after the child workflow is uninstalled, the parent workflow is again uninstalled and the following message is logged:

  Workflow uninstalled successfully.

Map the Version of the Associated Form

While installing a package, if the workflow contains an Invoke Form activity, you can change the configuration settings in the **Associated Form Details** grid to map the version of the associated form to the current workflow in the target environment.

You use the Associated Form Details grid to specify the version of the associated form to which the current workflow should be mapped in the target environment. In this grid, the Form Name property is not available in the configuration settings if the form name is configured using expression editor. You use the Associated Form Details grid to specify only the target version of the associated form to which the current form should be mapped. The configuration of the associated form is done separately from the package template as it is added as an individual artifact.

In the **Associated Form Details** grid, you can enter the version of the associated form to which the current form should be mapped. The **Associated Form Details** grid contains the following fields:

- **Form Name**: Displays the associated form name.
- **Source Version**: Displays the source version of the associated form. You can use the same values for the target, or change the values as required.
- **Target Version**: Displays the target version of the associated form. You can use the same values as that in source, or change the values as required.

  If you do not edit the **Target Version** field and retain the default value as "Latest", then the latest version of the associated form is mapped to the Invoke Form activity in the target environment.

  If the **Target version** field is left blank, then the source version of the associated form present in the target environment is mapped to the current form. If the source version is not present in the target environment,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

then the latest version of the associated form is mapped to the current version in the target environment.

If the **Target version** of the form is not present in the target environment, then the latest version of the associated form is mapped to the Invoke Form activity in the target environment.

Resolve Web Service Configurations

While installing a package or adding a workflow to package template, if the workflow activity contains InvokeWebService, you can edit the URL and credentials in the target.

If you have several InvokeWebService activities in the same workflow and if all the activities use one Web Service details, then the package identifies the Web Service as a single one and rectifies it accordingly.

If the workflow activity contains InvokeWebService activity, the **Edit WebService(s) used in workflow grid** to configure the URL and credentials in the target appears towards the lower section of the settings form.

The **WebService Details** grid contains the following fields:

- **Web Service Name:** This is the name of the web service.
- **Edit WebService Details:** This option enables you to edit the web service details.
- **Web Service ID:** This is a Read-only field and displays the ID for the respective web service. The ID is unique for each web service.

To edit Web Service Details

1. Click the **Edit WebService Details** button to configure the web service details. The WebService Details grid replaces the settings form.

   The WebService Details grid contains the following fields:

   Web service URL Credentials

   - **Web Service URL**: Type the Web Service URL. You must enter a WSDL URL. You can also enter an IPv6 address for the WebService URL. The IPv6 address must be enclosed within square brackets.
   - **Authentication Type:** Choose one of the authentication types, Windows or Anonymous.
     - **Windows**

       - User Name

       - Password

       - Domain Name

       - Proxy URL
     - **Anonymous**

       Bind Parameters Credentials
   - **Authentication Type:** Choose one of the authentication types, Windows or Anonymous.
     - **Windows**

       - User Name

       - Password

       - Domain Name
     - **Anonymous**

The **Source** fields displays the Web Service values in the source environment. You can use the same Web Service

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

credentials for the target, or change the values as required.

If you select **Anonymous** authentication type in the Source environment, then in the Target you need to enter only the Web service URL. In the target environment if web service requires windows authentication, then you can change the Authentication type to **Windows** mode and enter the **Log on Name**, **Password**, and **Domain Name**.

Note:
- When you first install a package that contains InvokeWebService activity, entries are made into the database. When you update the package subsequently, the **Web service URL Credentials** will not be updated in the database. Therefore, the updated credentials will not be available in the InvokeWebService activity. However, the credentials entered while installing package for the first time will be considered.

- When you uninstall a package that contains **InvokeWebService** activity, the database entries for **InvokeWebService** activity are not uninstalled.

Resolve File Details

While installing a package or adding a workflow to package template, if the workflow activity contains File Operations, Create-Open-Office document, and Wait for Email activities, you can edit the details of each file.

If the workflow activity contains File Operations, Create-Open-Office document, and Wait for Email activities, the **File Details** grid to configure the details of each file appears towards the lower section of the settings form. Use the **Edit File Location** button against each activity name.

The File Details grid contains the following fields:

- **Activity Name**: Displays the name of the activity that contains any of the File Operations, Create-Open-Office document, or Wait for Email activities.
- **Property Name**: Displays the property name. For example, Source Details, Destination Details, Attachment Save Location.
- **Location**: Displays the configured location of the file. Use the **Edit File Location** button to configure the target location of a file.

  Click **Edit File Location** for the file that you want to change. The File Details page appears replacing the package **Settings** form.

  The File Details page contains the following fields:

  - The **Source** fields display the values in the source environment. You can use the same values or credentials for the target, or change the values as required.
  - If you select **File** mode as the **Protocol** type in the source environment, then in the target environment you do not need to enter the user credentials. In the target environment, if file details require **FTP-HTTP** mode as the **Protocol** type, then you can change the **Protocol** type to **FTP-HTTP** mode and enter the **Location**, **Pattern**, **User name**, **Password**, **Domain name**, and **Port number**.
  - The **Port number** is independent of authentication type. If the target **Port number** is not specified, package considers the source **Port number**.
  - Click **Save Settings** to save the configuration. The **File Details** page closes, and the package **Settings** form appears.

    Note: Package does not validate the file details specified in the target fields. Ensure to provide valid inputs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Resolve Custom Document View URL

While installing a package, if the workflow activity contains a **Custom Document View URL** property, you can change the URL or path in the target. If the workflow contains a custom document view URL, the **Resolve Activity Properties** grid to configure the details of the URL in the target environment appears towards the lower section of the settings form.

The **Resolve Activity Properties** grid contains the following fields:

- **Activity Name**: Displays the name of the activity that contains the custom document view URL.
- **Property Name**: Displays the property name. For example, Custom Document View URL.
- **Source Value**: Displays the URL or path values in the source environment. You can use the same values for the target, or change the values as required.
- **Target Value**: Displays the URL or path values in the target environment. You can use the same values as that in the source, or change the values as required. You can select multiple properties with the same URL.

  You must not change the form definition. Consider that a form is an associated artifact of a custom URL. When you install the package, the target version of the form might not be same as the source version. You must manually specify the target version. For example, you have a form with two versions, v1 and v2. While installing form v2, the target version of the form might be v1 or vx, based on the availability of the form version in the target.

  You can also enter the URL of any customized .aspx page. Ensure the validity of the custom URL as the URL is not validated while installing package.

  If you create multiple activities using the same URL, ensure that you enter the same URL in the Source URL or Target URL field. Even a small change in the URL will list the activities in different rows.

Resolve Redirect URL

While installing a package, if the workflow activity contains a Redirect URL property, you can change the URL or path in the target environment. If the workflow contains a redirect URL property, the **Resolve Activity Properties** grid to configure the details of the redirect URL in the target environment appears towards the lower section of the settings form.

The grid contains the following fields:

- **Activity Name**: Displays the name of the activity that contains the Redirect URL.
- **Property Name**: Displays the property name. For example, Redirect URL.
- **Source Value**: Displays the URL or path values in the source environment. You can use the same values for the target, or change the values as required.
- **Target Value**: Displays the URL or path values in the target environment. You can use the same values as that in the source, or change the values as required. You can select multiple properties with the same URL.

  You must not change the form definition. Consider that a form is an associated artifact of a custom URL. When you install the package, the target version of the form might not be same as the source version. You must manually specify the target version. For example, you have a form with two versions, v1 and v2. While installing form v2, the target version of the form might be v1 or vx, based on the availability of the form version in the target.

  You can also enter the URL of any customized .aspx page. Ensure the validity of the custom URL as the URL is not validated while installing package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If you create multiple activities using the same URL, ensure that you enter the same URL in the Source URL or Target URL field. Even a small change in the URL will list the activities in different rows.

Resolve Template Path

While installing a package, if the workflow contains an Invoke Form activity, you can change the configuration settings of the **Template Path** property of the Invoke Form activity in the **Resolve Activity Properties** grid that appears towards the lower section of the settings form. In this grid, you can edit the path of the template file in both the source and the target environment.

The **Resolve Activity Properties** grid contains the following fields:

- **Activity Name**: Displays the name of the activity that contains the Template Path to be configured.
- **Property Name**: Displays the property name. For example, Template Path.
- **Source Value**: Displays the path values in the source environment. You can use the same values for the target, or change the values as required.
- **Target Value**: Displays the path values in the target environment. You can use the same values as that in the source, or change the values as required. Ensure to provide a valid template path that contains the valid files.

Resolve Workflows with Database Activity

When a workflow contains a database activity, the **Database Activity** grid displays the auto generated SQL scripts for the database activity. The SQL script can be a query, a function, or a procedure. If the workflow contains multiple database activities, multiple scripts are generated based on the associated artifact (for example, a database connection) of the database activity. While installing the package in the target environment, the SQL script is run first and then the package is installed.

While adding the Workflow with Database Activity to a package template, if you select the **Include the automatically generated SQL Script** option as *Yes*, the automatically generated SQL script is included in the package template. If you select the **Include the automatically generated SQL Script** option as *No*, the automatically generated SQL script is not included in the package template.

Use the **Install** tab or the **Update** tab in the **Settings** form to set the configurations for the Database Activity.

The **Database Activity** grid contains the following fields:

- **Activity Name**: Displays the database activity name.
- **SQL script**: Displays the auto-generated SQL script for the particular database activity.

  The SQL script can be a query, a function, or a procedure. You can append any additional scripts at the end of the auto-generated SQL script. The SQL script in the query window can be packaged from one SQL server machine and can be installed in another SQL server machine only.

  Note: The query generated using the Expression Editor is not considered while generating the SQL Script automatically.
- **Validate SQL script**: Validates the script for the usage of the restricted SQL commands, but does not validate syntax of the SQL query in the script. Ensure to enter error free SQL script.

  The following SQL commands are restricted:
  - Delete
  - Truncate

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Drop
- Update
- Create User
- Grant
- Revoke
- Execute

If the database activity contains a *query*, the script will be generated for the tables in the query. Parent tables and child tables are not considered. Such a query will contain the following information:

- Table structure of the tables in the query, with field name, data types, and their order
- Keys - primary key and foreign key
- Constraints - default or check, and so on
- Index

If the database activity contains a *procedure*, a script is generated for the procedure. However, objects such as tables or views used by the procedure are not generated. All such objects must be available in the target environment before package installation.

If the database activity contains a *function*, a script is generated for the procedure. If the function contains parameters you should specify the exact parameters.

The following points are not considered while generating the SQL script:

- User defined types
- Table partitioning
- Views
- Triggers
- Functions
- Synonyms
- Parent and child tables
- Query against views and functions

If you need any of the unsupported objects, then you can choose to do one of the following:

- Write and run the script manually in SQL Server
- Write and append the script to the SQL script text area

The auto-generated script contains a pre-condition statement which checks for the existing tables in the target environment. If the table exists in the target environment, the SQL script does not run. If the table does not exist in the target environment, the SQL script runs and creates the table, and adds the keys, index, and constraints as defined in the source environment.

Ensure that the object names used in the source repository does not conflict with the destination repository. If any of the objects (name of a table, key, constraint, or index) exists in the target environment, then the script execution will fail. This will be recorded in **Package History** page after the package installation.

As the script is auto-generated, we recommend not to modify the script. If you want to add your own piece of code sample to the script, you can either add it at the start of the script or append it to the end of the script. Do not add or edit the existing script.

Generating scripts for Oracle database is not supported. The SQL script text box will not be visible when

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the database in source environment is Oracle.

Note:
- If the SQL script is executed successfully, then the following message appears in the **Package History** dialog box:
The SQL script is executed successfully.

- If the SQL script execution fails, then the following message appears:
The SQL script execution failed.

- If you click the message, the **Message Details** window appears, where you can view the message details.

See Also: SQL Script Generation and Execution

Resolve Script Activity References

While adding a workflow to a package template, if the workflow contains Script Activity, the **Resolve Assembly Path** grid appears in the workflow settings form.

While installing a package, you can use the **Resolve Assembly Path** grid to resolve the assembly file path in the target environment.

The custom assembly files (DLLs) referenced in the workflow are automatically identified and appear in the package settings form.

The **Resolve Assembly Path** grid contains the following fields:

- **Assembly Name [Action Name]**: Displays the assemblies used in a workflow and the actions in which the assemblies are referenced. If one or more DLLs reference the same path, then the DLLs are displayed as comma separated list with their respective action name in the **Assembly Name [Action Name]** fie.
- **Source Assembly Path**: Displays the referenced DLL paths in the source environment.
- **Target Assembly Path**: Displays the referenced DLL paths in the target environment. The DLLs are copied to the same path in the target environment, as mentioned in the **Source Assembly Path**. To copy the DLLs to a different location in the target environment, enter the path in the **Target Assembly Path**.

  You can also specify the target assembly path of your choice in the **Target Assembly Path** field. In this case, ensure that the target assembly path you specify is valid and the files exist in it before you install the package. If the files do not exist in the path, then manually copy the files to the target environment .

  If the target assembly path is not valid or the files do not exist in it, then the workflow is installed in draft status (with the invalid assembly path) and you will not be able to publish the workflow. The following message appears in the pre-install validation:

  Files referenced in script activity do not exist (Workflow cannot be published) : <target assembly path\DLL name>

While entering the assembly path, ensure to follow the scripting guidelines as mentioned in Script - References topic.

While packaging a workflow with script activity, the package recognizes the DLLs or assemblies referenced in the script activity. However, these DLLs or assemblies are not packaged. Add these assemblies to the **Additional Package Files** to package the assemblies also in the package template. Using the additional package files, you can also install the assemblies referenced in the script activity along with the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Resolve Custom Dependent Files for Script Activity

You can install additional files in the target environment.When the package is installed, these files are copied to the installed path of the target environment. In the **Source Files** field, enter the source file path of these files.

Click **Add row** to add multiple rows to install additional files in the target environment.

If the source folder is relative to the AVEVA Work Tasks installed path, and the folder does not exist in the target environment then a new folder is created and the files are copied to it. If the source folder is not relative to the AVEVA Work Tasks installed path then the files are copied to the *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**bin* folder.

The **Install in Global Assembly Cache** field is optional and is used to install the custom dependent files in the Global Assembly Cache of the target environment. Check the **Install in Global Assembly Cache** box, if you want to install the files in the Global Assembly Cache of the target environment.

Note: At present, you must manually add the DLLs referenced in the Script Activity to the package using the **Additional Package Files** option under the configuration tree view at the left hand side of the **Add to Package Template** window.

Resolve User or Role Based Exception Template

While installing a package, if the Start activity contains Exception Template property, the following error message appears:

User or Role Based Exception Template Does Not Exist

In a start activity, consider that you have configured an exception with exception template for a workflow in the source environment. When you try to install the package in the target environment, an exception message appears in the logger if the exception template is not available in the target environment.

Create the required exception files in the following path of the target environment and install the package:

*[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**WorkflowElements**Default**en-US**Templates**ExceptionTemplate*\

Resolve Delivery Channels

While installing a package or adding workflow to a package template, if the workflow activity contains any Human Activity, you can edit the details of the delivery channels. In such cases, the **Resolve Activity Properties** grid to configure the details of the delivery channels in the target environment appears towards the lower section of the settings form.

The **Resolve Activity Properties** grid contains the following fields:

- **Activity Name**: Displays the name of the activity that contains the human activity.
- **Property Name**: Displays the property name.
- **Source Value**: Displays the URL or path values in the source environment. You can use the same values for the target, or change the values as required.
- **Target Value**: Enter the URL or path values in the target environment. You can use the same values as that in the source, or change the values as required. You can select multiple properties with the same URL.

You cannot package the user or role access rights.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

When you add an instance of AspxForm to a Work Item List control, provide valid URI in the **Aspx File Relative Uri** field. Package does not validate any relative URI of the ASPX file.

While installing a package, if the workflow activity contains a Custom Document View URL property, you can change the URL or path in the target. If the workflow contains a custom document view URL, the Resolve Activity Properties grid to configure the details of the URL in the target environment appears towards the lower section of the settings form.

Resolve XML Variables

While installing a package or adding a workflow to the package template, if you create Xml variables from the Start activity, the **Xml Forms Data Grid** and the **Xml List** variable types are resolved.

Xml Forms Data Grid

While creating an XML variable of type XML Forms Data Grid, the data grid is identified as an associated artifact.

Xml List

While creating an XML variable of type XML List, only the list created using **Manage List** option is identified as an associated artifact.

The following XML variables do not require corrections:

- Xml Iterator
- Xml List Iterator
- Xml Document

Resolve Custom Activities

When you package a workflow with a custom activity, the DLL, image, and help files automatically identified appear as dependent files. For other files, manually add them by clicking on **Add Row**. These files are copied to the installed path of the target environment when the package is installed.

If the workflow activity contains a custom activity, the **Custom Activities File Details** grid to configure the details of each file appears towards the lower section of the settings form. Use the **Edit File Location** button to configure the file details. The **Custom Activities File Details** grid replaces the settings form.

The **Custom Activities File Details** grid contains the following fields:

- **Source**: The **Source** fields display the values in the source environment.
- **Install in Global Assembly Cache:** The **Install in Global Assembly Cache** field is optional and is used to install the signed DLL in the GAC. When you install the package, clear the **Install in Global Assembly Cache** check box if the signed DLL is not required in the GAC.
- Click **Add Custom Activities File Details +** to add a new row for adding additional files which will be installed in the target environment. When you add a new row to include a signed DLL that is required in the GAC, select the **Install in Global Assembly Cache** check box.
- Click the **Delete Custom Activities File Details** icon to deleted the selected row. The dependent files will not be installed in the target environment.

Note:

- When you install a package with workflow containing custom activities again (that is in **Update** mode) but with

---

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

modifications to custom activity files, the modifications are not reflected in the target environment. In such cases, create a new Package Template and then install the same.

- When you uninstall a package with workflow containing custom activities, the custom activity files will not be removed.

- When you generate a package with workflow containing custom activities, the context-sensitive help for the properties (not the activity) will not be included in the Package Template. In such cases, manually add (using **Add Row** property) the context-sensitive help files, if required.

- The files available to add using the **Add Row** property must be relative to the AVEVA Work Tasks installation path as you cannot add files outside of the installation path.

- If the absolute path mentioned in action.xml is the installed path then the files are copied in the target environment.

- If the absolute path mentioned in action.xml is not the installed path then the files are not copied in the target environment. The files are copied to the Global Assembly Cache. You must manually copy the files to the path mentioned in the action.xml.

To correct custom activity during package design or install, see Correct Custom Activities.

Correct Custom Activities

When you design or install a package for custom activity with workflow, follow the steps given below if you need to correct the custom activity in the target environment.

1. Run the following SQL query in the SKAddInProviders table for the target repository database.

```
INSERT INTO SKAddInProviders (Type,[Name],ClassName,Assembly) VALUES
('PackageWorkflow','SampleApproval','SampleApprovalActivity.SampleApprovalCorrection'
,'bin\SampleApprovalActivity.dll')
--'SampleApproval' is the name of the custom activity.
--'SampleApprovalActivity.SampleApprovalCorrection' is the namespace of the custom
activity.
--'SampleApprovalActivity.dll' is the DLL in the BIN path of the target environment.
```

2. Implement **IWorkflowCorrection** interface in your custom activity by referring to the following DLLs, if you want to correct the custom activity in the target environment.

```
Skelta.Package.AdvancedPackageObjects;
Skelta.Forms.Core;
```

Example

```
SampleApprovalCorrection.cs
using Skelta.Package.AdvancedPackageObjects;
using Skelta.Package.Common;
using Skelta.Repository.List;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using System.Xml;
using Workflow.NET;
namespace SampleApprovalActivity
{
public class SampleApprovalCorrection: IWorkflowCorrection
{
public Skelta.Package.Common.WorkflowResponseObject
GetAssociatedObjects(Skelta.Repository.List.IListItem item,
Skelta.Forms.Core.Controls.BaseForm settingsForm, string settingsFormXml,
Workflow.NET.IAction action)
{
// Employee is the list name.
string listName = "Employee";
var responseObject = new WorkflowResponseObject();
var associatedListItemObjectsList = new List<CommonLib.PackageObjectInfo>();
// Access list item related information from the property of the activity. For example,
the list item name.
string listItemName = GetValueFromProperty(action);
if (!string.IsNullOrEmpty(listItemName))
{
var repositoryName = item.List.Application.ApplicationName;
var listDataHandler = new ListDataHandler(repositoryName, listName);
IListItem associatedListItem = listDataHandler.GetListItem(listItemName);
if (associatedListItem != null)
{
var objectInfo = this.GetBasicInfoOfListItem(associatedListItem);
if (!associatedListItemObjectsList.Contains(objectInfo))
{
associatedListItemObjectsList.Add(objectInfo);
}
}
}
else
{
throw new ArgumentException("ListItem cannot be null.");
}
responseObject.SettingsFormXml = settingsFormXml;
responseObject.AssociatedObjects = associatedListItemObjectsList;
return responseObject;
}
public Skelta.Package.Common.WorkflowResponseObject
PreinstallValidate(Skelta.Package.Deploy.IManagePackage package, string repositoryName,
Skelta.Package.Common.CommonLib.PackageObjectInfo selfDetails,
Skelta.Forms.Core.Controls.BaseForm settingsForm, string settingsFormXml, string
activityDefinition, string actionName)
{
throw new NotImplementedException();
}
public Skelta.Package.Common.WorkflowResponseObject
UpdateActivityDefinition(Skelta.Package.Deploy.IManagePackage package, string
repositoryName, Skelta.Package.Common.CommonLib.PackageObjectInfo selfDetails,
Skelta.Forms.Core.Controls.BaseForm settingsForm, string settingsFormXml, string
activityDefinition, string actionName)
{
// Replace GUID and Return XML.
WorkflowResponseObject responseObj = new WorkflowResponseObject();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Workflow.NET.Log log = new Workflow.NET.Log();
XmlDocument xmlMaindoc = new XmlDocument();
xmlMaindoc.LoadXml(activityDefinition);
try
{
// Correction.
string propertyName = "Custom Sample Property";
xmlMaindoc.SelectSingleNode("//properties//property[@name='" + propertyName +
"']").InnerText = "83bef27b-987f-46c1-a7b6-7da52f6a7462"; //// correct the guid based on
your requirement
}
catch (Exception ex)
{
log.LogError(ex, "Error while resolving Custom Activity Correction property " +
selfDetails.Name, repositoryName);
}
responseObj.CorrectedActivityXml = xmlMaindoc.InnerXml;
return responseObj;
}
// Access list item related information from the property of the activity. For example,
the list item name.
private static string GetValueFromProperty(IAction action)
{
string dataValue = string.Empty;
if (action != null && action.Properties.Contains("configCustomActivity"))
{
// configCustomActivity is the property name.
Workflow.NET.Property customActivityProperty =
action.Properties["configCustomActivity"];
if (!string.IsNullOrEmpty(customActivityProperty.Value.ToString()))
{
dataValue = (customActivityProperty.PropertyHandler).Value;
}
}
return dataValue;
}
// Method to get the basic information about the list item.
private CommonLib.PackageObjectInfo GetBasicInfoOfListItem(IListItem listItem)
{
var packageObjectInfo = new CommonLib.PackageObjectInfo();
packageObjectInfo.Name = listItem.Title;
packageObjectInfo.Id = listItem.Id;
packageObjectInfo.Type = listItem.List.Name;
packageObjectInfo.Version = string.Empty;
packageObjectInfo.ItemType = CommonLib.GetActualItemType(listItem.List.Name,
listItem.ItemType);
packageObjectInfo.IsFolder = listItem.IsFolder;
if (!listItem.ParentItemId.Equals(Guid.Empty))
{
packageObjectInfo.ParentItem = listItem.ParentItem.Title;
}
else
{
packageObjectInfo.ParentItem = string.Empty;
}
return packageObjectInfo;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
    }
    }
    }
```

Note:
- If custom activities are not created using AVEVA Work Tasks 2023 or later versions, then ensure you follow the steps given above to create DLLs for modifying custom activities.
- If you add a custom XML file for a custom activity, then ensure you remove the category data from the **action.xml** file.

Resolve MES XML Command Activity

While installing a package, if the workflow activity contains an **MES XML Command Activity**, you can change the **Unique Environment Name** and **Middleware Machine Name** properties in the target environment. If the workflow contains an **MES XML Command Activity**, the **Resolve Activity Properties** grid to configure the details of the **Unique Environment Name** and **Middleware Machine Name** properties in the target environment appears towards the lower section of the settings form.

The **Resolve Activity Properties** grid contains the following fields:

- **Activity Name**: Displays the name of the MES XML Command activity.
- **Property Name**: Displays the property name. For example, Unique Environment Name or Middleware Machine Name.
- **Source Value**: Displays the Unique Environment Name or Middleware Machine Name values in the source environment. You can use the same values for the target, or change the values as required.
- **Target Value**: Enter the target values for Unique Environment Name and Middleware Machine Name in the target environment. You can also select these values from the Runtime Initialization Settings form. If you want to use the values as specified in the source, then leave this field blank.

Note:
- Ensure that AVEVA Work Tasks Extension for MES is installed in the target environment.

Resolve Wait for MES Event Activity

While installing a package, if the workflow activity contains a **Wait for MES Event Activity**, you can change the **Unique Environment Name** and **Middleware Machine Name** properties in the target environment. If the workflow contains a **Wait for MES Event Activity**, the **Resolve Activity Properties** grid to configure the details of the **Unique Environment Name** and **Middleware Machine Name** properties in the target environment appears towards the lower section of the settings form.

The **Resolve Activity Properties** grid contains the following fields:

- **Activity Name**: Displays the name of the Wait for MES Event activity.
- **Property Name**: Displays the property name. For example, Unique Environment Name or Middleware Machine Name.
- **Source Value**: Displays the Unique Environment Name or Middleware Machine Name values in the source environment. You can use the same values for the target, or change the values as required.
- **Target Value**: Enter the target values for Unique Environment Name and Middleware Machine Name in the target environment. You can also select these values from the Runtime Initialization Settings form. If you

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

want to use the values as specified in the source, then leave this field blank.

Note: Ensure that AVEVA Work Tasks Extension for MES is installed in the target environment.

Resolve Start MES Transaction Activity

While installing a package, if the workflow activity contains a **Start MES Transaction Activity**, you can change the **Unique Environment Name** and **Middleware Machine Name** properties in the target environment. If the workflow contains a **Start MES Transaction Activity**, the **Resolve Activity Properties** grid to configure the details of the **Unique Environment Name** and **Middleware Machine Name** properties in the target environment appears towards the lower section of the settings form.

The **Resolve Activity Properties** grid contains the following fields:

- **Activity Name**: Displays the name of the Start MES Transaction Activity.
- **Property Name**: Displays the property name. For example, Unique Environment Name or Middleware Machine Name.
- **Source Value**: Displays the Unique Environment Name or Middleware Machine Name values in the source environment. You can use the same values for the target, or change the values as required.
- **Target Value**: Enter the target values for Unique Environment Name and Middleware Machine Name in the target environment. You can also select these values from the Runtime Initialization Settings form. If you want to use the values as specified in the source, then leave this field blank.

Note:
- Ensure that AVEVA Work Tasks Extension for MES is installed in the target environment.
- When packages are upgraded from AVEVA Work Tasks 2023 to AVEVA Work Tasks 2023 SP1, the workflows with transaction activities must be added again to the package template.
- For existing packages, whose transaction activities were configured earlier, the activity properties will not be automatically resolved. These activities will be displayed as custom activities, but their package functionalities will remain the same.

Resolve Commit MES Transaction Activity

While installing a package, if the workflow activity contains a **Commit MES Transaction Activity**, you can change the **Unique Environment Name** and **Middleware Machine Name** properties in the target environment. If the workflow contains a **Commit MES Transaction Activity**, the **Resolve Activity Properties** grid to configure the details of the **Unique Environment Name** and **Middleware Machine Name** properties in the target environment appears towards the lower section of the settings form.

The **Resolve Activity Properties** grid contains the following fields:

- **Activity Name**: Displays the name of the Commit MES Transaction Activity.
- **Property Name**: Displays the property name. For example, Unique Environment Name or Middleware Machine Name.
- **Source Value**: Displays the Unique Environment Name or Middleware Machine Name values in the source environment. You can use the same values for the target, or change the values as required.
- **Target Value**: Enter the target values for Unique Environment Name and Middleware Machine Name in the target environment. You can also select these values from the Runtime Initialization Settings form. If you

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

want to use the values as specified in the source, then leave this field blank.

Note:
- Ensure that AVEVA Work Tasks Extension for MES is installed in the target environment.
- When packages are upgraded from AVEVA Work Tasks 2023 to AVEVA Work Tasks 2023 SP1, the workflows with transaction activities must be added again to the package template.
- For existing packages, whose transaction activities were configured earlier, the activity properties will not be automatically resolved. These activities will be displayed as custom activities, but their package functionalities will remain the same.

Resolve Rollback MES Transaction Activity

While installing a package, if the workflow activity contains a **Rollback MES Transaction Activity**, you can change the **Unique Environment Name** and **Middleware Machine Name** properties in the target environment. If the workflow contains a **Rollback MES Transaction Activity**, the **Resolve Activity Properties** grid to configure the details of the **Unique Environment Name** and **Middleware Machine Name** properties in the target environment appears towards the lower section of the settings form.

The **Resolve Activity Properties** grid contains the following fields:

- **Activity Name**: Displays the name of the Rollback MES Transaction Activity.
- **Property Name**: Displays the property name. For example, Unique Environment Name or Middleware Machine Name.
- **Source Value**: Displays the Unique Environment Name or Middleware Machine Name values in the source environment. You can use the same values for the target, or change the values as required.
- **Target Value**: Enter the target values for Unique Environment Name and Middleware Machine Name in the target environment. You can also select these values from the Runtime Initialization Settings form. If you want to use the values as specified in the source, then leave this field blank.

Note:
- Ensure that AVEVA Work Tasks Extension for MES is installed in the target environment.
- When packages are upgraded from AVEVA Work Tasks 2023 to AVEVA Work Tasks 2023 SP1, the workflows with transaction activities must be added again to the package template.
- For existing packages, whose transaction activities were configured earlier, the activity properties will not be automatically resolved. These activities will be displayed as custom activities, but their package functionalities will remain the same.

Resolve ArchestrA Activities

Resolving ArchestrA Wait For Event Activity

While installing a package, if the workflow activity contains an **ArchestrA Wait For Event Activity**, you can change the **Unique Environment Name** and **Galaxy Name** properties in the target environment. If the workflow contains an **ArchestrA Wait for Event Activity**, the **Resolve Activity Properties** grid to configure the details of the **Unique Environment Name** and **Galaxy Name** properties in the target environment appears towards the lower section of the settings form.

The **Resolve Activity Properties** grid contains the following fields:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Activity Name**: Displays the name of the ArchestrA Wait for Event activity.
- **Property Name**: Displays the property name. For example, Unique Environment Name or Galaxy Name.
- **Source Value**: Displays the Unique Environment Name or Galaxy Name values in the source environment. You can use the same values for the target, or change the values as required.
- **Target Value**: Enter the target values for Unique Environment Name and Galaxy Name in the target environment. You can also select these values from the Runtime Initialization Settings form. If you want to use the values as specified in the source, then leave this field blank.

Resolving AttributeReadWrite Activity

While installing a package, if the workflow activity contains an **AttributeReadWrite Activity**, you can change the following properties in the target environment.

- **Unique Environment Name**
- **authenticationusername**
- **authenticationuserpassword**
- **verifierusername**
- **verifieruserpassword**

If the workflow contains an **AttributeReadWrite Activity**, the **Resolve Activity Properties** grid to configure the details of the aforementioned properties in the target environment appears towards the lower section of the settings form. The values in the **Resolve Activity Properties** grid are displayed for the activity based on the **Attribute Operation Configuration** of the **AttributeReadWrite** activity in the workflow.

The **Resolve Activity Properties** grid contains the following fields:

- **Activity Name**: Displays the name of the AttributeReadWrite activity.
- **Property Name**: Displays the property name for the aforementioned properties.
- **Source Value**: Displays the values for the aforementioned properties in the source environment. You can use the same values for the target, or change the values as required.
- **Target Value**: Enter the target values for the aforementioned properties in the target environment. You can also select these values from the Runtime Initialization Settings form. If you want to use the values as specified in the source, then leave this field blank.

Note:
- The value entered for the **authenticationusername** property is the user account used to write the values to ArchestrA Object attributes.
Based on the **Write** type of the attribute, the user credential information varies as listed below:
- **FreeAccess**: The user account does not require any additional authentication information.
- **Operate**: The credentials of the user who is performing the 'Write' operation.
- **SecuredWrite**: The credentials of the user who is performing the secured 'Write' operation.
- **VerifiedWrite**: The credentials of the user who is performing the 'Write' operation and also the credentials of the user's supervisor.
- For the **Read** operation, the credential fields are not displayed.
- For the **ArchestrA Wait For Event Activity**, ensure that the values for event name and the object name are same in the source and target environments.
- For the **AttributeReadWrite Activity**, ensure that the values for object name, attribute name, and attribute

---

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

types are same in the source and target environments.
- ArchestrA events will not be included as a part of the package. You have to manually associate the workflow with the ArchestrA events, so that these events will be packaged along with the workflow.
- If you pass the values for the **Unique Environment Name** and **Relative Reference Object** properties at runtime through ArchestrA events by associating the workflow to the ArchestrA event in the source environment, then ensure that you perform the same in the target environment.

Resolve Wait For Logic App Alert

While installing a package, if the workflow activity contains a Wait For Logic App Alert activity, you can edit the details for each activity in the target environment. Use the **Logic App Alert Filter** grid to configure the details of the Logic App Alert activities in the target environment that appears towards the lower section of the settings form.

The grid contains the following fields:

- **Logic App Alert Filter**: Use this property to set the alert key along with the Web API URL to post from the Logic App.

  Property Type: Mandatory

  In the Logic App Alert Filter window, configure these as follows:

  - **Source**
    - **Enterprise Console**: Displays the name of the Enterprise Console site with the Server name in the Source environment.
    - **URL**: Displays the source URL, which is used by Logic App to post the data to the workflow.
    - **Authorization Token**: Displays the source Authorization Token, which is used by Logic App to pass as authorization header while posting the data using URL.
  - **Target**
    - **Enterprise Console**: Enter the name of the Enterprise Console site with the Server name in the Target environment.
    - **URL**: Enter the URL, which is used by Logic App to post the data to the workflow.
    - **Authorization Token**: Enter the Authorization Token, which is used by Logic App to pass as authorization header while posting the data using URL.

      Note: The URL and Authorization token are generated based on the selected Enterprise Console site. Hence, it is mandatory to provide target values for Enterprise Console, URL, and Authorization token fields. If you want to use the values as specified in the source, then leave these fields blank.

Resolve Call Logic App Connection

While installing a package, if the workflow activity contains a Call Logic App activity, you can change the Logic App Connection in the target environment. Use the **Resolve Activity Properties** grid to configure the details of the Logic App Connection in the target environment that appears towards the lower section of the settings form.

The grid contains the following fields:

- **Activity Name**: Displays the name of the Call Logic App Activity.
- **Property Name**: Displays the Logic App Connection field name of Logic App Connection and Data property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Source Value**: Displays the name of the Logic App Connection used in Call Logic App activity.
- **Target Value**: Enter the name of the Logic App Connection which you want to link in the target environment.

Note: The displayed property name is the Logic App Connection field name of Logic App Connection and Data property of Call Logic App activity.

Work Item Field

You can package and deploy the Work Item Field from one environment or repository to another environment or repository using the package feature in AVEVA Work Tasks.

Note: Consider, the work item fields already exist in the target environment. When installing a package having work item fields created in the earlier versions of the product, the existing work item fields in the target machine get swapped. To avoid swap the work item fields, create the new package and install it on the target machine.

To add Work Item Field to Package Template

1. In the **Process Designer for Workflow** window, point to **Settings**, and then click **Work Item Fields**.
   The **Work Item Field** dialog box appears.
2. Right-click the work item field that you want to package, and then click **Add to Package Template**.
   The **Add to Package Template** dialog box appears.

For more information about adding artifacts to Package Template, see Adding Artifacts to a Package Template.

The naming convention of the **Work Item Field** at workflow level is Work Item Field name(Workflow name).

If you change an existing Work Item Field that is already added to the package template, you can rebuild the data in the package template.

Configuration Settings for Work Item Field

The Work Item Field has a unique set of configuration settings required to install, update, and uninstall the package.

You can view and edit the configuration settings from one of the following windows:

- **Source Environment > Package Template** window
- **Target Environment > Manage Package** window

You can use the **Settings** tab to define the configuration settings. The **Associated Artifacts** tab displays the associated artifacts of the **Work Item Field** definition. The **Child Artifacts** tab displays child artifacts of the Work Item Field definition.

The Package Template configuration window contains the following tabs:

- Settings
  - Install
  - Update
  - Uninstall
- Associated Artifacts
- Child Artifacts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings tab

You can use the properties in the **Settings** tab to take appropriate actions to install, update, or uninstall a package.

Install

The configuration settings in the Install tab are referred to when installing the package.

- **Work Item Field Exists**: This property allows you to take appropriate action when the Work Item Field to be installed is present in the target environment.
  - **Do not Install Work Item Field**: Select this option if you want to use the Work Item Field which is present in the target environment.
  - **Install to overwrite existing Work Item Field**: Select this option if you want to overwrite the Work Item Field is present in the target environment with the one that you packaged.
- **Associated Artifacts Not Present**: This property allows you to take appropriate action when the associated artifacts of the Work Item Field is not present in the target environment.
  - **Install Work Item Field**: Select this option if you want to install the Work Item Field even if the associated artifacts are not present in the target environment.
  - **Do not install Work Item Field**: Select this option if you do not want to install the Work Item Field as its associated artifacts are not present.

Update

The settings in this tab are referred to when the same package is installed for the consecutive time.

- **Delete Work Item Field**: Select this option if you want to delete the Work Item Field that is present in the target environment.
  - **Yes**: Select this option to remove the Work Item Field in the target environment. On selection of this option, the **Work Item Field Is Modified** property appears.

    Work Item Field is modified: Use this property to retain or delete the Work Item in the target environment even after the package is uninstalled. This property is applicable only if **Delete Work Item Field** is set to *Yes* and the Work Item Filed is modified.
    - **Retain**: Select this option if you want to retain the Work Item Field along with the modifications, in the target environment even after the package is updated.
    - **Rollback to Original State**: Select this option if you want to roll back to the original state. All modifications that were done to the Work Item Field after it was installed in the target environment will be reverted. However, the Work Item Field will be retained in the target environment.
    - **Delete**: Select this option if you want to delete the Work Item Field from the target environment when updating the package.
  - **No**: Select this option if you want to retain the Work Item Field in the target environment. On selection of this option, the **Work Item Field Exists** option appears.

    Work Item Field Exists: Use this property to take appropriate action when the Work Item Field definition specified in the package is already present in the target environment.
    - **Do not Install Work Item Field**: Select this option if you want to use the Work Item Field that is present in the target environment.
    - **Install to overwrite existing Work Item Field**: Select this option if you want to overwrite the Work Item Field that is present in the target environment with the one present in the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Associated Artifacts Not Present: This property allows you to take appropriate action when the associated artifacts of the Work Item Field is not present in the target environment.

- **Install Work Item Field**: Select this option if you want to update the Work Item Field even if the associated artifacts are not present in the target environment.
- **Do not install Work Item Field**: Select this option if you do not want to update the Work Item Field as its associated artifacts are not present.

Uninstall

You can refer to the following settings in the Uninstall tab to uninstall a Work Item Field:

- **Delete Work Item Field**: You can use this property to decide if you want to retain or delete the Work Item Field in the target environment after you uninstall the package.
  - **Yes**: Select this option if you want to delete the Work Item Field from the target environment while uninstalling the package.
  - **No**: Select this option if you want to retain the Work Item Field in the target environment after uninstalling the package.
- **Work Item Field Is Modified**: You can use this property to decide if you want to retain, roll back, or delete the Work Item Field in the target environment after uninstalling the package. This property is displayed only if **Delete Work Item Field** is set to *Yes*. However, if **Delete Work Item Field** is set to *No*, this property is not displayed.
  - **Retain**: Select this option if you want to retain the Work Item Field with the modifications in the target environment after uninstalling the package.
  - **Rollback to original State**: Select this option if you want to roll back to the original state. All modifications that were done to the Work Item Field after it installation in the target environment will be reverted. However, the Work Item Field will be retained in the target environment.
  - **Delete**: Select this option if you want to delete the Work Item Field from the target environment when uninstalling the package.

Associated Artifacts tab

The Associated Artifacts tab displays all the associated artifacts of the selected Work Item Field.

The Work Item Field can have the following associated artifacts:

- Workflow

  The Work Item Field at workflow level considers the workflow as an associated artifact. When you package the Work Item field, the associated workflow cannot be packaged as the workflow is in draft state. You must manually package the workflow, install the workflow in target, and then install the Work Item Field.

Child Artifacts tab

Work Item Field does not have any child artifact.

Note: If the Workflow is already added to the package, it automatically takes the Work Item Field. So before installing the package, it is recommended to add the Work Item Field to the package before adding the Workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Include Child Item

If a folder is added to the package template, a child item in that folder is also added to the package template based on the **Include Child Item** option selected by the user in the Package Template window. If a child item is added to the folder after adding the folder to the package template, the child item will not be added automatically to the package template. You must then add the child item manually to the package template containing the folder. You can right-click the folder from the left hand side tree view in the package template, and click **Include Child Item** to add the child item.

To add a child item, right-click an item for which you want to add child item and then select **Show Child Item**. Click **New** and then create a child item.

The child item can also be another folder. When you add the child item, all the associated artifacts, child artifacts, and security artifacts of the child item also gets added to the package template. You can add up to top three levels of child item.

Consider that the **Include Child Artifacts** is selected as **No** while creating new package template. If you right-click on the folder and select **Include Child Item** option, then the child items get added to the package template. Similarly, associated artifacts for the child item gets added on including child item for folder even though the **Include Associated Artifacts** is selected as **No** while creating the package template.

File Distribution

The following list describes the behavior of the custom file distribution in a package module:

Note: The dll files having names starting with *SE.BPM.*, *SE.TOKENPROVIDER*, *WORKTASKS*, *WORKFLOW*, *XMLRESOURCEHANDLER.NET2* will not be included while creating the package.

- While installing the package, if the target environment contains files that are not versioned then these may not be overwritten.
- When you uninstall a package containing custom dependent files, the files will not be removed.
- When you generate a package with workflow containing custom activities, the context-sensitive help for the properties (not the activity) will not be included in the package template. In such cases, manually add (using **Add Row** property) the context-sensitive help files, if required.
- The files available to add using the **Add Row** property must be relative to the AVEVA Work Tasks installation path as you cannot add files outside of the AVEVA Work Tasks installation path.
- We recommend to package custom files from AVEVA Work Tasks relative path. When you package the custom files from a non-relative path of AVEVA Work Tasks, all package files get installed in the *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**BIN* folder.
  - While installing the package containing custom dependent files, if the source folder is relative to the AVEVA Work Tasks installed path, and the folder does not exist in the target machine, then a new folder is created and the file is copied to it. For example, if the path of the *CustomEmployee.dll* file is *C:**Program files**AVEVA**Work Tasks**Custom Folder**CustomEmployee.dll***,** then a folder named *Custom Folder* is created in the target machine and *CustomEmployee.dll* is copied to it.
  - While installing the package containing custom dependent files, if the source folder is not relative to the AVEVA Work Tasks installed path then the files are copied to *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**BIN* folder. For example, if the path of the file *CustomEmployee.dll* is *C:**Custom Folder**CustomEmployee.dll***,** then *CustomEmployee.dll* is copied to the *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**BIN* folder in the target machine.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Only strong-named assemblies can be installed into the GAC. Ensure that you select the GAC option accordingly for packages.

Note: As the installation path for AVEVA Work Tasks 2023 SP1 is different, the existing packages in the repository having custom files pointing to old path may not work as expected when the product is upgraded.

SQL Script Generation and Execution

When a package contains a Manage List with an existing custom list, a database lookup, a queue, or a database activity, while configuring the package template in source environment, a related script is auto generated and populated in the SQL script text area. You do not need to write the script or run the script manually. While installing the package in the target environment, the SQL script is run first and then the package is installed. The SQL script does not execute during updation or uninstallation of package.

You can also add customized script to the auto-generated SQL script. We recommend to add the customized scripts either at the start or at the end of the auto-generated SQL script.

In artifacts such as Database Connection, the SQL scripts are not auto-generated. You can add a SQL script, if required.

If a list exists in the target environment, the list items are not installed or tables are not updated. For example, consider that a list is created and packaged in the source environment. While packaging, the relevant SQL script is auto-generated. When this package is installed in the target environment, the SQL script is executed and the relevant tables are created. If new fields are added to the list tables in the source environment, and a new package is generated, the auto-generated SQL script is also updated with the new information. While installing (or updating) the new package in the target environment, the package verifies if the object is available in the target environment. As the list object exists from the previous installation, the newly auto-generated SQL script does not execute, but a success message appears. Though the list gets updated, the relevant tables does not get updated. You must ensure to update the tables in the target environment based on the modifications done.

Two objects with the same name, irrespective of their types, cannot exist in the target environment during package installation. Consider that a table, *Department* exists in the package template. While installing the package in the target environment, the auto-generated SQL script executes if the target environment does not contain a table with the same name, *Department*. However, if the target environment contains another object type, such as an index with the same name, *Department,* the auto-generated SQL script executes, but returns an exception.

If a package template exists in an upgraded environment, then its existing artifacts does not contain any auto generated script. In such case, you must create a new package with the existing artifacts. However, if you want to use the existing package, use the **Complete Rebuild** option to rebuild the package template. The **Complete Rebuild** option generates the script for the selected and supported artifact. So the same will be executed during the installation in the target environment . The rebuild option appears only after you modify an artifact in the existing package template.

In an upgraded environment, consider that a package is already installed in the target environment. Now, consider that a database lookup (or any other supported artifact) is added to the package and a package template is created in the source environment. Now, while updating the package in the target environment, though the package template contains the auto-generated SQL script, the script will not be executed. To overcome this behavior, you can do one of the following:

1. Add the new artifact to a new package template in the source environment, and then install it in the target environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. If the artifact is added to the existing package template, then execute the SQL script manually in the target environment.
3. Uninstall the package from the target environment and install it again with the added artifacts (not recommended).

Note:
- The following keywords are restricted. Hence, only these keywords can be validated as part of "Validate SQL Script".
- Delete
-Truncate
- Drop
- Update
- Create User
- Grant
- Revoke
- Execute

- The keyword "Alter" is restricted for DB Connection and Queue.

See Also:

- Queue
- Manage List
- Database Connection
- Lookup
- Database Activity in Workflows

Package in a Distributed Environment

Consider the following points when working in a distributed environment:

- Before installing a package with custom actions, the AVEVA Work Tasks client service should be running in all the servers which are connected to each other.
- In a distributed environment, you must ensure all the computers (servers and clients) belong to a 64-bit platform before you install a package that includes custom files.
- Ensure that the firewall allows communication on port 8863 to ensure that files or packages can be installed in the client computers.

Troubleshoot Packages

1. This installation will be aborted as another instance of the same package is being installed.

Probable Cause: This error might occur when you try to install a package, which is being installed by another user.

Solution: It is recommended to wait till the package installation is complete, before installing the same package, if required.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. When packaging artifacts, the scripts are not packaged.

Solution: Match the schema in the source and target environment match.

If the schema is invalid in the target environment, ensure that you edit the schema prior to installing the package.

3. When packaging artifacts, the physical files are not packaged.

Solution: Copy all the .aspx page, images files and all dependant files from the source environment to the package environment before installing package.

4. An artifact that was not installed by package gets when uninstalling Package.

Scenario:This behavior occurs, in case you install and uninstall an artifact through the package. After uninstalling the package create an artifact in target environment with the same name as the one uninstalled by the package. Later, when you uninstall this artifact which was manually created through package, the artifact gets uninstalled. Refer to the following illustration:

Illustration:

1. Create an artifact **[Artifact-1]** and add it to a package **[Package-1].**
2. Publish, generate, and export **[Package-1],** in the source environment.
3. Import the .awspkg file of **[Package-1],** in the target environment.
4. Install **[Package-1].** The artifact [**Artifact-1]** will be installed successfully.
5. Uninstall **[Package-1**]. **[Artifact-1**] will be uninstalled.
6. Create a new artifact **[Artifact-1]** and give it the same name as the artifact that was uninstalled by [**Package-1].**
7. Verify the behavior of the newly created artifact and package, when you try to update or uninstall **[Package-1].**
8. Update **[Package-1**]. An alert appears, intimating you that the artifact [Artifact-1] cannot be updated.

   [Workflow-1] was not updated as it is not the one installed by package, but the one created in step f.

   Now, verify the behavior of the artifact and package when [**Package-1]** is uninstalled.
9. Uninstall **[Package-1], [Artifact-1**] gets uninstalled.

   Probable Cause

   This occurs as the Package was uninstalled earlier and a backup of the artifact was taken. From the next consecutive attempt to uninstall, the artifacts will be deleted.

5. Error *Cannot delete the selected item. Error while loading the Assembly* appears when deleting a Forms Control installed through package, even though the dll file was copied to the specified path.

Cause

This error occurs if the Forms Control was installed through a package and the assembly dll is not present in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

specified path.

This error will reoccur even after you manually copy the required dll to the specified path in the target environment.

Solution

1. Copy the required dll from source environment to the target environment.
2. Perform an **IIS** reset.
3. Clear the temp folder and the cookies
4. Restart all AVEVA Work Tasks Services engine.
5. Uninstall the package to delete the forms control.

6. Event Association is deleted, even though Do not uninstall is selected in Package Configuration.

Probable Cause:

This occurs if the associated workflow is deleted.

Even a File Watcher Event Association that was not installed by the package will be deleted, if it is associated to the workflow that is being uninstalled or deleted.

Event Association with Enterprise Console at List item Level

1. While trying to uninstall an associated package item, the following message appears:

Enterprise Console Event Association will be retained.

Analysis:

Consider that you delete an associated item after installation. You then create the item manually with the same configuration as the associated item that was installed with the package. If you try to uninstall the manually created item, you get the message that the Enterprise Console Event Association will be retained.

2. While trying to uninstall a package item, the following message appears:

Enterprise Console Event Association will not be uninstalled as it was not installed by current package.

Analysis:

Consider that an Event Association exists in the target. When you update the package, the Event Association does not get overwritten. So, when you try to uninstall the updated package, the Event Association item does not get uninstalled as it is not a part of the updated package.

3. While trying to install a package item, the following message appears:

Event Association already exists and cannot be updated.

Solution:

This message appears for event associations of all types, namely Scheduler, Enterprise Console, and File Watcher. The message appears when an item with the same configuration already exists in the target. The overwrite

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

options is not applicable for an Event Association.

If you want to install the association in the package, or use the latest event association of the package, delete the event association in the target, and run the package again.

4. While trying to update a package item of the type ListItem, the following message appears:

Invalid Enterprise Console Event Association exists. Delete the invalid Enterprise Console Event Association and run the package again.

Analysis:

This message appears only if the Enterprise Console Event Association is installed by the current package, and if the Enterprise Console Event Association is of type *ListItem*.

For example, consider that you create an Enterprise Console Event Association and associate it to a form called "Employee". Package the association and install the package in the target environment with the associated form. Now, delete the associated form, "Employee", and run the package again in update mode. You can see a message that an invalid Enterprise Console Event Association already exists and it has to be deleted.

Solution:

This message appears when an associated list item is deleted without removing the event association. Delete the existing event association manually, and run the package again.

Packageable Artifacts

You can package the following list of items:

- ArchestrA Events
- Calendar
- Workflows
- BAM
- Queue Management
- Database Connection
- Logic App Connection
- Message Queue Connections
- MQTT Connections
- Event Associations
  - Event Associations of Type Scheduler
  - Event Associations of Type Enterprise Console
  - Event Associations of Type File Watcher
  - Event Associations of Type System Platform Events
  - Event Associations of Type Logic App
  - Event Associations of Type Message Queue
- File Watcher
- Folder

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Forms
- Forms Association
- Form Controls
- Global Work Item Layout
- Lookup Settings
- Manage List
- Manage View
- Navigation List
- Queue Management
- Reports
- Report Connections
- Report Templates
- Security Group
  - Security Group - Repository Level
  - Security Group - List Level
  - Security Group - List Item Level
  - Security Group with Forms as Associated Artifacts
  - Security Group with Workflow as Associated Artifacts
- Security Right
- SOA Assembly
- SOA Folder
- Web API
- Web API Source
- Work Item Field

Following are not supported for packaging:

- ArchestrA Services Event Association
- Business Activity Monitoring (BAM)
- Users

Localization

Globalization is the process of making a product multi-lingual. It involves two phases -**Internationalization** and **Localization**.

Internationalization refers to the Design phase in software development in which the software is designed to enable its use in non-US locales. In this phase, all language-related aspects of the program are separated from the code, and are embedded in **PII** (Program Integrated Information) files.

Localization relates to the implementation phase and is the process of adapting an internationalized product to a specific language, script, cultural, and coded character set environment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

AVEVA Work Tasks has been designed to facilitate localization. The **AVEVA Work Tasks Localization Guide** provides instructions to localize AVEVA Work Tasks.

Scope of Localization

You can localize all the user interfaces, messages, and all features of AVEVA Work Tasks except for the following:

1. Names of database tables and database columns
2. Engine alert messages
3. Script related messages
4. Help files
5. Features other than button names and Expression Builder tree view in the Expression Editor window
6. The font style selection option in Text Editor window for elements such as, Subject and Body
7. Any license related error messages
8. The Lookup dropdown item, " ___ out of ____ ".
9. In certain cases AVEVA Work Tasks uses inbuilt ASP functions, such as alert, webpart, or file explorer dialogs to display messages and settings. These ASP functions are browser dependent. In such cases, the button text or the property text might appear in the default browser language. Check your browser documentation for localizing the browser language settings.

Steps to Localize AVEVA Work Tasks

To localize AVEVA Work Tasks for cultures other than English (US), do the following:

1. Install your edition of AVEVA Work Tasks and activate the license by following the steps described in the *Administrator Guide*.
2. For AVEVA Work Tasks to work on an operating system with a different locale, the UI Culture for the application must be changed. Specify the following Globalization setting in the Web.Config file present in the Virtual Directory path of each site, corresponding to the application. By default, the Web.Config file is present in the following path:
   [AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\EnterpriseConsole
   If you create your own Enterprise Console, you can find the Web.Config file in the **Web** folder of your Enterprise Console.

   For example, if you are changing the culture to French, add the following entry within the **<system.web>** tag in the **web.config** file of the **Enterprise Console** site, with the **uiCulture** tag as **fr-FR**.

   ```
   <globalization culture="fr-FR" uiCulture="fr-FR" requestEncoding="utf-8"
   responseEncoding="utf-8"/>
   ```

   For more information, see the *Installing AVEVA Work Tasks Locales for Brazilian Portuguese* section in the *Administrator Guide*.
3. To achieve localization in other languages, refer http://msdn2.microsoft.com/en- us/library/ms533052.aspx for respective language codes. In case of English culture other than English (US), specify the appropriate English UICulture. For example, if your culture is English (Australian), specify the UICulture as "en-AU".

   Language Codes

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The following table 1 lists all the possible language codes used to specify various system settings.

Table 1 - Language Codes

af	Afrikaans	sq	Albanian
ar-sa	Arabic (Saudi Arabia)	ar-iq	Arabic (Iraq)
ar-eg	Arabic (Egypt)	ar-ly	Arabic (Libya)
ar-dz	Arabic (Algeria)	ar-ma	Arabic (Morocco)
ar-tn	Arabic (Tunisia)	ar-om	Arabic (Oman)
ar-ye	Arabic (Yemen)	ar-sy	Arabic (Syria)
ar-jo	Arabic (Jordan)	ar-lb	Arabic (Lebanon)
ar-kw	Arabic (Kuwait)	ar-ae	Arabic (U.A.E.)
ar-bh	Arabic (Bahrain)	ar-qa	Arabic (Qatar)
eu	Basque	bg	Bulgarian
be	Belarusian	ca	Catalan
zh-tw	Chinese (Taiwan)	zh-cn	Chinese (PRC)
zh-hk	Chinese (Hong Kong SAR)	zh-sg	Chinese (Singapore)
hr	Croatian	cs	Czech
da	Danish	nl	Dutch (Standard)
nl-be	Dutch (Belgium)	en	English
en-us	English (United States)	en-gb	English (United Kingdom)
en-au	English (Australia)	en-ca	English (Canada)
en-nz	English (New Zealand)	en-ie	English (Ireland)
en-za	English (South Africa)	en-jm	English (Jamaica)
en	English (Caribbean)	en-bz	English (Belize)
en-tt	English (Trinidad)	et	Estonian
fo	Faeroese	fa	Farsi
fi	Finnish	fr	French (Standard)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

fr-be	French (Belgium)	fr-ca	French (Canada)
fr-ch	French (Switzerland)	fr-lu	French (Luxembourg)
gd	Gaelic (Scotland)	gd-ie	Gaelic (Ireland)
de	German (Standard)	de-ch	German (Switzerland)
de-at	German (Austria)	de-lu	German (Luxembourg)
de-li	German (Liechtenstein)	el	Greek
he	Hebrew	hi	Hindi
hu	Hungarian	is	Icelandic
id	Indonesian	it	Italian (Standard)
it-ch	Italian (Switzerland)	ja	Japanese
ko	Korean	ko	Korean (Johab)
lv	Latvian	lt	Lithuanian
mk	Macedonian (FYROM)	ms	Malaysian
mt	Maltese	no	Norwegian (Bokmal)
no	Norwegian (Nynorsk)	pl	Polish
pt-br	Portuguese (Brazil)	pt	Portuguese (Portugal)
rm	Rhaeto-Romanic	ro	Romanian
ro-mo	Romanian (Republic of Moldova)	ru	Russian
ru-mo	Russian (Republic of Moldova)	sz	Sami (Lappish)
sr	Serbian (Cyrillic)	sr	Serbian (Latin)
sk	Slovak	sl	Slovenian
sb	Sorbian	es	Spanish (Spain)
es-mx	Spanish (Mexico)	es-gt	Spanish (Guatemala)
es-cr	Spanish (Costa Rica)	es-pa	Spanish (Panama)
es-do	Spanish (Dominican Republic)	es-ve	Spanish (Venezuela)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

es-co	Spanish (Colombia)	es-pe	Spanish (Peru)
es-ar	Spanish (Argentina)	es-ec	Spanish (Ecuador)
es-cl	Spanish (Chile)	es-uy	Spanish (Uruguay)
es-py	Spanish (Paraguay)	es-bo	Spanish (Bolivia)
es-sv	Spanish (El Salvador)	es-hn	Spanish (Honduras)
es-ni	Spanish (Nicaragua)	es-pr	Spanish (Puerto Rico)
sx	Sutu	sv	Swedish
sv-fi	Swedish (Finland)	th	Thai
ts	Tsonga	tn	Tswana
tr	Turkish	uk	Ukrainian
ur	Urdu	ve	Venda
vi	Vietnamese	xh	Xhosa
ji	Yiddish	zu	Zulu

4. Create a folder for the new culture under the Locales folder. For example, to setup the locale for French, create the folder fr-FR in *C:\Program files\AVEVA\Work Tasks\\Locales\fr-FR*.
5. Copy contents of the folder *C:\Program files\AVEVA\Work Tasks\\Locales\en-US* to the new folder.
6. Change the messages in all the Text files to the new language.

   For example, if the locale is French, then in Calendar.txt change the second string for all the keys. In the key, Calendar_Heading:Calendar Control, change the String "Calendar Control" to "Commande De Calendrier".

   Here, the left string "Calendar_Heading" is a Key and the second string "Calendar Control" is the value. Similarly, change the values for the following keys and verify.

   Calendar_Name: Nom De Calendrier

   Calendar_Provider: Fournisseur

   Calendar_TimeZone: Fuseau horaire

   Calendar_InheritsFrom : Hérite

   Save the changes as Unicode Encoding since the ANSI version does not support the display of special characters.
7. Stop AVEVA Work Tasks Services and reset IIS. Delete the cache files from *C:\WINDOWS\Temp\AVEVAWorkTasksTemp* or *C:\ WinNT\Temp\AVEVAWorkTasksTemp*. Delete the Framework Temp files. For example, delete the Framework64 from the *C:\WINDOWS\Microsoft.NET\Framework64\v4.0.30319\Temporary ASP.NET* folder.
8. Now restart the AVEVA Work Tasks Services and log into the EC Application. Open the Calendar module. Verify that all the text that was localized in the Calendar.text is reflected.

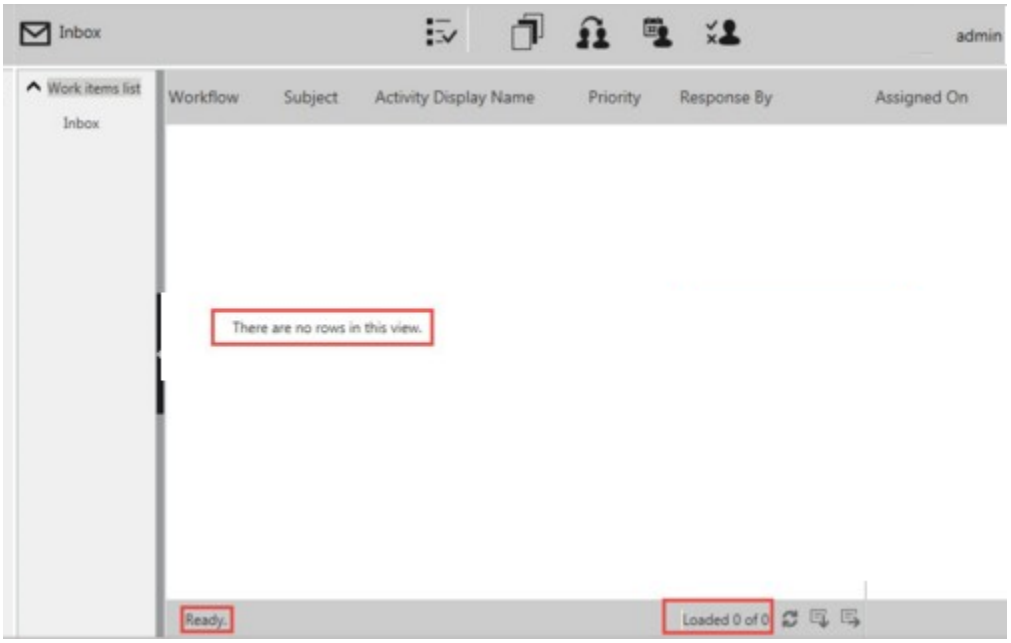
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: This change is required only for non-English cultures and need not be done for English cultures other than English (US).

9. Republish all the Forms for the new locales to get reflected.
10. To localize the user interfaces, create a locale-specific folder under the *..\WorkflowElements\Default* folder. For example, to localize user interfaces for French, create the fr-FR folder in *C:\Program files\AVEVA\Work Tasks\\Locales\fr-FR*.
11. Change the Images, JavaScripts, and StyleSheets to match the selected locale.

Steps to Localize Grid Control

In the **Grid Control**, the text highlighted in the image below can be localized using the instructions given below:



1. Follow the steps given in the link Steps to Localize AVEVA Work Tasks.
2. Refer to the following folder:

   ..\BPMUITemplates\Default\InterSoft\Grid\CommonLibrary\WebGrid\V10_0_7200\Localization

   Note: The default.xml file contains all the localization key value pairs related to the English culture. If the user configures the site in different cultures, few key value pairs might be missing. So the user will have to manually copy the missing key value pairs from default.xml to their respective culture XML file.

The value mentioned in the keys given below (Sort A to Z) are in English in other culture XML files also. If the users configures the site in different cultures, they can edit the value according to their culture.

```
<SortAscText>Sort A to Z</SortAscText>
<SortDescText>Sort Z to A</SortDescText>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to Localize Editor Control in WorkflowMemo and Wizard

Follow the instructions below to Localize Editor control in WorkflowMemo and Wizard.

1. Follow the steps given in the link Steps to Localize AVEVA Work Tasks.
2. Browse to the Telerik folder according to the theme configured to the repository. For example, if the default theme is enabled, then the Telerik folder will be inside - ..\BPMUITemplates\Default\themes\default\telerik\RadControls\Editor\Localization folder path.

   If the repository is not theme enabled, then refer to - BPMUITemplates\Default\telerik\RadControls\Editor\Localization folder path.

   Note: By default, Rad Editor localization folder contains English, French, and German culture key values. If the user configures in different cultures then the user must copy the **en-us** folder and copy it to their respective culture folders.

Steps to Localize the Escalation Email Wizard

Steps to Localize the Escalation Email wizard:

1. Follow the steps given in the link Steps to Localize AVEVA Work Tasks.
2. Change EmailBody.txt file available in **BPMUITemplates\Default\ProcessDesigner\Wizards\EmailContent\en-US**

Steps to Localize Repository Name in Enterprise Console

To Localize Repository Name in Enterprise Console

1. Follow the steps given in the link Steps to Localize AVEVA Work Tasks.
2. The repository name cannot include some characters for example certain UTF-8 and ASCII characters. To facilitate user understanding, the repository name can be localized by adding a key in the Locales folder.

Example: If the repository name is 'Human Resource' and the user wants to localize it to its Chinese equivalent '人力资源 ', the following key has to be added in the Locales folder:

SK_RepositoryDisplayName_Human Resource: 人力资源

The localized repository name will be displayed in:

1. Enterprise Console menu
2. Process Designer
3. Forms Designer

Note: The name displayed in the Database, Exception mails, and Logger Console cannot be localized. These will retain and display the original Repository name.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to Localize List Names

To Localize the List names follow the instructions given below:

1. Follow the steps given in the link Steps to Localize AVEVA Work Tasks.
2. Let us consider a user list with a list name as "Test List". The user now wants to localize the List name to Chinese language.

   In order to do this, the user has to:

   Add two keys in the Locales folder (EX: locales/en-US) in the following format:

   a. "ec_list_name_<List Name>:<Localized Name>"

   b. <List Name>:<Localized Name>

      In this scenario, it will be:

      "ec_list_name_Test List:"測試列表"

      and Test List:"測試列表"

Steps to Localize Forms

For information on localization steps, refer section *Localizing Forms* in *Forms and Lists User Guide*.

Steps to Localize Default Properties of Web Part Editor

This section gives you in-depth details of localizing the default properties of Web Part editor.

What are Default Properties

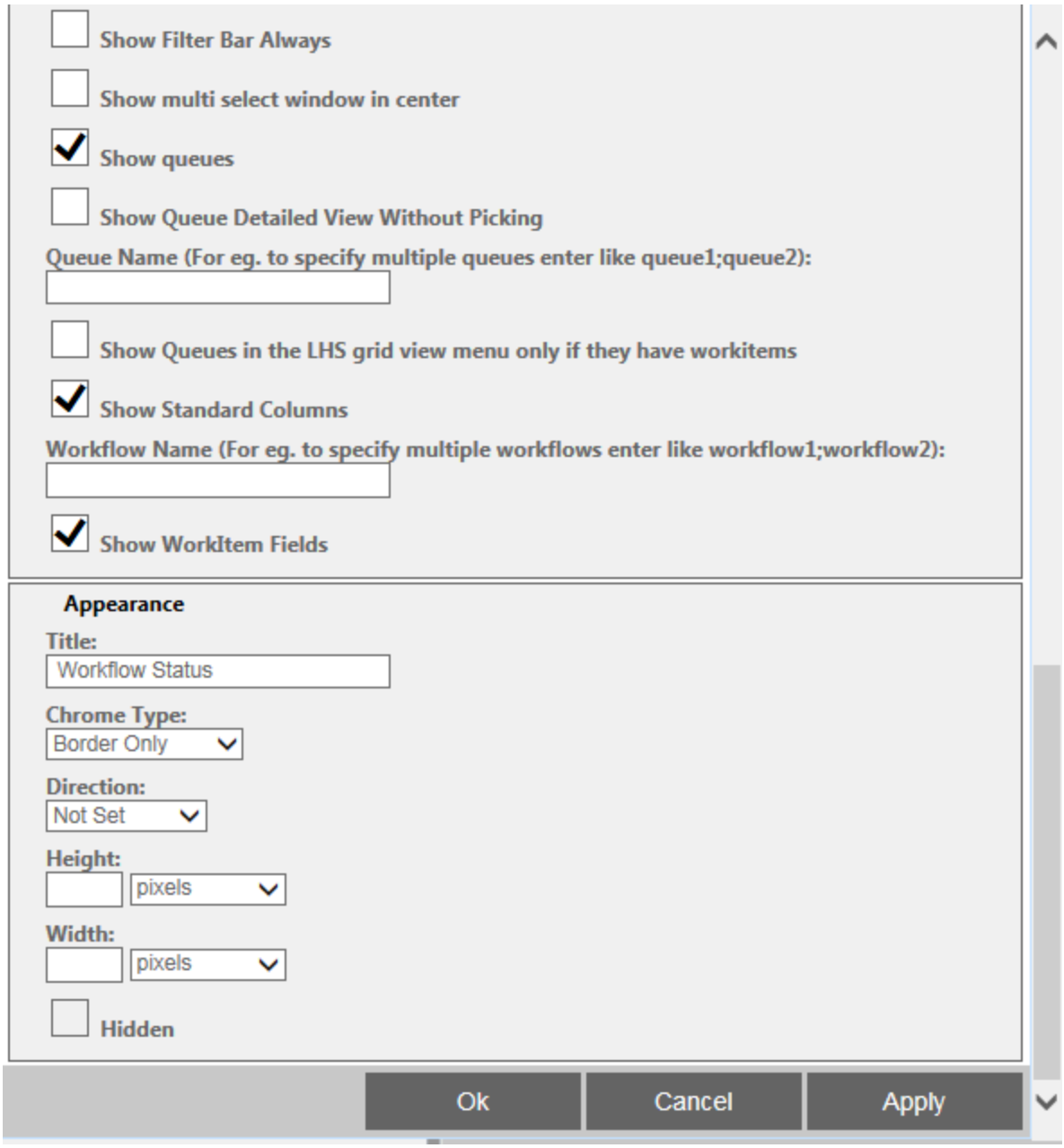
Default Properties are the properties of a web part editor which are localized internally by asp.net (Microsoft). These properties inherit the localization from the language pack specified in the web.config file.

Default Properties of Web Part Editor

Listed below are the Default Properties of the Web Part Editor:

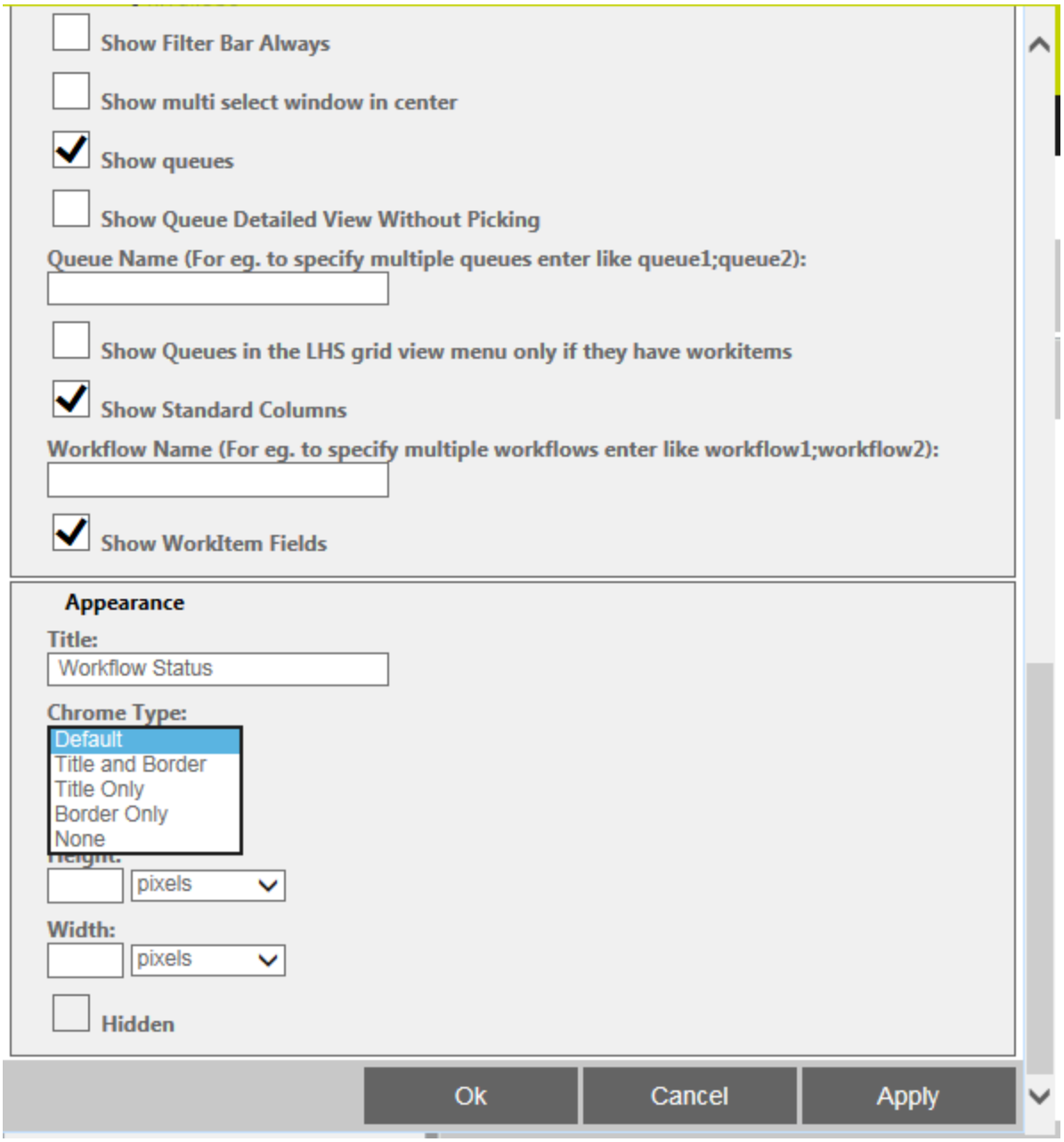
- All properties listed in the section of **Web Part Editor**, as shown in the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



- All the drop downs options of the default properties listed in the **Appearance** and **Layout** section of **Web Part Editor**, as shown in the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



1. Follow the steps given in the following link Steps to Localize AVEVA Work Tasks.
2. To localize the default properties **install the required language pack** from Microsoft web site.

Steps to Localize Provider in Role Lookup User Lookup and Assign Actor Property

In the user lookup, selecting a provider displays the localized provider property display texts in the drop down.

Pre-requisite

Follow the steps given in the following link Steps to Localize AVEVA Work Tasks.

Localizing Provider Property Display Text

The provider property display text is localized from the localization files. The key in the localization files is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

"<provider type>.<provider property display text>". If the key is found in the localization files, then the value would be picked from there; else it would use the "<provider property display text>".

For Eg: Examine the part of the data source provider XML for active directory:

```
<dataproviders>
<provider name="activedirectory" type="activedirectory">
<properties>
<property name="name" type="string"  displayname="Resource ID"
entitymappedproperty="UserName"/>
<property name="sAMAccountName" persistcolumn="UserIDString" type="string"
purpose="identifier" displayname="Account Name" entitymappedproperty="UserId"/>
:
<property name="department" type="string" purpose="department" displayname="Department"
entitymappedproperty="UserDepartment"/>
:
</provider>
</dataproviders>
```

From the above XML part, the "<provider type>" value is "activedirectory", and the "<provider property display text>" values are "Resource ID", "UserIDString", and "Department".

AVEVA Work Tasks has given localization key for two of the above property display texts from the xml part in the localization files as below:

activedirectory.Resource ID:Name

activedirectory.Account Name:Id

The bold text is the key in the localization file.

But since there is no entry for "Department" property display text, it would use the same display text, i.e., "Department".

Similar localization can be performed for provider property display texts if required.

Supported special characters for user and role lookup are:

: . / \ | } { ( ) + ' * ^ % @ ! ~ ` - _

Non-supported Localization Tags

Currently, the following sections are not localized due to internal purposes:

- Process Execution View
- ExpressionBuilder: Tree view structure
- Exception Emails. The message content in the Exception Emails have not been localized. However, other mail headings have been localized.
- All the list names in the List Name drop-down are not localized.
- Security Group Names: Except the Security Group Names, every other text in this screen is localized.
- XMLVariable Schema and tool tips are not localized.
- Tool tip for Calendar
- Tool tip for Time Picker
- Time picker heading
- Repository and Authentication Provider names in Enterprise Console Repository List page

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Localization Folders

The following folders in AVEVA Work Tasks contain localization elements:

1. Locales folder
2. Workflow Elements Folder

Locales Folder

The **<AVEVA Work Tasks Installation Folder>\Locales** folder is installed when you install AVEVA Work Tasks. This folder contains folders specific to the locale configured for the AVEVA Work Tasks installation. The default locale is "en-US" with the default path being *C:\Program files\AVEVA\Work Tasks\\Locales\en-US*.

Workflow Elements Folder

The **<AVEVA Work Tasks Installation Folder>\WorkflowElements\Default** folder is installed when you install AVEVA Work Tasks. This folder contains folders with images, JavaScript files, and Style Sheets used to render the AVEVA Work Tasks user-interface.

In addition, this folder also contains locale-specific folders that are used to localize the Images, JavaScript files, and Style Sheets for the AVEVA Work Tasks installation. The default locale folder is "en-US" with the default path being *C:\Program files\AVEVA\Work Tasks\\WorkFlowElements\Default\en-US*.

The en-US folder contains three sub-folders for Images, JavaScripts, and StyleSheet that contain the corresponding user-interface elements for the selected locale.

List of Keys to be Localized for Work Item List Control

The following are list of keys to be localized for Work Item List control:

1. **ec_newHeader** - EnterpriseConsole.txt
2. **ec_Repository** - EnterpriseConsole.txt
3. **ec_Provider** - EnterpriseConsole.txt
4. **ec_Home** - EnterpriseConsole.txt
5. **Subject** - Activities.txt
6. **Manage Work Item** - Activities.txt
7. **Workflow** - Strings.txt
8. **Activity Display Name** - WorkItemList.txt
9. **Response By** - WorkItemList.txt
10. **Assigned On** - WorkItemList.txt

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

11. **Take Ownership** - NewKeysBPM.txt
12. **Priority** - Propertypage.txt
13. **Go** - Activities.txt
14. **Approve** - Activities.txt
15. **Reject** - Activities.txt
16. **Comments** - Activities.txt
17. **Execution Details** - NewKeysBPM.txt
18. **Workflow Instance Id** - NewKeysBPM.txt
19. **Execution Details Id** - NewKeysBPM.txt
20. **Work Item Id** - NewKeysBPM.txt
21. **Activity** - NewKeysBPM.txt
22. **Data Submitted** - NewKeysBPM.txt
23. **Application** - Pdmenu.txt
24. **Show queue** - NewKeysBPM.txt
25. **Show alternate actor** - NewKeysBPM.txt
26. **Show resource calendar** - NewKeysBPM.txt
27. **Set preferences** - NewKeysBPM.txt
28. **Enable Multi Select** - NewKeysBPM.txt
29. **Disable Multi Select** - NewKeysBPM.txt
30. **Work Items list** - WorkItemList.txt
31. **Inbox** - WorkItemList.txt
32. **Worklist Folders** - WorkItemList.txt
33. **Save Layout** - WorkItemList.txt
34. **Delete Saved Layout** - WorkItemList.txt
35. **Enter the view name** - NewKeysBPM.txt
36. **View All Users** - NewKeysBPM.txt
37. **Ownership Taken?** - NewKeysBPM.txt
38. **Resource List** - Activities.txt
39. **Resource Name** - Activities.txt
40. **Action Taken** - Activities.txt
41. **Ownership** - NewKeysBPM.txt
42. **Yes** - NewKeysBPM.txt
43. **Hold** - NewkeysBPM.txt
44. **Reassign** - NewkeysBPM.txt
45. **Release** - NewkeysBPM.txt
46. **Delegate Work Item** - NewkeysBPM.txt
47. **Ownership** - NewkeysBPM.txt
48. **WorkList Layout Customization** - NewkeysBPM.txt

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

49. **AVEVA Work Tasks** - NewkeysBPM.txt
50. **Work item delivery preferences** - NewkeysBPM.txt
51. **Personal Work item delivery preferences**- NewkeysBPM.txt
52. **NOTE Work Items will be delivered to only Unchecked channels.NOTE** - NewkeysBPM.txt
53. **WebWorkList** - NewkeysBPM.txt
54. **MailWorkList** - NewkeysBPM.txt

Add the following two keys in NewKeysBPM.txt

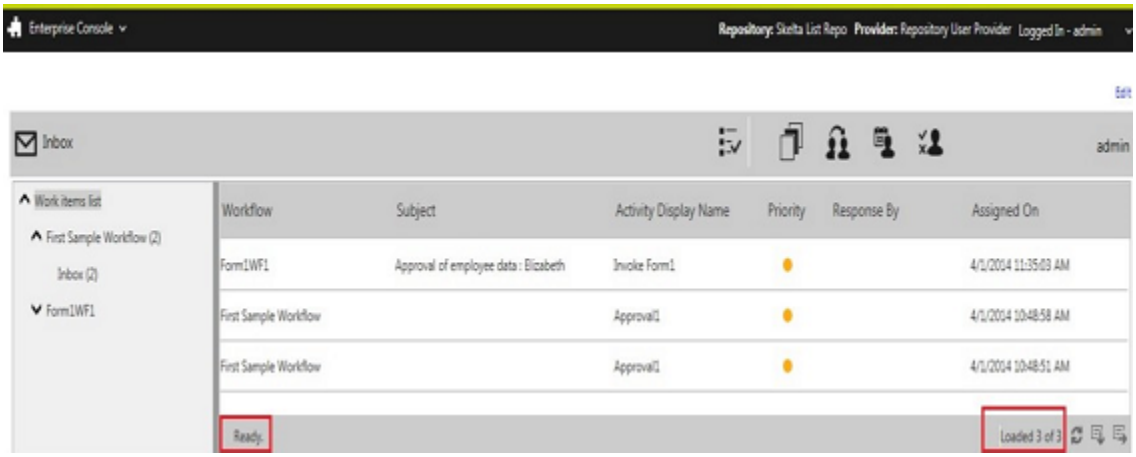
1. Layout saved successfully - NewKeysBPM.txt
2. Layout deleted successfully - NewKeysBPM.txt
3. Enable Multi Select - NewKeysBPM.txt
4. Work item list of - NewKeysBPM.txt

To set alignment for the document view, refer the file CustomDocViewDefault.html from the following path:

C:\Program files\AVEVA\Work Tasks\\WorkflowElements\Default\en-US\Activities\Activity

Keys for Localizing WebGrid Control

User can also localize WebGrids like 'Ready', 'Loading more data', 'Refreshing data'.



The keys for localizing WebGrid control are available in the following path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\BPMUITemplates\Default\InterSoft\Grid\CommonLibrary\WebGrid\V10_0_7200\Localization

Example: Suppose you want to localize the text "There are no rows in this view" in the WebGrid of Work Item page, do the following:

1. Go to [15 hive]\TEMPLATE\LAYOUTS\BPMUITemplates\Default\InterSoft\Grid\CommonLibrary\WebGrid\V10_0_7200\Localization folder.
2. Take a copy of **default.xml** file.
3. Rename this file to **wglang_en-US.xml** and change the file as shown in the following workflow.

   Before localizing

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<?xml version="1.0" encoding="utf-8" ?>
- <Localization culture="default" description="Default Language">
  - <CommonText>
      <Ready>Ready.</Ready>
      <More>Loading more data...</More>
      <LoadGroup>Loading group...</LoadGroup>
      <Sort>Sorting column...</Sort>
      <Group>Grouping column...</Group>
      <Ungroup>Ungrouping column...</Ungroup>
      <ChangeGroup>Changing group column...</ChangeGroup>
      <FilterText>Filtering column...</FilterText>
      <LoadChild>Loading child rows...</LoadChild>
      <ColumnChange>Changing columns...</ColumnChange>
      <Refresh>Refreshing data...</Refresh>
      <RefreshAll>Refreshing data and structure...</RefreshAll>
      <RefreshChild>Refreshing child table...</RefreshChild>
      <AddRow>Adding new row...</AddRow>
      <UpdateRow>Updating row...</UpdateRow>
      <DeleteRow>Deleting row...</DeleteRow>
      <ReadyError>Unhandled error occurred on last action. Click for more details.</ReadyError>
      <ColumnMove>Moving column...</ColumnMove>
      <Loaded>Loaded [n] of [total]</Loaded>
      <ExportText>Exporting...</ExportText>
      <NoData>There are no rows in this view.</NoData>
      <NewRowInfo>Please click here to add new row...</NewRowInfo>
      <GroupByBoxInfo>Drag a column header here to group by that column.</GroupByBoxInfo>
      <LoadSelfRefChild>Loading self referencing childrows...</LoadSelfRefChild>
```

After localizing:

```
wglang_en-US.xml - Notepad
File  Edit  Format  View  Help
    <LoadChild>Loading child rows...</LoadChild>
    <ColumnChange>Changing columns...</ColumnChange>
    <Refresh>Refreshing data...</Refresh>
    <RefreshAll>Refreshing data and structure...</RefreshAll>
    <RefreshChild>Refreshing child table...</RefreshChild>
    <AddRow>Adding new row...</AddRow>
    <UpdateRow>Updating row...</UpdateRow>
    <DeleteRow>Deleting row...</DeleteRow>
    <ReadyError>Unhandled error occurred on last action. Click for more details.</ReadyError>
    <ColumnMove>Moving column...</ColumnMove>
    <Loaded>Loaded [n] of [total]</Loaded>
    <ExportText>Exporting...</ExportText>
    <NoData>There are no rows to be displayed in this WebGrid.....</NoData>
    <NewRowInfo>Please click here to add new row...</NewRowInfo>
    <GroupByBoxInfo>Drag a column header here to group by that column.</GroupByBoxInfo>
    <LoadSelfRefChild>Loading self referencing childrows...</LoadSelfRefChild>
    <CalendarToday>Today</CalendarToday>
    <CalendarNone>None</CalendarNone>
    <Filtered>Filtered</Filtered>
    <Unfiltered>Unfiltered</Unfiltered>
    <LoadPreviewRow>Loading preview row...</LoadPreviewRow>
    <InputRequiredStatus>One or more cells require user input.</InputRequiredStatus>
    <UpdateSuccess>Record is updated successfully.</UpdateSuccess>
    <AddSuccess>Record is added successfully.</AddSuccess>
    <DeleteSuccess>Record is deleted successfully.</DeleteSuccess>
    <ReadOnlyCell>This cell is read only.</ReadOnlyCell>
    <ModifiedCell>This cell is modified.</ModifiedCell>
    <EditedCell>This cell is being edited.</EditedCell>
    <PagingLoading>Loading page {0}...</PagingLoading>
    <PagingStatus>Page {0} of {1}</PagingStatus>
    <PagingPartialGroup>(Record {0}-{1} of {2})</PagingPartialGroup>
  </CommonText>
  <MessageBoxText>
    <NoChildRow>There are no child rows for this record.</NoChildRow>
    <InProgress>Please wait while WebGrid is in progress processing request...</InProgress>
    <UnhandledError>Unhandled exception occurred. See following for details.</UnhandledError>
    <LoadMoreException>Load more data is only available when there are no group columns in RootTable.</LoadMoreExceptio
    <AddNewRow>The new row has been edited. Do you want to add the new row?</AddNewRow>
    <UpdateRow>The selected row has been edited. Do you want to update the changed row?</UpdateRow>
    <DeleteRow>Are you sure want to delete this row?</DeleteRow>
  </MessageBoxText>
  <Tooltip>
    <Refresh>Refresh grid data</Refresh>
    <LoadMore>Load more data</LoadMore>
    <ExportText>Export grid</ExportText>
    <Help>WebGrid.NET Client Help</Help>
    <RemoveFilters>Click to remove filters from the records</RemoveFilters>
    <RestoreFilters>Click to filter the records using the last saved filter</RestoreFilters>
    <NoFilter>There is no current filter</NoFilter>
    <InputRequired>This cell requires user input</InputRequired>
    <PagingFirst>First Page</PagingFirst>
    <PagingPrev>Previous Page</PagingPrev>
```

4. Save the file.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Keys for Localizing List Names

User can also localize list names like Workflows, Users, Event Provider, Navigation, SOA Assembly Folder, BizTalk Connections, Database Connections, File Watcher, Lookup and Dynamic Lists.

The default lists have "ec_list_name_" preceded the listname

Example:

ec_list_name_workflow:Workflow

ec_list_name_Users:Users

ec_list_name_Event Providers:Event Providers

ec_list_name_Event Associations:Event Associations

ec_list_name_Forms List:Manage Forms List

These list localization keys will be available in the following path.

[AVEVA Work Tasks Installed Path]\Locales\en-US\EnterpriseConsole.txt

To localize the list name:

1. Go to *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Locales**en-US*\ path.
2. Open **EnterpriseConsole.txt** file.
3. Provide list name that you would like to give in List Names section of the **EnterpriseConsole.txt** file as shown in the following figure.

```
// List Names
ec_list_name_Workflow:Workflows
ec_list_name_NavigationDetails:Navigation List
ec_list_name_Impersonation Users:Impersonation Users
ec_list_name_Users:Users
ec_list_name_Event Providers:Event Providers
ec_list_name_Event Associations:Event Associations
ec_list_name_Forms List:Manage Forms List
ec_list_name_Initiate Form:Initiate Form
ec_list_name_Assembly List:Assembly List
ec_list_name_Folder List:Folder List
ec_list_name_Security Right:Security Right
ec_list_name_Security Group:Security Group
ec_list_name_Security Group Mapping:Security Group Mapping
ec_list_name_BizTalk Connections:BizTalk Connections
ec_list_name_SAP Connections:SAP Connections
ec_list_name_FileWatcher List:FileWatcher List
ec_list_name_Manage List:Manage List
ec_list_name_Connection List:Connection List
ec_list_name_Lookup List:Lookup List
ec_list_name_List-Form Association List:List-Form Association List
```

4. Save the file.

Example: Localize Workflows to List of Workflows

To do the same:

1. Go to *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Locales**en-US*\ path.
2. Open **EnterpriseConsole.txt** file.
3. Provide list name as 'List of Workflows' in List Names section of the **EnterpriseConsole.txt** file as shown in the following figure.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Before localizing

```
// List Names
ec_list_name_Workflow:Workflows
ec_list_name_NavigationDetails:Navigation List
ec_list_name_Impersonation Users:Impersonation Users
ec_list_name_Users:Users
ec_list_name_Event Providers:Event Providers
ec_list_name_Event Associations:Event Associations
ec_list_name_Forms List:Manage Forms List
ec_list_name_Initiate Form:Initiate Form
ec_list_name_Assembly List:Assembly List
ec_list_name_Folder List:Folder List
ec_list_name_Security Right:Security Right
ec_list_name_Security Group:Security Group
ec_list_name_Security Group Mapping:Security Group Mapping
ec_list_name_BizTalk Connections:BizTalk Connections
ec_list_name_SAP Connections:SAP Connections
ec_list_name_FileWatcher List:FileWatcher List
ec_list_name_Manage List:Manage List
ec_list_name_Connection List:Connection List
ec_list_name_Lookup List:Lookup List
ec_list_name_List-Form Association List:List-Form Association List
```

After localizing

```
// List Names
ec_list_name_Workflow:List of Workflows
ec_list_name_NavigationDetails:Navigation List
ec_list_name_Impersonation Users:Impersonation Users
ec_list_name_Users:Users
ec_list_name_Event Providers:Event Providers
ec_list_name_Event Associations:Event Associations
ec_list_name_Forms List:Manage Forms List
ec_list_name_Initiate Form:Initiate Form
ec_list_name_Assembly List:Assembly List
ec_list_name_Folder List:Folder List
ec_list_name_Security Right:Security Right
ec_list_name_Security Group:Security Group
ec_list_name_Security Group Mapping:Security Group Mapping
ec_list_name_BizTalk Connections:BizTalk Connections
ec_list_name_SAP Connections:SAP Connections
ec_list_name_FileWatcher List:FileWatcher List
ec_list_name_Manage List:Manage List
ec_list_name_Connection List:Connection List
ec_list_name_Lookup List:Lookup List
ec_list_name_List-Form Association List:List-Form Association List
```

4. To verify this, login to the **Enterprise Console**.
5. From the **Enterprise Console** pull-down menu, select Workflows. The list name is localized from the 'Workflows' to the 'List of Workflows'.

Note: If already the application is opened, then just refresh the page to reflect the localized name.

User can follow the above steps to localize the other lists.

Note: If user has created any dynamic list then an entry should be made into *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Locales**en-US**EnterpriseConsole.txt* and "ec_list_name_" should precede the new listname.

Example: Assume that the 'Employee List' is created at the repository level. The entry 'ec_list_name_employee_list' should be placed in *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Locales**en-US**EnterpriseConsole.txt* and later you can localize this dynamic list.ec_list_name_employee_list:Employees

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

List

Localize the Response Sent for an Invalid Email to the Communication Engine

The response sent for an invalid email, for the Human Activities email or for the WaitForEmail Activity, to the AVEVA Work Tasks Communication Engine, can be localized. The value of **InvalidFilterMailMessage** key in the channel.txt file is emailed as a response to any invalid email. The channel.txt file can be found at **<AVEVA Work Tasks Install Directory>/Locales/en-US.**

The (key, value) pair can be specified in the channel.txt file as follows:

InvalidFilterMailMessage: <Localized message>

A sample of the value for this key is shown below

"System could not process your mail. Some of the key words not found in your mail to process it. Reply to the original message again."

Code Samples

For detailed code samples, refer to **AVEVA Work Tasks API Scenarios**.

Note: AVEVA Work Tasks supports Microsoft .NET Framework, accordingly build all custom codes on Microsoft .NET Framework supported version.

Restart Services

Dll REFERENCE

Workflow.NET.NET2.dll

NAMESPACE USED

```
Workflow.NET
Skelta.Core
Skelta.FarmManager
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Workflow.NET;
using Skelta.Configurations.DB;
using Skelta.Configurations.DB.Interfaces;
using Skelta.FarmManager;
using Skelta.Core;
namespace Scenarios
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
public partial class SkeltaFarmConfigurations : Form
{
// Change this connection string with actual connection string
string connectionString = "Data Source=sqlServerName;Integrated Security=True;";
string databaseType = "SQL server";
string databaseName = "SkeltaFarm";

// Settings for a Database user having sysadmin rights in SQL server instance
string databaseUser = "John";
string databasePassword = "John";
string databaseDomain = "domain1";

public SkeltaFarmConfigurations()
{
InitializeComponent();
}

#region "Restart Services"
/// <summary>
/// Restart workflow engine
/// </summary>
private void RestartEngine()
{
RestartService("SkeltaWorkflow2");
}

/// <summary>
/// Restart Advanced Server
/// </summary>
private void RestartAdvancedServer()
{
RestartService("SkeltaAdvanceServer");
}

/// <summary>
/// Restart SMTP notification
/// </summary>
private void RestartNotification()
{
RestartService("SMTPNotificationEngine2");
}

/// <summary>
/// Restart Task Schedular
/// </summary>
private void RestartTaskSchedular()
{
RestartService("SkeltaTaskScheduler2");
}

/// <summary>
/// Restarts service
/// </summary>
/// <param name="serviceName">Name of the service to be restarted</param>
private void RestartService( string serviceName )
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (ServiceMapping.IsServiceMappedForCurrentServer(serviceName))
{
ServiceMapping serviceMapping = new ServiceMapping(serviceName, Server.Local);
SkeltaClientService clientService = new SkeltaClientService();
try
{
clientService.Stop(serviceMapping);
clientService.Start(serviceMapping);
}
finally
{
clientService.Close();
}
}
else
throw new Exception("This service is not mapped to the current server");
}

#endregion
}
}
```

Execute Workflow

Sample code to execute a published workflow:

```
Workflow.NET.Engine.Client objclient = new Workflow.NET.Engine.Client("LeaveReportsRepo",
"Leave WF"); //Provide Application/Repository name and Workflow name.
int executionid = objclient.Execute("activedirectory::john", "<test>m</test>"); //Provide
user id string
objclient.Close();
```

Custom Activity

AVEVA Work Tasks provides a set of interfaces and their supporting classes and members for the user to create and implement a custom activity.

Configure Custom Activity without any Properties

Create a new XML file with the following code template and update as required.

```
<?xml version="1.0" encoding="utf-8"?>
<actionsdata>
<version number="3.0" />
<propertytypes>
<propertytype name="SampleCustomPropertyModified"
classname="Workflow.NET.SampleCustomPropertyModified.SampleProperty" assemblyname="bin\
SampleCustomPropertyModified.dll"
webuiclassname="Workflow.NET.SampleCustomPropertyModified.SamplePropertyWebUI"
webuiassemblyname="bin\SampleCustomPropertyModified.dll" />
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</propertytypes>
<category name="Custom Files" sortorder="15">
<action name="CustomActivity" displayname="Custom Activity">
<description>Custom activity. </description>
<image resourcename="bell.png"></image>
<handler classname="SampleCustomAction.CustomAction" assembly="bin\
SampleCustomAction.dll"></handler>
<properties></properties>
<return>
<value helpstring="Success">Success</value>
<value helpstring="UnSuccess">UnSuccess</value>
</return>
</action>
</category>
</actionsdata>
```

If the **properties** tag exists, then append to the existing properties instead of creating additional **properties** tag. Otherwise, add an empty **properties** tag to the XML file as follows.

```
<properties></properties>
```

Note: We recommend to create a new XML file instead of editing the existing XML files.

When you package a workflow with custom activity, if you need to correct the custom activity in the target repository, follow the steps mentioned in "*Resolving Custom Activities*" and "*Correcting Custom Activities*". For more information, see Resolve Custom Activities and Correct Custom Activities.

Custom Approval Activity

AVEVA Work Tasks provides a set of interfaces and their supporting classes and members that can be implemented in the class of a custom activity to develop a AVEVA Work Tasks human activity.

The following code sample shows how to create a sample Approval activity that looks and behaves like the AVEVA Work Tasks defined Approval activity.

Procedure

Dll Reference

```
System.Web
Workflow.NET.NET2
Skelta.HWS.ActivityHandler
```

Namespace Used

```
System
System.Web
System.Web.UI
System.Collections.Generic
System.Text
System.Collections
Workflow.NET.Engine
Workflow.NET.Interfaces
Skelta.Core
Skelta.HWS
Skelta.HWS.ActivityHandler
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To create a Sample Custom Approval Activity

- Go to **Start > Visual Studio 2005 > New > Projects->Class Library->SampleApprovalActivity**. This step will create a Class Library project in the path specified by the name SampleApprovalActivity.
- Create two class files under the project SampleApprovalHandler.cs and SampleApprovalControl.cs.
- In the SampleApprovalHandler.cs class file, inherit the HWSActivityProviderBase class which is the base class for all the resource activities.
- In the SampleApprovalControl.cs class file, inherit the class ActivityUIBase which while rendering the Work Item detail view. Inherit the INativeWebFormUIControl interface and it is used for setting the Current Work Item context to be used in Work Item detail view.

  SampleApprovalHandler.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using Workflow.NET;
using Workflow.NET.Storage;
using Workflow.NET.Interfaces;
using Workflow.NET.Engine;
using Workflow.NET.Engine.Interfaces;
using Skelta.Core;
using Skelta.HWS;
using Skelta.HWS.WorkListChannel;
namespace SampleApproval
{
public class SampleApprovalHandler : HWSActivityProviderBase
{
public SampleApprovalHandler()
{
}

/// <summary>
/// Resolves the property names. Existing properties can be given different names using
this API.
/// MaximumActorsRequiredForActionCompletion is used in the Skelta for getting the
property value
/// as for approval it will be Affirmative Answer for Approval [% or Number] and for
checklist it will  ///  be Affirmative Answer for CheckList [% or Number].
/// </summary>
/// <param name="propertyName">Standard property name.</param>
/// <returns>Property name set for the activity.</returns>
public override string ResolvePropertyName(string propertyName)
{
switch (propertyName)
{
case "MaximumActorsRequiredForActionCompletion":
return "Affirmative Answer for Approval [% or Number]";
case "MinimumActorsRequiredForActionCompletion":
return "Action Limit[% or Number]";
}
return base.ResolvePropertyName(propertyName);
}
/// <summary>
/// Based on decision status the action output is set.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
    /// </summary>
    /// <param name="decisionStatus">Decision status which is internal status which is
    arrived based on decision counters.</param>
    /// <param name="actionResult">Action result.</param>
    /// <returns>Action output.</returns>
    /// <remarks>
    public override string GetActionOutput(string decisionStatus, out ActionResult
    actionResult)
    {
```

```
string MappedActionOutput = "";
switch (decisionStatus.ToLower())
{
case "decisionsuccess":
MappedActionOutput = "Approved";
actionResult = ActionResult.Completed;
break;
case "decisionfailure":
MappedActionOutput = "Rejected";
actionResult = ActionResult.Completed;
break;
default:
MappedActionOutput = base.GetActionOutput(decisionStatus, out actionResult);
break;
}
return MappedActionOutput;
}


/// <summary>
/// Gets the possible status list for the activity meaning output for the activity.
/// </summary>
/// <returns>List of possible outputs.</returns>

public override Dictionary<string, string> GetPossibleStatusList()
{
Dictionary<string, string> valueAndDisplayName = new Dictionary<string, string>();
valueAndDisplayName.Add("Approved", "Approve");
valueAndDisplayName.Add("Rejected", "Reject");
return valueAndDisplayName;
}

/// <summary>
/// To set the output of the Activity as CompletionMaker or not.
/// </summary>
/// <param name="Status"></param>
/// <returns></returns>
public override bool IsCompletionMaker(string Status)
{
return true;
}

/// <summary>
/// Boolean flag to check if the status has to be alerted to engine if the output is not a
completion maker.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// </summary>
/// <param name="status">status</param>
/// <returns>true or false.</returns>
public override bool AlertCurrentStatusAsActionOutput(string Status)
{
return false;
}
}
}
```

SampleApprovalControl.cs

```
using System;
using System.Web;
using System.Web.UI;
using System.Collections.Generic;
using System.Text;
using Workflow.NET.Engine;
using Workflow.NET.Interfaces;
using Workflow.NET.Storage;
using Skelta.Core;
using Skelta.HWS;
using System.Collections;
using Skelta.HWS.ActivityHandler;

namespace SampleApproval
{
public class SampleApprovalControl : ActivityUIBase, INamingContainer,
 Skelta.HWS.WorkListChannel.Web.INativeWebFormUIControl
{
public HWSActivity _HWSActivity;
public WorkItem _WorkItem;
Workflow.NET.Engine.Context WorkflowContext;
Workflow.NET.Log logger = new Workflow.NET.Log();
private Workflow.NET.Parser parseHtml = new Workflow.NET.Parser();
string strTempDirectoryPath = "", strDirectoryPath = "";
Workflow.NET.ActionsData _ActionsData;
/// <summary>
/// Initializes a new instance of the Sample  ApprovalControl class.
/// </summary>
public SampleApprovalControl()
{

}


/// <summary>
/// This method is used to render the Workitem Detail view
/// </summary>
protected override void CreateChildControls()
{
string InternalStatus = _WorkItem.Status.ToUpper();
string ErrorMessage = "";
string ActionTaken = "";
string Comment = Context.Request["txtComment"];
if (Comment != null)
Comment = Comment.Trim();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//Getting the value of the button click
if (Context.Request["btnApprove"] != null)
{
ActionTaken = "Approve";
}
else if (Context.Request["btnReject"] != null)
{
ActionTaken = "Reject";
}

if (ActionTaken != "")
{
try
{
//Submitting the work item with the respective output
SynchronousActionData syncData = new SynchronousActionData(HttpContext.Current);
if (ActionTaken == "Approve")
 {
_WorkItem.Submit("WebWorkList", "Approved", "", Comment, syncData);
 }
else
 {
 _WorkItem.Submit("WebWorkList", "Rejected", "", Comment, syncData);
 }
}
catch(Exception ex)
{
logger.LogError(ex,"Error while performing Approval Action" + ex.Message.ToString());
ErrorMessage = ex.Message.ToString();

}
//RefreshWorkItemList() is the commonfunction which is used for Refreshing
//the Workitem List after submitting the work item
this.Controls.Add(new LiteralControl("<script>RefreshWorkItemList();</script>"));

 }
 if (InternalStatus == "AW"|| InternalStatus == "OW" || InternalStatus == "HO")
 {
 //Error message is rendered below the notes.
 base.ErrorMessage = ErrorMessage;
 //This is called for Rendering a common UI in Workitem Detail View Which is displaying
Subject,Notes and Manage Workitem drop down and also
 // IFrame which will display the Custom Document view in Left hand side of Details view.
 RenderCommonActivityUI(parseHtml);
 //Rendering the Detail View UI(SampleApproveUI.htm)
 this.Controls.Add(new
LiteralControl(parseHtml.CheckAndProcessFileGetString(strDirectoryPath
 + "Activities\\Activity\\SampleApproveUI.htm", strTempDirectoryPath + "cache",
"Activities\\Activity\\")));
 }
}
protected override void Render(HtmlTextWriter writer)
{
this.RenderBeginTag(writer);
this.RenderChildren(writer);
this.RenderEndTag(writer);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}

/// <summary>
/// This method Provides information of the current work item instance.
/// </summary>
/// <param name="workItem">Current Workitem Instance</param>
/// <param name="channelName">Channel Name(WebWorkList)</param>
/// <param name="formName">Form Name(Approval Form)</param>
void Skelta.HWS.WorkListChannel.Web.INativeWebFormUIControl.SetContext(WorkItem workItem,
string channelName, string formName)
{
_WorkItem = workItem;
_HWSActivity = workItem.HWSActivity;
WorkflowContext = workItem.CurrentContext;
_ChannelName = channelName;
_FormName = formName;
strTempDirectoryPath =
workItem.HWSActivity.CurrentContext.ProcessDefinition.ActionsDefinition.DirTemp;
strDirectoryPath =
workItem.HWSActivity.CurrentContext.ProcessDefinition.ActionsDefinition.DirSourceElements;
_ActionsData = workItem.CurrentContext.ProcessDefinition.ActionsDefinition;
base.Initailize(this.Context, workItem);
 }
}
}
```

- Build the Class Library Project and place the assembly in the SkeltaInstalledPath\BPM.Net\bin folder.
- Create an HTML Page by the name SampleApproveUI.htm in the SkeltaInstalledPath\BPM.NET\ WorkflowElements\Default\en-US\Activities\Activity\ folder which is getting used for rendering the UI. Please copy the contents below in the created HTML page.

```
TABLE WIDTH="100%" height="100%" BORDER="0" CELLPADDING="2" CELLSPACING="1"
bgcolor="#ffffff">
 <%#errorMessage%>
 <TR VALIGN="TOP" class="righttdbg">
 <TD align="left" width="<%#iframewidth%>" bgcolor="<%#iframebgcolor%>">
 <%#iframeui%>
 </TD>
 <TD VALIGN="TOP">
 <TABLE WIDTH="100%" BORDER="0" CELLSPACING="1" CELLPADDING="4" >
 <TR>
 <TD align=left class="subtitle">
 <strong>@@Comments@@</strong>
 </TD>
 </TR>
 <TR>
 <TD align="left">
 <TEXTAREA name='txtComment' cols=35 rows=8 wrap="VIRTUAL"
class="inputtextarea"></TEXTAREA>
 </TD>
 </TR>
 <!--<TR>
 <TD align="center"> </TD>
 </TR>-->
 </TABLE>
 </TD>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 </TR>
 <TR VALIGN="TOP">
 <TD> </TD>
 <TD height="25" align="right" bgcolor="#ffffff">
 <input  type='SUBMIT'  class="inputbutton" value='@@Approve@@' name='btnApprove' />
    
 <input  type='SUBMIT' class="inputbutton" value='@@Reject@@' name='btnReject' />
 </TD>
 </TR>
</TABLE>
```

- Configure the Actions.Xml which will be in Skeltainstalledpath\BPM.NET\WorkflowElements\Default\en-US\Actions\XML for the Created Sample Approval under the Custom Activities Category. While configuring the Custom Activity, we need to specify the image filename of PNG type of size 40x40 for the custom activity which should be in the following folders. [SkeltaInstalledPath]\BPM.NET\WorkflowElements\Default\en-US\Actions\Images\Normal\Large, [SkeltaInstalledPath]\BPM.NET\WorkflowElements\Default\en-US\Actions\Images\Selected\Large,[SkeltaInstalledPath]\BPM.NET\BPMUITemplates\Default\ActivityImages.

Note: If an XML exception occurs after copying the below configuration in Actions.XML, open the actions.xml file in the browser and remove the extra spaces if any.

```
<action name="SampleApproval" helptemplate="ApprovalActionHelp.html"
helptemplatewinheight="450" helptemplatewinwidth="600">
 <description>SampleApprovalDescription</description>
 <image resourcename="Approval.png"></image>
 <handler classname="SampleApproval.SampleApprovalHandler" assembly="bin\
SampleApprovalActivity.dll"></handler>
 <visual-attributes>
 <shape>roundedrectangle</shape>
 <backcolor>lightgreen</backcolor>
 <width>50</width>
 <height>50</height>
 </visual-attributes>
 <report rendertype="xsl" xsl="ActionDescription.xsl">
 <handler class="Workflow.NET.Report.ReportHandler" assembly="bin\
Workflow.NET.ReportHandler.NET2.dll" />
 <messages>
 <message id="ActionStarted">
 <![CDATA[Action <%#ActionName%> of type <%#ActionType%> initiated at
 <script Language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.
 ]]>
 </message>
 <message id="ResourceScarcity">
 <![CDATA[ Action <%#ActionName%> of type <%#ActionType%> completed execution due to
 lack of actor.<%#ActionName%> completed with the output <%#ActionOutput%>.
]]>
 </message>
 <message id="FilteredResourceCount">
 <![CDATA[
 The <%#ActionType%> action <%#ActionName%> selected <%#ResourceCount%>
 actor<%#ResourceOrResources%>.
 ]]>
 </message>
 <message id="RequestResource">
 <![CDATA[
 The actor named  <%#ResourceName%> with identifier <%#ResourceIdentifier%>
 has been identified to take part in the <%#ActionName%> action.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 ]]>
 </message>
 <message id="ResourceUnavailable">
 <![CDATA[
 The actor <%#ResourceName%> with identifier <%#ResourceIdentifier%> and email
<%#ResourceEmail%> is unavailable.<br>
 The 'Ignore unavailable actor' property for the action was set to
<%#IgnoreUnavailableResource%>.<br>
 The 'Alternate actor allowed' property was set to <%#AlternateResourceAllowed%>.
 ]]>
 </message>
 <message id="AvailableResourceCount">
 <![CDATA[
 The action <%#ActionName%> request was sent to <%#ResourceCount%>
actor<%#ResourceOrResources%>.<br>
 The Minimum No.Of ownerships required is <%#PropertyMin. Acknowledgements [% or
Number]()%>.<br>
 The Maximum No.Of ownerships is
<%#PropertyMax. Acknowledgements [% or Number]()%>.<br>
 The Minimum human activity count required  is <%#PropertyAction Limit[% or
Number]()%>.<br>
 The Approval count required for the action to be approved is <%#PropertyAffirmative
Answer for Approval [% or Number]()%>.<br>
 ]]>
 </message>
 <message id="TimeOutSet">
 <![CDATA[
 Timeout <%#TimeOutType%> scheduled at <script
language='javascript'>WriteClientTime('<%#TimeOutValue%>',1);</script>.
]]>
 </message>
 <message id="TimeOutWarningSet">
 <![CDATA[
 Timeout <%#TimeOutType%> warning scheduled at <script
language='javascript'>WriteClientTime('<%#TimeOutValue%>',1);</script>.
]]>
 </message>
 <message id="ActionForwarded">
 <![CDATA[
 Action <%#ActionName%> of type <%#ActionType%> was forwarded by actor
<%#ResourceName%> with identifier <%#ResourceIdentifier%>
to
<%#ResourceName1%> with identifier <%#ResourceIdentifier1%>
at <script
 language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.
]]>
 </message>
 <message id="ActionCompleted">
 <![CDATA[
Action <%#ActionName%> of type <%#ActionType%> completed execution at <script
 language='javascript'>WriteClientTime('<%#EventTime%>',1);</script> with the output
 <%#ActionOutput%>.
]]>
 </message>
 <message id="ActionSleeping">
 <![CDATA[
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<%#ResourceName%> with identifier <%#ResourceIdentifier%> selected
 <%#ActionOutput%> for the
Action <%#ActionName%> of type <%#ActionType%>. <br> The Action executed with output
<%#ActionOutput%> at <script
 language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.
]]>
 </message>
 <message id="ActionEvent">
 <![CDATA[
 <%#ResourceName%> with identifier <%#ResourceIdentifier%> has selected
 "<b><%#ActionEvent%><b>" for the Action <%#ActionName%> of type
<%#ActionType%>  at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.
The comment given by actor is :<%#ActionComment%>.
]]>
 </message>
 <message id="AckActionEvent">
 <![CDATA[
<%#ResourceName%> with identifier <%#ResourceIdentifier%> has <%#ActionEvent%> for
the Action <%#ActionName%> of type <%#ActionType%>  at <script
 language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.
]]>
 </message>
 <message id="ActionEventOutput">
 <![CDATA[
Perfomed <%#TimeOutType%>  calculations based on the properties set.
The output <%#ActionOutput%> was alerted to engine  at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.
]]>
 </message>
 <message id="SchedulerEvent">
 <![CDATA[
Perfomed Timeout <%#TimeOutType%>  from the scheduler at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.
The ouput <%#ActionOutput%> was alerted to engine.
]]>
 </message>
 <message id="SchedulerWarningEvent">
 <![CDATA[
Perfomed Timeout <%#TimeOutType%> warning from the scheduler at <script
 language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.
The ouput <%#ActionOutput%> was alerted to engine.
]]>
 </message>
 </messages>
 </report>
 <properties>
 <property category="cat_name;01"  name="Action Display Name" displayname="Display Name"
type="string" helpstring="Display name for the action in activity list."></property>
 <property category="cat_configuration;02"  name="Delivery Channels"
type="ChannelSelection">
 <propertychannel>
 <channels>
 <channel name="WebWorkList" supported="true">
 <somechanneldata visible="" selected="" mandatory="">
 </somechanneldata>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 <forms>
 <form type="NativeWebForm" supported="true"  defaultsetting=""
  uihandlerclassname="SampleApproval.SampleApprovalControl"
 uihandlerassembly="bin\SampleApprovalActivity.dll">
 <instance name="Approval Form"/>
 </form>
 </forms>
 </channel>
 <channel name="MailWorkList" supported="true" selected="false">
 <forms>
 <form type="MailWorkForm" supported="true"  defaultsetting="">
 <instance name="Approval Email Form"/>
 </form>
 </forms>
 </channel>
 </channels>
 </propertychannel>
 </property>

 <property category="cat_configuration" name="Work Item Fields" type="WorkItemField"
helptemplate="Approval-WorkItemFields.html" helptemplatewinheight="450"
helptemplatewinwidth="600">
 </property>
 <property category="cat_participants;03" name="To" displayname="Assign Actor(s)"
type="resource" helpstring="Actor filter, Expression which selects the actors to whom
the request
 needs to be sent." mandatory="false" helptemplate="Approval-AssignActors.html"
 helptemplatewinheight="450" helptemplatewinwidth="600">
 </property>
 <property category="cat_participants" name="Assign Queue" displayname="Assign Queue(s)"
  type="Queue" helptemplate="Approval-AssignQueues.html" helptemplatewinheight="450"
helptemplatewinwidth="600">
 </property>
 <property category="cat_participants" name="Min. Acknowledgements [% or Number]"
 displayname="Min. Ownerships [% or Number]" mandatory="true" type="limit"
defaultvalue="1"
 helpstring="Minimum Number of actors to acknowledge the approval request. Approve and
Reject
 actions are considered as acknowledged.">
 </property>
 <property category="cat_participants" name="Max. Acknowledgements [% or Number]"
  displayname="Max. Ownerships [% or Number]" type="limit" defaultvalue="100%"
  helpstring="Maximum Number of actors who should acknowledge and own the Approval
  Request">
 </property>
 <property category="cat_participants" name="Forwarding Allowed?" displayname="Re-Assign
Allowed?"
 type="choice" defaultvalue="No" helpstring="Allow request to be re-assigned by an actor
to another actor.">
 <choice>Yes</choice>
 <choice>No</choice>
 </property>
 <property  category="cat_participants" name="Ignore Unavailable Resources?"
 displayname="Ignore Unavailable Actors?" type="choice" defaultvalue="No"
helpstring="Ignore
 actors who are on Holidays?.">
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 <choice>Yes</choice>
 <choice>Yes - if No Alternate</choice>
 <choice>No</choice>
 </property>

 <property  category="cat_participants" name="Alternate Resource Allowed?"
displayname="Alternate Actor Allowed?" type="choice" defaultvalue="Yes"
helpstring="Allow
 request to be processed by an Alternate actor as specified by the original recipient
actor.">
 <choice>Yes</choice>
 <choice>No</choice>
 </property>


 <property category="cat_notification;04" name="Subject" type="workflowmemo"
  helpstring="Subject displayed to user in activity list. Also used as the subject for
notification email
sent to the user." helptemplate="Approval-Subject.html" helptemplatewinheight="450"
helptemplatewinwidth="600" ></property>
 <property category="cat_notification" name="Notes" displayname="Body"
type="workflowmemo"
  helpstring="Body/Notes for the approvers. Also used as the body for notification email
sent to the
 user."  helptemplate="Approval-Body.html" helptemplatewinheight="450"
 helptemplatewinwidth="600">
 </property>

 <property category="cat_notification" name="From Email Address" type="email"
helpstring="The
 from email address for email notification sent to the user."></property>
 <property category="cat_notification" name="Hide Responses?" type="choice"
helpstring="Incase
 multiple actors are acting on this approval, responses from all of them are public or
hidden during
 execution of this action.">
 <choice>Yes</choice>
 <choice>No</choice>
 </property>

 <property category="cat_notification" name="Show Custom Document View Window"
type="choice"
  defaultvalue="Yes" helpstring="Displays a window in activity list,which can have
custom document
view provided">
 <choice>Yes</choice>
 <choice>No</choice>
 </property>
 <property  category="cat_notification" name="Priority" type="choice"
defaultvalue="Medium"
 helpstring="Severity of the activity">
 <choice>High</choice>
 <choice>Medium</choice>
 <choice>Low</choice>
 </property>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 <property  category="cat_notification" name="Send Notification Email" defaultvalue="No"
  type="choice" helpstring="Default notification email to be sent or not.">
 <choice>Yes - HTML</choice>
 <choice>Yes</choice>
 <choice>No</choice>
 </property>

 <property category="cat_notification"  name="Redirect URL" type="workflowmemo"
  helpstring="URL to be redirected after the activity is added to a actor."
helptemplate="Approval-
RedirectURL.html" helptemplatewinheight="450"
helptemplatewinwidth="600"><![CDATA[http://<%ApplicationPath%>/]]></property>
 <property category="cat_notification" name="Custom Document View URL"
type="workflowmemo"
 helpstring="URL to be shown in the custom document view." helptemplate="Approval-
 CustomDocumentViewURL.html" helptemplatewinheight="450"
 helptemplatewinwidth="600">
 <![CDATA[http://<%ApplicationPath%>/]]>
 </property>

 <property category="cat_decision;05" name="Affirmative Answer for Approval [% or
Number]"
  type="limit" defaultvalue="100%" helpstring="Number or % of actors should approve in
order for
the outcome of this action to be set as approved">
 </property>

 <property category="cat_decision" name="Action Limit[% or Number]" type="limit"
 defaultvalue="100%" helpstring="Minimum number or % of actors to act upon the approval
 request to avoid action timeouts and warning.">
 </property>

 <property category="cat_decision"  name="Wait For All The Participants To Act"
displayname="Wait
for all the participants to Act" type="choice" defaultvalue="No" helpstring="Wait for
all the
 participants to take decision on the action.">
 <choice>Yes</choice>
 <choice>No</choice>
 </property>

 <property category="cat_timeout;06" name="Set Calendar" displayname="Set Calendar for
 Timeout" type="calendar" helptemplate="Approval-SetCalendarForTimeout.html"
 helptemplatewinheight="450" helptemplatewinwidth="600">
 </property>

 <property category="cat_timeout" name="Minimum Time for Action" displayname="Minimum
 available time for Activity" type="timespan" defaultvalue="0" helpstring="Min Time (in
Day:Hours:Minutes) that is required to finish the action.User should have the this time
left for the
 day to act up on the request.">
 </property>

 <property category="cat_timeout" name="Timeout warning for Acknowledgement"
displayname="Timeout warning for Ownership" type="timespan" defaultvalue="0"
helpstring="Max
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Time (in Day:Hours:Minutes) which can elapse for actors to acknowledge the request
before action
 will trigger the Timeout warning for ownership link">
 </property>
 <property category="cat_timeout" name="Timeout for Acknowledgement"
displayname="Timeout
 for Ownership" type="timespan" defaultvalue="0" helpstring="Max Time (in
Day:Hours:Minutes)
 which can elapse for actors to acknowledge the request before action will exit with
Timeout for
ownership link">
 </property>

 <property category="cat_timeout" name="Timeout warning for Action" displayname="Timeout
warning for Activity" type="timespan" defaultvalue="0" helpstring="Max Time (in
 Day:Hours:Minutes) which can elapse for actors to Act on the request before action will
trigger the
Timeout warning for Action link">
 </property>

 <property category="cat_timeout" name="Timeout for Action" displayname="Timeout for
Activity"
 type="timespan" defaultvalue="0" helpstring="Max Time (in Day:Hours:Minutes) which can
elapse
 for actors to Act on the request before action will exit with Timeout for Action link">
 </property>

 <property category="cat_timeout" name="Multiple TimeOut Warning" displayname ="Multiple
Timeout Warning" type="grid" helpstring="Multiple warning escalations for the action, If
user
 doesn't act up on the action for a given time." helptemplate="Approval-
 MultipleTimeoutWarning.html" helptemplatewinheight="450" helptemplatewinwidth="600">
 <cols>
 <col mandatory="true" displaywidth="25%" validationtype="timespan">
 <titlestring>Time Interval</titlestring>
 <choicedefaultvalue>0.00:00:00</choicedefaultvalue>
 </col>
 <col type="select" values="Select:,Yes:YES,No:NO" mandatory="true" displaywidth="10%">
 <titlestring>Recurring?</titlestring>
 </col>
 <col mandatory="true" displaywidth="40%">
 <titlestring>OutPut</titlestring>
 <choicedefaultvalue>TimeOutWarning-</choicedefaultvalue>
 </col>
 <col displaywidth="25%" validationtype="timespan">
 <titlestring>Recurring Time Interval</titlestring>
 <choicedefaultvalue>0.00:00:00</choicedefaultvalue>
 </col>
 </cols>
 </property>
 </properties>

 <return allowblankoutput="false">
 <link property="Multiple TimeOut Warning" defaultcolumn="3"/>
 <value helpstring="Request is Approved">Approved</value>
 <value helpstring="Request is Rejected">Rejected</value>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 <value helpstring="Filter Condition did not yield enough actors to whom the request can
be
 sent">Not Enough Resources to Acknowledge</value>
 <value helpstring="Minimum Actors as specified didnt acknowledge the request in the
given
 time">Timeout - Acknowledgement</value>
 <value helpstring="Some or All Actors didn't perform the required action in given
time">
 Timeout – Action
 </value>
 <value helpstring="Warning for Minimum Actors as specified didn't acknowledge the
request in the
 given time">Timeout Warning - Acknowledgement</value>
 <value helpstring="Warning since Some or All Actors didn't perform the required action
in given
time">Timeout Warning - Action</value>
 </return>
 </action>
```

- The new Custom Activity will get displayed in the Process Designer under the category "Custom Activities", as shown in the image below.

Custom Choice Activity

AVEVA Work Tasks provides a set of interfaces and their supporting classes that can be implemented in the class of a custom activity to provide human activity behavior.

The following code sample shows how to create a sample choice activity that looks and behaves like the AVEVA Work Tasks defined choice activity with the usage of Sample aspx property inside the Sample Choice activity.

Procedure

Dll REFERENCE

```
System.Web
Workflow.NET.NET2
Skelta.HWS.ActivityHandler
```

NAMESPACE USED

```
System
System.Web
System.Web.UI
System.Collections
System.Collections.Generic
System.Text
Workflow.NET.Engine
Workflow.NET.Interfaces
Skelta.Core
Skelta.HWS
```

To create a Sample Custom Choice Activity

1. Go to **Start > Visual Studio > New > Projects >Class Library >SampleChoiceActivity.** This step will create a Class Library project in the path specified by the name SampleChoiceActivity.
2. Create two class files under the project SampleChoiceHandler.cs and SampleChoiceControl.cs.
3. In the SampleChoiceHandler.cs class file, inherit the HWSActivityProviderBase class which is the base class

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

for all the resource activities.

4. In the SampleChoiceControl.cs class file, inherit the class ActivityUIBase while rendering the Work Item Detail view. Inherit the INativeWebFormUIControl interface and it is used for setting the Current Work Item context to be used in the Work Item Detail view

```
using System;
using System.Collections.Generic;
using System.Text;
using Workflow.NET;
using Workflow.NET.PropertyTypes;
using Workflow.NET.Storage;
using Workflow.NET.Interfaces;
using Workflow.NET.Engine;
using Workflow.NET.Engine.Interfaces;
using Skelta.Core;
using Skelta.HWS;
namespace SampleChoiceActivity
{
public class SampleChoiceHandler : HWSActivityProviderBase
{
/// <summary>
/// Constructor for sample Choice Handler Class
/// </summary>
public SampleChoiceHandler()
{
}
/// <summary>
/// Resolves the property names. Existing properties can be given different names using
this API.
/// MaximumActorsRequiredForActionCompletion is used in the Skelta for getting the
property value
/// as for approval it will be Affirmative Answer for Approval [% or Number] and for
choice it will be
/// Affirmative Answer for TaskAssignment Completion [% or Number].
/// </summary>
/// <param name="propertyName">Standard property name.</param>
/// <returns>Property name set for the activity.</returns>
public override string ResolvePropertyName(string propertyName)
{
switch (propertyName)
{
case "MaximumActorsRequiredForActionCompletion":
return "Affirmative Answer for TaskAssignment Completion [% or Number]";
case "MinimumActorsRequiredForActionCompletion":
return "Action Min Limit TaskAssignment Completion [% or Number]";
}
return base.ResolvePropertyName(propertyName);
}
/// <summary>
/// Overrides the Run Method of HWSActivityProviderBase for accessing the Custom Aspx
Property
/// </summary>
/// <param name="ExecutionID">ExecutionID of the Workflow</param>
/// <param name="ExecutionDetailsID">ExecutionDetailsID of the Activity</param>
/// <param name="CurrentContext">Context Details of Currently Executing Workflow</param>
/// <param name="CurrentAction">Current Executing Activity</param>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// <param name="InlinkOutput">The Output from previous Activity Linked to this
Activity</param>
/// <param name="Retrying"></param>
/// <param name="Output">Output</param>
/// <returns>ActionResult</returns>
public override Workflow.NET.Engine.Interfaces.ActionResult Run(int ExecutionID, int
ExecutionDetailsID, Workflow.NET.Engine.Context CurrentContext, Workflow.NET.Action
CurrentAction, string InlinkOutput, bool Retrying, out string Output)
{
//Access custom property before calling the base run
Workflow.NET.Property property = CurrentAction.Properties["SampleASPXProperty"];
Workflow.NET.SampleAspxProperty.SampleCustomASPXPropertyType aspxProperty =
(Workflow.NET.SampleAspxProperty.SampleCustomASPXPropertyType)property.PropertyHandler;
CurrentContext.log.LogInformation("@#@AspxProperty.DropDownValue:" +
aspxProperty.DropDownValue);
CurrentContext.log.LogInformation("@#@AspxProperty.TextBoxValue:" +
aspxProperty.TextBoxValue);
return base.Run(ExecutionID, ExecutionDetailsID, CurrentContext, CurrentAction,
InlinkOutput, Retrying, out Output);
}
/// <summary>
/// Gets the possible status list for the activity meaning output for the activity.
/// Choice Activity Uses a Skelta grid property (Action Outputs) for defining output for
the action.
/// Below code shows the reading of Action outputs grid property for getting all the
action outputs.
/// </summary>
/// <returns>List of possible outputs.</returns>
public override Dictionary<string, string> GetPossibleStatusList()
{
Dictionary<string, string> choiceValueandDisplayValue = new Dictionary<string,
string>();
string[] statusList = new string[10];
PropertyGrid p =
(PropertyGrid)(((Workflow.NET.Property)CurrentHWSActivity.CurrentActivity.Properties["Ac
tion Outputs"]).PropertyHandler);
System.Collections.Hashtable _HtButtonTypes = new System.Collections.Hashtable();
object[] rows = p.GridValue;
int k = 0;
if (rows != null)
{
foreach (object row in rows)
{
object[] cols = (object[])row;
if (cols != null)
{
object[] multipleArray = new object[cols.Length + 1];
int j = 0;
foreach (object col in cols)
{
multipleArray[j] = col;
j++;
}
multipleArray[j] = null;
_HtButtonTypes.Add(k, multipleArray);
k++;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
}
}
foreach (object[] objRow in _HtButtonTypes.Values)
{
string buttonvalue = objRow[0].ToString();
string decisionmakervalue = objRow[1].ToString();
choiceValueandDisplayValue.Add(buttonvalue, buttonvalue);
}
return choiceValueandDisplayValue;
}
/// <summary>
/// This method is used to check the specified output is completion maker or not.
/// </summary>
/// <param name="Status">Status Passed</param>
/// <returns>True or False</returns>
public override bool IsCompletionMaker(string Status)
{
bool CompletionMaker = true;
PropertyGrid p =
(PropertyGrid)(((Workflow.NET.Property)CurrentHWSActivity.CurrentActivity.Properties["Ac
tion Outputs"]).PropertyHandler);
System.Collections.Hashtable _HtDecisionMaker = new System.Collections.Hashtable();
object[] rows = p.GridValue;
int k = 0;
if (rows != null)
{
foreach (object row in rows)
{
object[] cols = (object[])row;
if (cols != null)
{
object[] multipleArray = new object[cols.Length + 1];
int j = 0;
foreach (object col in cols)
{
multipleArray[j] = col;
j++;
}
multipleArray[j] = null;
_HtDecisionMaker.Add(k, multipleArray);
k++;
}
}
}
foreach (object[] objRow in _HtDecisionMaker.Values)
{
string buttonvalue = objRow[0].ToString();
string decisionmakervalue = objRow[1].ToString();
if (buttonvalue != "")
{
if (buttonvalue.ToUpperInvariant().Trim() == Status.ToUpperInvariant().Trim())
{
if (decisionmakervalue.ToUpperInvariant() == "YES")
{
CompletionMaker = true;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
else
{
CompletionMaker = false;
}
}
}
}
return CompletionMaker;
}
}
}
```

SampleChoiceControl.cs

```
using System;
using System.Web;
using System.Web.UI;
using System.Collections.Generic;
using System.Text;
using Workflow.NET.Engine;
using Workflow.NET.Interfaces;
using Skelta.Core;
using Skelta.HWS;
using System.Collections;
namespace SampleChoiceActivity
{
/// <summary>
/// The UI for sample custom choice activity is rendered from an HTML file. This will
help us in editing the UI with out changing the compiled activity code.
/// The HTML template is placed in folder ([SkelatIntalledPath]\WorkflowElements\
Default\en-US\Activities\Activity\SampleChoiceUI.htm)
/// </summary>
public class SampleChoiceControl : Skelta.HWS.ActivityHandler.ActivityUIBase,
INamingContainer, Skelta.HWS.WorkListChannel.Web.INativeWebFormUIControl
{
public HWSActivity _HWSActivity;
public WorkItem _WorkItem;
Workflow.NET.Engine.Context WorkflowContext;
Workflow.NET.Log logger = new Workflow.NET.Log();
private Workflow.NET.Parser parseHtml = new Workflow.NET.Parser();
string strTempDirectoryPath = "", strDirectoryPath = "";
Workflow.NET.ActionsData _ActionsData;
/// <summary>
/// Constructor for SampleChoiceControl class
/// </summary>
public SampleChoiceControl()
{
}
/// <summary>
/// This method is used to render the Workitem Detail view
/// </summary>
protected override void CreateChildControls()
{
string InternalStatus = _WorkItem.Status.ToUpperInvariant();
string ErrorMessage = "";
string VariableName = "";
string ActionTaken = "";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
string strContent = "";
string strStatus = "";
Dictionary<string, string> GetStatusList = new Dictionary<string, string>();
int poscnt = 0;
string Comment = Context.Request["txtComment"];
if (Comment != null)
Comment = Comment.Trim();
strStatus = Context.Request["btnSubmit"];
if (strStatus != null && strStatus != "")
{
try
{
SynchronousActionData syncData = new SynchronousActionData(HttpContext.Current);
//Submitting the Workitem with the respective output
_WorkItem.Submit("WebWorkList", strStatus, "", Comment, syncData);
//RefreshWorkItemList() is the commonfunction which is used for Refreshing
//the Workitem List after submitting the workitem
this.Controls.Add(new LiteralControl("<script>RefreshWorkItemList();</script>"));
}
catch (Exception e)
{
}
}
if (InternalStatus == "AW"
|| InternalStatus == "OW" || InternalStatus == "HO")
{
//Error message is rendered below the notes.
base.ErrorMessage = ErrorMessage;
//This is called for Rendering a common UI in Workitem Detail View Which is displaying
Subject,Notes and Manage Workitem drop down and also
// IFrame which will display the Custom Document view in Left hand side of Details view.
RenderCommonActivityUI(parseHtml);
((IHWSActivityProvider)_HWSActivity.CurrentActivity.ActionHandler).CurrentHWSActivity =
_HWSActivity;
GetStatusList =
((IHWSActivityProvider)_HWSActivity.CurrentActivity.ActionHandler).GetPossibleStatusList
();
int cnt = GetStatusList.Count;
foreach (string buttonvalue in GetStatusList.Values)
{
if (buttonvalue != "")
{
poscnt++;
strContent = "<input type=\"submit\" class=\"inputbutton\" value=\"" + buttonvalue + "\"
name=\"btnSubmit\">";
parseHtml.Replace("Position" + poscnt, strContent);
}
}
int remainingCount = 10 - GetStatusList.Count;
if (remainingCount > 0)
{
for (int m = cnt + 1; m <= 10; m++)
{
parseHtml.Replace("Position" + m, "");
}
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//Rendering the Detail View UI(SampleChoiceUI.htm)
this.Controls.Add(new
LiteralControl(parseHtml.CheckAndProcessFileGetString(strDirectoryPath +
"Activities\\Activity\\SampleChoiceUI.htm", strTempDirectoryPath + "cache",
"Activities\\Activity\\")));
}
}
/// <summary>
/// Overrides Render Method of Webcontrol
/// </summary>
/// <param name="writer"></param>
protected override void Render(HtmlTextWriter writer)
{
this.RenderBeginTag(writer);
this.RenderChildren(writer);
this.RenderEndTag(writer);
}
#region INativeWebFormUIControl Members
/// <summary>
/// This method Provides information of the current workitem instance.
/// </summary>
/// <param name="workItem">Current Workitem Instance</param>
/// <param name="channelName">Channel Name(WebWorkList)</param>
/// <param name="formName">Form Name(Approval Form)</param>
void Skelta.HWS.WorkListChannel.Web.INativeWebFormUIControl.SetContext(WorkItem
workItem, string channelName, string formName)
{
_WorkItem = workItem;
_HWSActivity = workItem.HWSActivity;
WorkflowContext = workItem.CurrentContext;
_ChannelName = channelName;
_FormName = formName;
strTempDirectoryPath =
workItem.HWSActivity.CurrentContext.ProcessDefinition.ActionsDefinition.DirTemp;
strDirectoryPath =
workItem.HWSActivity.CurrentContext.ProcessDefinition.ActionsDefinition.DirSourceElement
s;
_ActionsData = workItem.CurrentContext.ProcessDefinition.ActionsDefinition;
base.Initailize(this.Context, workItem);
}
#endregion
}
}
```

5. Build the Class Library Project and place the assembly in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\bin* folder.
6. Create an HTML Page by the name SampleChoiceUI.htm in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Activities\Activity* folder. This SampleChoiceUI.htm is used for rendering the UI in the Work Item list as a Work Item Detail view. Copy the following contents in the created HTML page.

```
<TABLE WIDTH="100%" height="100%" BORDER="0" CELLPADDING="2" CELLSPACING="1"
class='righttdbg'>
<%#errorMessage%>
<TR VALIGN="TOP">
<TD align="left" width="<%#iframewidth%>" bgcolor="<%#iframebgcolor%>">
<%#iframeui%>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</TD>
<TD>
<TABLE WIDTH="100%" BORDER="0" CELLSPACING="1" CELLPADDING="5">
<TR>
<TD align=left class='subtitle'>
<strong>@@Comments@@</strong>
</TD>
</TR>
<TR>
<TD align="left">
<TEXTAREA name='txtComment' cols=35 rows=15 wrap="VIRTUAL"
class="inputtextarea"></TEXTAREA>
</TD>
</TR>
<!--<TR>
<TD align="center"> </TD>
</TR>-->
</TABLE>
</TD>
</TR>
<TR VALIGN="TOP">
<TD height="25" colspan=2 align="right" bgcolor="#ffffff">
<%#Position1%>
<%#Position2%>
<%#Position3%>
<%#Position4%>
<%#Position5%>
<%#Position6%>
<%#Position7%>
<%#Position8%>
<%#Position9%>
<%#Position10%>
</TD>
</TR>
</TABLE>
```

7. Configure the Actions.XML file available in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Actions\XML* path for the created Sample Choice activity in the Custom Activities category. While configuring the Custom Activity, you must specify the image filename of type .PNG. The .png image must be of size 40x40 for the custom activity. The image must be available in the following folders:

   [AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Actions\Images\Normal\Large

   [AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Actions\Images\Selected\Large

   [AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\ActivityImages

   Note: If an XML exception occurs after copying the following configuration to the Actions.XML file, open the Actions.XML file in a browser, and remove the extra spaces, if any.

   Copy the following XML code to the <category name= "Custom Activities"> section of the Actions.XML file.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For example, <category name= "Custom Activities" sortorder="6" image="custom.gif">

```
<action name="SampleChoice" helptemplate="ChoiceAction1.html"
helptemplatewinheight="450" helptemplatewinwidth="600">
<description>Dynamic button action outputs.</description>
<image resourcename="choice.png"></image>
<handler classname="SampleChoiceActivity.SampleChoiceHandler" assembly="bin\
SampleChoiceActivity.dll"></handler>
<wizards>
<wizard name="Setup Actors" windowwidth="540" windowheight="290" windowposition="center"
category="cat_participants" default="true" description="Use this wizard for quickly
setting up Actors for the action" url="ActorsSelectionWizard.aspx">
<![CDATA[Resource=To,Minimum Acknowledgers=Min. Acknowledgements [% or Number],
Maximum Acknowledgers=Max. Acknowledgements [% or Number],
Maximum Resources to Act=Affirmative Answer for TaskAssignment Completion [% or Number],
Minimum Resources to Act=Action Min Limit TaskAssignment Completion [% or Number]]]>
</wizard>
<wizard name="Setup Escalations" windowwidth="540" windowheight="290" description="Use
this wizard for setting up Escalations incase the actors dont act on time"
url="EscalationEmailWizard.aspx">
</wizard>
</wizards>
<report rendertype="xsl" xsl="ActionDescription.xsl">
<handler class="Workflow.NET.Report.ReportHandler" assembly="Workflow.NET.ReportHandler\
bin\Debug\Workflow.NET.ReportHandler.NET2.dll" />
<messages>
<message id="ActionStarted">
<![CDATA[Action <%#ActionName%> of type <%#ActionType%> initiated at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.]]>
</message>
<message id="ResourceScarcity">
<![CDATA[Action <%#ActionName%> of type <%#ActionType%> completed execution due to lack
of actor.
<%#ActionName%> completed with the output <%#ActionOutput%>.]]>
</message>
<message id="FilteredResourceCount">
<![CDATA[The <%#ActionType%> action <%#ActionName%> selected <%#ResourceCount%>
actor<%#ResourceOrResources%>.]]>
</message>
<message id="RequestResource">
<![CDATA[The actor named <%#ResourceName%> with identifier <%#ResourceIdentifier%>
has been identified to take part in the <%#ActionName%> action.]]>
</message>
<message id="ResourceUnavailable">
<![CDATA[The actor <%#ResourceName%> with identifier <%#ResourceIdentifier%> and email
<%#ResourceEmail%> is unavailable.<br>
The 'Ignore unavailable actor' property for the action was set to
<%#IgnoreUnavailableResource%>.<br>
The 'Alternate actor allowed' property was set to <%#AlternateResourceAllowed%>.]]>
</message>
<message id="AvailableResourceCount">
<![CDATA[The action <%#ActionName%> request was sent to <%#ResourceCount%>
actor<%#ResourceOrResources%>.<br>
The Minimum No.Of ownerships required is <%#PropertyMin. Acknowledgements [% or
Number]()%>.<br>
The Maximum No.Of ownerships is <%#PropertyMax. Acknowledgements [% or Number]()%>.<br>
The Minimum human activity count required is <%#PropertyAction Min Limit TaskAssignment
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Completion [% or Number]()%>.<br>
The Choice count required for the action to be completed is <%#PropertyAffirmative
Answer for TaskAssignment Completion [% or Number]()%>.<br>]]>
</message>
<message id="TimeOutSet">
<![CDATA[Timeout <%#TimeOutType%> scheduled at <script
language='javascript'>WriteClientTime('<%#TimeOutValue%>',1);</script>.]]>
</message>
<message id="TimeOutWarningSet">
<![CDATA[Timeout <%#TimeOutType%> warning scheduled at <script
language='javascript'>WriteClientTime('<%#TimeOutValue%>',1);</script>.]]>
</message>
<message id="ActionForwarded">
<![CDATA[Action <%#ActionName%> of type <%#ActionType%> was forwarded by actor
<%#ResourceName%> with identifier <%#ResourceIdentifier%> to
<%#ResourceName1%> with identifier <%#ResourceIdentifier1%>at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.]]>
</message>
<message id="ActionCompleted">
<![CDATA[Action <%#ActionName%> of type <%#ActionType%> completed execution at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script> with the output
<%#ActionOutput%>.]]>
</message>
<message id="ActionEvent">
<![CDATA[<%#ResourceName%> with identifier <%#ResourceIdentifier%> selected
<%#ActionOutput%> for the Action <%#ActionName%> of type <%#ActionType%>. <br>
The Action executed with output <%#ActionOutput%> at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.
The comment given by actor is :<%#ActionComment%>.]]>
</message>
<message id="ActionSleeping">
<![CDATA[<%#ResourceName%> with identifier <%#ResourceIdentifier%> selected
<%#ActionOutput%> for the Action <%#ActionName%> of type <%#ActionType%>.at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>. <br>
The comment given by actor is :<%#ActionComment%>.]]>
</message>
<message id="AckActionEvent">
<![CDATA[<%#ResourceName%> with identifier <%#ResourceIdentifier%> <%#ActionEvent%> the
Action <%#ActionName%> of type <%#ActionType%> at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.]]>
</message>
<message id="ActionEventOutput">
<![CDATA[Perfomed <%#TimeOutType%> calculations based on the properties set. The ouput
<%#ActionOutput%> was alerted to engine at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>..]]>
</message>
<message id="SchedulerEvent">
<![CDATA[Perfomed Timeout <%#TimeOutType%> from the scheduler at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.
The ouput <%#ActionOutput%> was alerted to engine.]]>
</message>
<message id="SchedulerWarningEvent">
<![CDATA[Perfomed Timeout <%#TimeOutType%> warning from the scheduler at <script
language='javascript'>WriteClientTime('<%#EventTime%>',1);</script>.
The ouput <%#ActionOutput%> was alerted to engine.]]>
</message>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<message id="CustomFieldWarning">
<![CDATA[Custom field <%#CustomFieldName%>(<%#CustomFieldDisplayName%>) of type
<%#CustomFieldType%> couldn't not be added. Process definition for action not updated
after changes in custom fields.]]>
</message>
</messages>
</report>
<properties>
<property category="cat_name;01" name="Action Display Name" displayname="Display Name"
type="string" helpstring="Display name for the action in activity list."></property>
<property name="SampleASPXProperty" displayname="SampleASPXProperty"
type="SampleAspxProperty" helpstring="Sample for rendering UI through an aspx
page."></property>
<property category="cat_configuration;02" name="Delivery Channels"
type="ChannelSelection">
<propertychannel>
<channels>
<channel name="WebWorkList" supported="true">
<somechanneldata visible="" selected="" mandatory="">
</somechanneldata>
<forms>
<form type="NativeWebForm" supported="true" defaultsetting=""
uihandlerclassname="SampleChoiceActivity.SampleChoiceControl" uihandlerassembly="bin\
SampleChoiceActivity.dll">
<instance name="Choice Form"/>
</form>
</forms>
</channel>
<channel name="MailWorkList" supported="true" selected="false">
<forms>
<form type="MailWorkForm" supported="true" defaultsetting="">
<instance name="Choice Email Form"/>
</form>
</forms>
</channel>
</channels>
</propertychannel>
</property>
<property category="cat_configuration" name="Action Outputs" type="grid"
helpstring="Different button types to be displayed." mandatory="true"
helptemplate="Choice-ActionOutputs.html" helptemplatewinheight="450"
helptemplatewinwidth="600">
<cols>
<col mandatory="true" titlestring="Button Name" displaywidth="50%" />
<col type="select" values="Select:,Yes:Yes,No:No" mandatory="true">
<titlestring>Completion Maker</titlestring>
</col>
<defaultvalue></defaultvalue>
</cols>
</property>
<property category="cat_configuration" name="Work Item Fields" type="WorkItemField"
elpstring="save activity level data to be used for activity list" helptemplate="Choice-
WorkItemFields.html" helptemplatewinheight="450" helptemplatewinwidth="600"></property>
<property category="cat_participants;03" name="To" displayname="Assign Actor(s)"
type="resource" helpstring="Actor filter, Expression which selects the actors to whom
the request needs to be sent." mandatory="false" helptemplate="Choice-AssignActors.html"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
helptemplatewinheight="450" helptemplatewinwidth="600"></property>
<property category="cat_participants" name="Assign Queue" displayname="Assign Queue(s)"
type="Queue" helptemplate="Choice-AssignQueues.html" helptemplatewinheight="450"
helptemplatewinwidth="600"></property>
<property category="cat_participants" name="Min. Acknowledgements [% or Number]"
displayname="Min. Ownerships [% or Number]" mandatory="true" type="limit"
defaultvalue="1" helpstring="Minimum Number of actors who should acknowledge and own the
Task Request"></property>
<property category="cat_participants" name="Max. Acknowledgements [% or Number]"
displayname="Max. Ownerships [% or Number]" type="limit" defaultvalue="100%"
helpstring="Maximum Number of actors who should acknowledge and own the Task
Request"></property>
<property category="cat_participants" name="Forwarding Allowed?" displayname="Re-Assign
Allowed?" type="choice" defaultvalue="No" helpstring="Allow request to be re-assigned by
a actor to another actor?">
<choice>Yes</choice>
<choice>No</choice>
</property>
<property category="cat_participants" name="Ignore Unavailable Resources?"
displayname="Ignore Unavailable Actors?" type="choice" defaultvalue="No"
helpstring="Ignore actors who are on Holidays?">
<choice>Yes</choice>
<choice>Yes - if No Alternate</choice>
<choice>No</choice>
</property>
<property category="cat_participants" name="Alternate Resource Allowed?"
displayname="Alternate actor Allowed?" type="choice" defaultvalue="Yes"
helpstring="Allow request to be processed by an Alternate actor as specified by the
orignial receipent actor">
<choice>Yes</choice>
<choice>No</choice>
</property>
<property category="cat_notification;04" name="Subject" type="workflowmemo"
displaymode="html" helpstring="Subject displayed to user in activity list. Also used as
the subject for notification email sent to the user." helptemplate="Choice-Subject.html"
helptemplatewinheight="450" helptemplatewinwidth="600"></property>
<property category="cat_notification" name="Notes" displayname="Body"
type="workflowmemo" displaymode="html" helpstring="Body/Notes for the approvers.Also
used as the body for notification email sent to the user." helptemplate="Choice-
Body.html" helptemplatewinheight="450" helptemplatewinwidth="600"></property>
<property category="cat_notification" name="From Email Address" type="email"
helpstring="The from email address for email notification sent to the user."></property>
<property category="cat_notification" name="Hide Responses?" defaultvalue="Yes"
type="choice" helpstring="Incase multiple actors are acting on this approval, responses
from all of them are public or hidden during execution of this action?.">
<choice>Yes</choice>
<choice>No</choice>
</property>
<property category="cat_notification" name="Priority" type="choice"
defaultvalue="Medium" helpstring="Severity of the activity">
<choice>High</choice>
<choice>Medium</choice>
<choice>Low</choice>
</property>
<property category="cat_notification" name="Show Custom Document View Window"
type="choice" defaultvalue="Yes" helpstring="Displays a window in activity list,which
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
can have custom document view provided">
<choice>Yes</choice>
<choice>No</choice>
</property>
<property category="cat_notification" name="Send Notification Email" defaultvalue="No"
type="choice" helpstring="Notification email to user about the request.">
<choice>Yes - HTML</choice>
<choice>Yes</choice>
<choice>No</choice>
</property>
<property category="cat_notification" name="Notify if alternate considered"
defaultvalue="No" type="choice" helpstring="Default notification email to be sent or
not.">
<choice>Yes - HTML</choice>
<choice>Yes</choice>
<choice>No</choice>
</property>
<property category="cat_notification" name="Redirect URL" type="workflowmemo"
helpstring="URL to be redirected after the activity is added to a actor."
helptemplate="Choice-RedirectURL.html" helptemplatewinheight="450"
helptemplatewinwidth="600"><![CDATA[http://<%ApplicationPath%>/]]></property>
<property category="cat_notification" name="Custom Document View URL"
type="workflowmemo" helpstring="URL to be shown in the custom document view."
helptemplate="Choice-CustomDocumentViewURL.html" helptemplatewinheight="450"
helptemplatewinwidth="600">
<![CDATA[http://<%ApplicationPath%>/]]>
<plugins>
<plugin name="sharepointexplorer" hiddenfield="" jsfunction="WritePluginContentViewUrl"
pageurl="../plugins/sharepoint/sharepointexplorer.aspx"
pagefeatures="width=450,height=700,top=10" imageurl="../Plugins/SharePoint/Images/
sharepoint-explorer.png" description="Click to open the SharePoint Explorer">
<parameters>
<parameter name="Help">Select the SharePoint view which you have defined with required
fields to be shown in the custom document view</parameter>
<parameter name="ExplorerLevel">View</parameter>
<parameter name="SelectionLevel">View</parameter>
<parameter name="MultiSelection">False</parameter>
<parameter name="ViewMode">1</parameter>
<parameter name="ReturnParentData">True</parameter>
</parameters>
</plugin>
</plugins>
</property>
<property category="cat_decision;05" name="Affirmative Answer for TaskAssignment
Completion [% or Number]" displayname="Max Actors Required For Choice Completion [% or
Number]" type="limit" defaultvalue="100%" helpstring="Number or % of actors should act
up on the Task in order for the outcome of this action to be set as
completed."></property>
<property category="cat_decision" name="Action Min Limit TaskAssignment Completion [% or
Number]" displayname="Min Actors Required For Choice Completion [% or Number]"
type="limit" defaultvalue="100%" helpstring="Minimum number or % of actors should act up
on the Task in order to avoid action timeouts and warning."></property>
<property category="cat_timeout;06" name="Set Calendar" displayname="Set Calendar for
Timeout" type="calendar" helptemplate="Choice-SetCalendarForTimeout.html"
helptemplatewinheight="450" helptemplatewinwidth="600"></property>
<property category="cat_timeout" name="Minimum Time for Action" displayname="Minimum
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
available time for Activity" type="timespan" defaultvalue="0" helpstring="Min Time (in
Day:Hours:Minutes) that is required to finish the action.User should have the this time
left for the day to act up on the request."></property>
<property category="cat_timeout" name="Timeout warning for Acknowledgement"
displayname="Timeout warning for Ownership" type="timespan" defaultvalue="0"
helpstring="Max Time (in Day:Hours:Minutes) which can elapse for actors to acknowledge
the request before action will trigger the Timeout warning for ownership
link"></property>
<property category="cat_timeout" name="Timeout for Acknowledgement" displayname="Timeout
for Ownership" type="timespan" defaultvalue="0" helpstring="Max Time (in
Day:Hours:Minutes) which can elapse for actors to acknowledge the request before action
will exit with Timeout for ownership link"></property>
<property category="cat_timeout" name="Timeout warning for Action" displayname="Timeout
warning for Activity" type="timespan" defaultvalue="0" helpstring="Max Time (in
Day:Hours:Minutes) which can elapse for actors to Act on the request before action will
trigger the Timeout warning for Action link"></property>
<property category="cat_timeout" name="Timeout for Action" displayname="Timeout for
Activity" type="timespan" defaultvalue="0" helpstring="Max Time (in Day:Hours:Minutes)
which can elapse for actors to Act on the request before action will exit with Timeout
for Action link"></property>
<property category="cat_timeout" name="Multiple TimeOut Warning" displayname ="Multiple
Timeout Warning" type="grid" helpstring="Multiple warning escalations for the action, If
user doesn't act up on the action for a given time." helptemplate="Choice-
MultipleTimeoutWarning.html" helptemplatewinheight="450" helptemplatewinwidth="600">
<cols>
<col mandatory="true" displaywidth="25%" validationtype="timespan">
<titlestring>Time Interval</titlestring>
<choicedefaultvalue>0.00:00:00</choicedefaultvalue>
</col>
<col type="select" values="Select:,Yes:YES,No:NO" mandatory="true" displaywidth="10%">
<titlestring>Recurring?</titlestring>
</col>
<col mandatory="true" displaywidth="40%">
<titlestring>OutPut</titlestring>
<choicedefaultvalue>TimeOutWarning-</choicedefaultvalue>
</col>
<col mandatory="true" displaywidth="25%" validationtype="timespan">
<titlestring>Recurring Time Interval</titlestring>
<choicedefaultvalue>0.00:00:00</choicedefaultvalue>
</col>
</cols>
</property>
<property name="Enable Redirection?" displayname="Redirection Enabled?" type="choice"
defaultvalue="No" helpstring="Indicates whether the subsequent actions should be
synchronized. If 'Yes', the client waits for the URL from the server and redirects the
UI to that URL.">
<choice>Yes</choice>
<choice>No</choice>
</property>
</properties>
<return>
<link property="Action Outputs" defaultcolumn="1"/>
<link property="Multiple TimeOut Warning" defaultcolumn="3"/>
<value helpstring="All the actors has taken part.">Action Completed</value>
<value helpstring="All the actors has not taken part.">Action Completed - Affirmative
Percentage Not Reached</value>
```

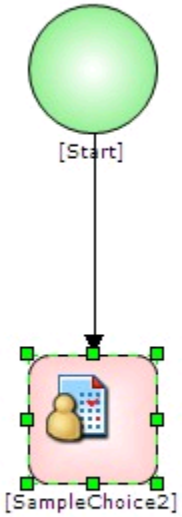
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<value helpstring="Filter Condition didnt yield enough actors to whom the request can be
sent">Not Enough Resources to Acknowledge</value>
<value helpstring="Minimum Actors as specified didnt acknowledge the request in the
given time">Timeout - Acknowledgement</value>
<value helpstring="Some or All Actors didnt perform the required action in given
time">Timeout - Action</value>
<value helpstring="Warning for Minimum Actors as specified didnt acknowledge the request
in the given time">Timeout Warning - Acknowledegement</value>
<value helpstring="Warning since Some or All Actors didnt perform the required action in
given time">Timeout Warning - Action</value>
<value helpstring="Queue Participant didn't perform the required action in given
time">Timeout Warning- Queue Participant</value>
<value helpstring="No Participant picked the work item from the Queue in given
time">Timeout Warning- Queue</value>
<value helpstring="Push dispatch pattern could not find any Actor.">Queue Actor
Unavailable</value>
</return>
</action>
```

Note: Ensure to set the assembly (assembly and uihandlerassembly attributes) correctly. Also ensure that the current configuration loads the dll from AVEVA Work Tasks installed bin folder.

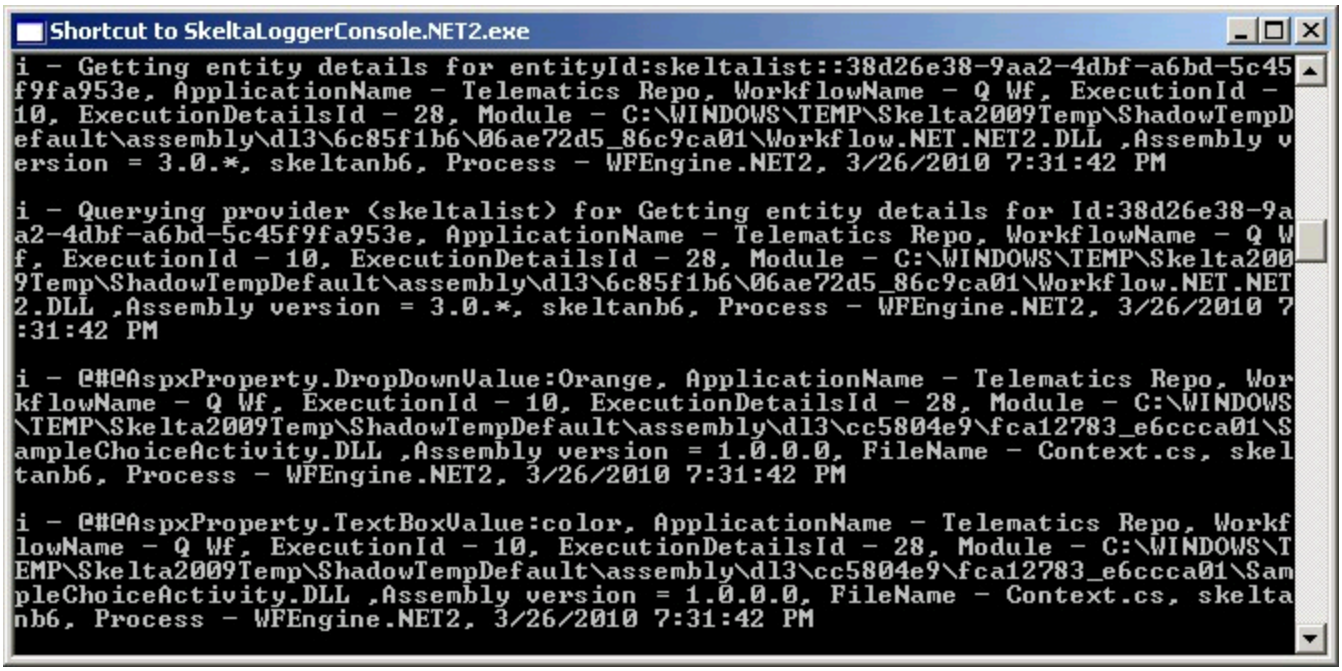
8. Open the Process Designer. The activity is displayed under the Custom Activities category.
9. Drag and drop the activity into the Process Designer. You can see the Custom Aspx Property in the Process Designer

The following figure shows the sample workflow created using the Sample Choice Activity.



We can the see the values set for the ASPX Property in the AVEVA Work Tasks Logger Console.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Implementation of Custom WorkItemForm

The WorkItemForm interface exposes different events available for a work item. For example, when a work item is created for an actor, a work item is updated by an actor etc.

Procedure

Dll Reference

```
System
System.Data
System.Xml
Workflow.NET.NET2
Skelta.HWS
```

Namespace Used

```
System
System.Collections.Generic
System.Text
System.Xml
Workflow.NET
Skelta.HWS.WorkListChannel
Skelta.HWS.WorkListChannel.Web
Skelta.HWS
Skelta.Core
```

To create a Custom Property

1. Go to Start->Visual Studio 2005->New->Projects->Class Library->CustomWorkItemForm. This step will create a Class Library project in the path specified by the name CustomWorkItemForm.
2. Create a class file under the project CustomWorkItemForm.cs.
3. Inherit the WorkItemFormBase class which internally implementing the IWorkItemForm interface and implement the interface ISkeltaAddInProvider. Activity and work item level information is maintained within

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the application by overriding the methods OnWorkItemTransaction, OnActivityTransaction, and OnWorkItemCreation. Suppose we want to perform some extra step when a work item is forwarded by the user, for this we can handle it in the method OnWorkItemTransaction.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using Workflow.NET;
using Skelta.HWS.WorkListChannel;
using Skelta.HWS.WorkListChannel.Web;
using Skelta.HWS;
using Skelta.Core;

namespace Skelta.HWS.WorkListChannel.Web
{
public class CustomWorkItemForm : WorkItemFormBase, ISkeltaAddInProvider
{
string _Name = null;
Workflow.NET.Log logger = new Workflow.NET.Log();
#region IWorkItemForm Members
public override string InitialFormUri
{
get
{
return "";
}
}

/// <summary>
/// Gets the boolean flag to check if new instance of form type can be added.
/// </summary>
public override bool CanAddNewInstance
{
get
{
bool found = false;
if (ParentFormsCollection != null)
{
foreach (IWorkItemForm form in ParentFormsCollection.SelectedFormInstances.Values)
{
if (form.FormType == this.FormType)
{
found = true;
break;
}
}
}

if (found)
return false;
else
return true;
}
}

/// <summary>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// Gets fired for all the transactions for a workitem.
/// </summary>
/// <param name="transactionType">Transactiontype</param>
/// <param name="workItem">Current workitem instance.</param>
/// <param name="transactionData">Transaction data.</param>
/// <remarks>
/// Few of the transactiontypes are update, forward, pause, resume , abort.
/// Transactiondata contains the actor object if the transaction type is forward  and it
contains the activity output details if the transaction type is  submitting a workitem.
/// </remarks>
public override void OnWorkItemTransaction(string transactionType, WorkItem workItem,
Dictionary<string, object> transactionData)
{
logger.LogInformation("****************************************");
logger.LogInformation("Called During Workitem transaction");
logger.LogInformation("****************************************");
logger.LogInformation("Status of the workitem is "+workItem.Status);

}


/// <summary>
/// Gets fired for all the transactions for an activity.
/// </summary>
/// <param name="transactionType">Transaction type.</param>
/// <param name="hwsActivity">Current hws activity instance.</param>
/// <param name="transactionData">Transaction data.</param>
/// <remarks>
/// Few of the transactiontypes are timeout, pause, resume, abort,  actioncompletion.
/// </remarks>
public override void OnActivityTransaction(string transactionType, HWSActivity
hwsActivity, Dictionary<string, object> transactionData)
{
logger.LogInformation("****************************************");
logger.LogInformation("Called for On Activity transaction");
logger.LogInformation("****************************************");
logger.LogInformation("Transaction type is "+transactionType+" and activity name is
"+hwsActivity.ActivityName);
}


/// <summary>
///  Gets fired when a workitem is created while executing a resource activity.
/// </summary>
/// <param name="workItem">Workitem Instance</param>
public override void OnWorkItemCreation(WorkItem workItem)
{
logger.LogInformation("****************************************");
logger.LogInformation("Called during the Worktem creation");
logger.LogInformation("Workitem : "+workItem.ActivityDisplayName+" and the Workitem
status is: "+workItem.Status);
logger.LogInformation("****************************************");

}

/// <summary>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// Gets custom configuration settings page. This page is loaded through the  channel
property in the Process Designer.
/// </summary>
public override string FormConfigurationUserInterfaceURL
{
get
{
 return "";
}
}

/// <summary>
/// Gets or sets the form name.
/// </summary>

public override string Name
{
get
{
return _Name;
}

set
{
_Name = value;
}
}

/// <summary>
/// Gets the form type.
/// </summary>

public override string FormType
{
get
{
return "CustomWorkItemForm";
}
}

/// <summary>
/// Gets an array of supported  technology types.
/// </summary>
/// <remarks>
/// e.g: "Web", "Mail".
/// </remarks>

public override string[] SupportedTechnologyTypes
{
get
{
return new string[] { "Web" };
}
}

/// <summary>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// Returns default form instances.
/// </summary>
/// <returns>Default form instances of type IWorkItemForm</returns>
public override IWorkItemForm[] GiveDefaultInstances()
{
Skelta.HWS.WorkListChannel.Web.CustomWorkItemForm frm = new
Skelta.HWS.WorkListChannel.Web.CustomWorkItemForm();
frm.Name = "Custom Workitem Form";

IWorkItemForm[] formins = new IWorkItemForm[] { frm };
return formins;

}

/// <summary>
/// Returns a cloned form instance
/// </summary>
/// <returns>Cloned form instance</returns>
public override IWorkItemForm Clone()
{

Skelta.HWS.WorkListChannel.Web.CustomWorkItemForm nativeWebForm =
(Skelta.HWS.WorkListChannel.Web.CustomWorkItemForm)MemberwiseClone();
if (this.Roles != null && this.Roles.Length > 0)
nativeWebForm.Roles = new List<string>(this.Roles).ToArray();

if (this.Actors != null && this.Actors.Length > 0)
nativeWebForm.Actors = new List<Guid>(this.Actors).ToArray();
nativeWebForm._RunTimeProperties = new Dictionary<string, object>();

foreach (string key in this._RunTimeProperties.Keys)
{
nativeWebForm._RunTimeProperties.Add(key, _RunTimeProperties[key]);
}

return nativeWebForm;

}

/// <summary>
/// Gets a list of errors after validation of form type.
/// </summary>
/// <param name="currentActivity">current activity</param>
/// <returns>list of errors.</returns>
/// <remarks>
/// Useful for performing configuration settings validation.
/// </remarks>
public override List<Workflow.NET.ProcessValidationError> IsValid(Workflow.NET.Action
currentActivity)
{

List<ProcessValidationError> errors = new List<ProcessValidationError>();
return errors;
}

#endregion
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
#region ISkeltaAddInProvider Members

string _Settings;
Guid _Id;
public Guid Id
{
get { return _Id; }
}

public void InitializeProvider(string settings, Guid id)
{
this._Settings = settings;
this._Id = id;
}

public string Settings
{
get { return _Settings; }
}

#endregion
}
}
```

All these methods are executed in the Transaction mode.

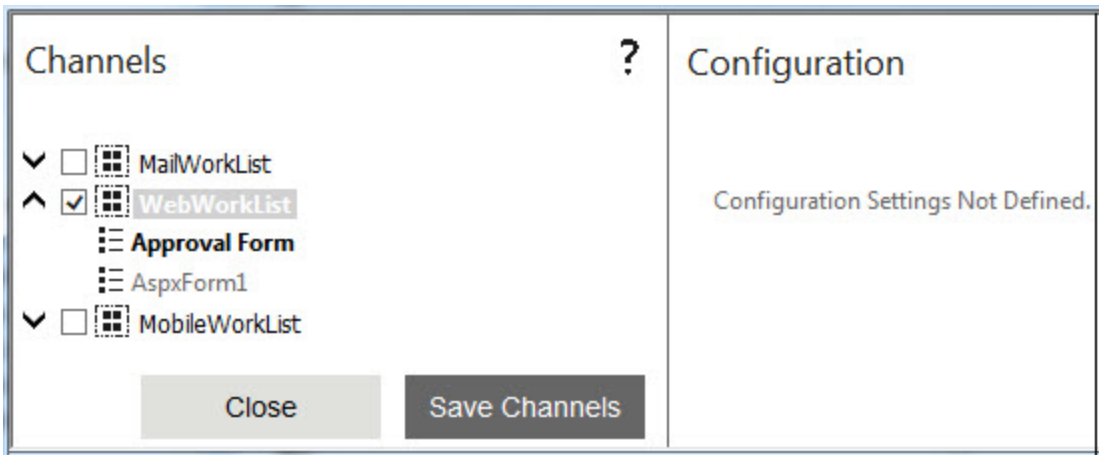
4. Compile the project and place the dll in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work TasksBPM.Net\bin* folder.
5. Configuration of Custom Work Item Form in the SKAddinProviders table of the Repository Database. Add a row in the SKAddInProviders table mentioning the type as Form and Name as return value of the method "FormType". i.e. CustomWorkItemForm as mentioned in the sample. The class name and the Assembly path should be provided accordingly. The following figure illustrates this entry:

```
Insert into
SKAddinProviders(Type,Name,Description,ClassName,Assembly,LastUpdatedDateTime)
values('Form','CustomWorkItemForm',NULL,'Skelta.HWS.WorkListChannel.Web.CustomWorkItemFo
rm',
'bin\CustomWorkItemForm.dll',getutcdate())
GO
```

Id	Type	Name	Descri...	ClassName	Assembly	Settings	IsGacAssembly	IsDefault	IsGlobal	LastUpdatedDateTime
66e15636-c8da-...	Form	CustomWorkItemForm	NULL	Skelta.HWS.WorkListChannel.Web.CustomWorkItemForm	bin\CustomWorkItemForm.dll	NULL	False	False	False	3/23/2010 12:45:55 PM
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

6. After the configuration of the Custom Work Item Form in the SKAddinProviders table, the CustomWorkItemForm will get displayed in the Delivery Channels Property, as shown below. The figure below shows the Approval Activity's Delivery Channels Property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



7. The messages put in the Custom Work Item form like Called for On Work Item Creation, etc can be seen in the Logger Console when we execute the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure Custom ASPX Property

Using the AVEVA Work Tasks defined interfaces, custom ASPX property can be created that can be used in any of the custom activity where this property can be used. This custom ASPX property will be available in the Activity Properties of the custom activity which is using this property.

Procedure

Dll Reference

```
System.Web
Workflow.NET.NET2
Workflow.NET.Web.Designer.NET2
```

Namespace Used

```
System
System.IO
System.Xml
System.Xml.XPath
System.Collections.Generic
System.Text
Workflow.NET
Workflow.NET.Engine
Workflow.NET.Interfaces
Workflow.NET.Web.Designer.Interfaces;
```

To create a Custom ASPX Property

1. Go to Start->Visual Studio 2005->New->Projects->Class Library->SampleCustomAspxProperty. This step will create a Class Library project in the path specified by the name SampleCustomAspxProperty.
2. Create two class files SampleCustomASPXProperty.cs and SampleCustomASPXPropertyWebUI.cs under the project.
3. There are two interfaces of AVEVA Work Tasks which will be used while developing a custom property. One is the IPropertyTypeWebUI interface and the other is the IPropertyType2 interface.
4. The IPropertyTypeWebUI interface is used to render the property in the Process Designer.
5. The IPropertyType2 interface is used for setting and getting the details of the property. IPropertytype2 interface inherits the IpropertyType interface internally.

   SampleCustomASPXProperty.cs Which Implements the IpropertyType2 Interface:

```
using System;
using System.IO;
using System.Xml;
using System.Xml.XPath;
using System.Collections.Generic;
using System.Text;
using Workflow.NET;
using Workflow.NET.Engine;
using Workflow.NET.Interfaces ;
namespace Workflow.NET.SampleAspxProperty
{
public class SampleCustomASPXPropertyType :IPropertyType2
{

public SampleCustomASPXPropertyType()
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}

string _defaultvalue;
bool _mandatory;
string _name;

InitializationContext _contextInfo;



#region IPropertyType2 Members


/// <summary>
/// This class provides the current running instance details of workflow at runtime
/// (e.g : ContextInfo.CurrentContext.ExecutionID).
/// This class is used in property for accessing the running details of workflow
/// like applicationname,workflowname,currentcontext.
/// </summary>

public InitializationContext ContextInfo
{
get
{
return _contextInfo;
}
set
{
_contextInfo = value;
}
}

#endregion


string _dropDownValue = "";




/// <summary>
/// Drop down value captured from the ASPX page ui.
/// </summary>

public string DropDownValue
{
get
{
return _dropDownValue;
}

set
{
_dropDownValue = value;
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}

string _textBoxValue;

/// <summary>
/// Text box value captured from the ASXP page ui
/// </summary>

public string TextBoxValue
{
get
{
return _textBoxValue;
}

set
{
_textBoxValue = value;
}

}



#region IPropertyType Members

/// <summary>
/// Clearing the value of the custom property.
/// </summary>
public void Clear()
{
_dropDownValue = "";
if (string.IsNullOrEmpty(_defaultvalue))
_dropDownValue = _defaultvalue;

}

/// <summary>
/// Clones the property instance.
/// Improper implementation of clone method could result in an unexpected value
display.
/// We have to make sure the required properties  are set from the clone method.
/// </summary>

public IPropertyType Clone()
{
SampleCustomASPXPropertyType oSampleAspxProperty = new
SampleCustomASPXPropertyType();
oSampleAspxProperty._mandatory = _mandatory;
oSampleAspxProperty._defaultvalue = _defaultvalue;
oSampleAspxProperty._name = _name;
oSampleAspxProperty._dropDownValue = _dropDownValue;
oSampleAspxProperty._textBoxValue = _textBoxValue;

return oSampleAspxProperty;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}

/// <summary>
/// Name of the Custom Property
/// </summary>
public string Name
{
get
{
return _name;
}
set
{
_name = value;
}
}

/// <summary>

/// If there is some definition which needs to be passed and read for initializing
the property.
/// The definition is defined along with the property in the actions.xml.
///</summary>
public void ReadDefinitionXml(XmlTextReader xtr)
{
_defaultvalue = xtr.GetAttribute("defaultvalue");
string mandatory = xtr.GetAttribute("mandatory");
if (mandatory != null)
{

if (mandatory.ToUpperInvariant() == "TRUE")
_mandatory = true;
else
_mandatory = false;

}


((IPropertyType)this).Clear();

}

/// <summary>
/// The ReadXml method is called whenever the action which is having the property is
loaded from process designer.
/// The ReadXml method is also gets called when the property is getting accessed at
the runtime.
/// </summary>

public void ReadXml(XmlTextReader xtr)
{
while (xtr.Read())
{

if (xtr.NodeType == XmlNodeType.EndElement && xtr.Name == "property") break;
if (xtr.NodeType == XmlNodeType.EndElement && xtr.Name == "aspxproperty") break;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (xtr.NodeType == XmlNodeType.Element && xtr.Name == "dropdownvalue")
{
_dropDownValue = xtr.ReadElementString();
}

if (xtr.NodeType == XmlNodeType.Element && xtr.Name == "textboxvalue")
{

_textBoxValue = xtr.ReadElementString();

}

}

}

public override string ToString()
{
return Value;
}

public string ToString(string XPath)
{
throw new Exception("The method or operation is not implemented.");
}

public string ToString(bool ReturnXml)
{
throw new Exception("The method or operation is not implemented.");
}

/// <summary>
///  This property is used to set  or get the Value for the custom property.
///  Using this property the value of the property can be accessed at runtime.
/// </summary>
public string Value
{
get
{

if (string.IsNullOrEmpty(_dropDownValue))
{

if (!string.IsNullOrEmpty(_defaultvalue))

_dropDownValue = _defaultvalue;

}

return _dropDownValue;

}

set
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{

string ErrorString = "";

if (((IPropertyType)this).isValidValue(value, ref ErrorString))

_dropDownValue = value;

else

throw new System.Exception(ErrorString);

}

}

/// <summary>
/// This method is used to write the property value data in the Process Definition in
the XML format.
/// This method will be called when the activity in which this property is used is
rendered.
/// This method is used to persist data which is saved after rendering the property
in process designer
/// and can be used in runtime.
/// The value persisted here is read back from the ReadXml method of the
IPropertytype interface.
/// </summary>
public void WriteXml(XmlTextWriter xtw)
{
xtw.WriteStartElement("aspxproperty");
xtw.WriteStartElement("dropdownvalue");
xtw.WriteCData(_dropDownValue);
xtw.WriteEndElement();
xtw.WriteStartElement("textboxvalue");
xtw.WriteCData(_textBoxValue);
xtw.WriteEndElement();
xtw.WriteEndElement();
}

/// <summary>
/// Used to check the whether the property value is valid or not.
/// The cross dependent validation can be done at the human activity level
/// which has implemented the IValidateAction interface.
/// </summary>
/// <param name="Value">Property value</param>
/// <param name="ErrorString">Error string returned if the value is not
valid.</param>
/// <returns>true or false</returns>
public bool isValidValue(string Value, ref string ErrorString)
{
bool valid = false;

if (_mandatory)
{
if (Value != null)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (Value.Trim() != "")
valid = true;

}

if (!valid)
{
if (ErrorString != null)

ErrorString = "Value cannot be left empty [" + _name + "]";

_dropDownValue = "";

}

}
else
valid = true;

return valid;
}

#endregion
}
}
```

SampleCustomASPXPropertyWebUI.cs which implements the IPropertyTypeWebUI interface

```
using System;
using System.Web;
using System.Web.UI;
using System.Collections.Generic;
using System.Text;
using Workflow.NET.Web.Designer.Interfaces;
using Workflow.NET.Web.Designer;
using Workflow.NET;
namespace Workflow.NET.SampleAspxProperty
{
/// <summary>
/// Class which implements IPropertyTypeWebUI interface for rendering the custom
property
/// in the process designer when it is used within the Custom Activity.
/// </summary>

 public class SampleCustomASPXPropertyWebUI :IPropertyTypeWebUI
{
 public SampleCustomASPXPropertyWebUI()
 {

 }

#region IPropertyTypeWebUI Members


 /// <summary>
 /// This method is used to render the UI for the Custom Aspx Property in the process
designer.
 /// This method has the logic of specifying the custom aspx page which will open on
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
click of the button.
 /// </summary>

 /// <param name="oAction">Action class details to which this custom property
belongs.</param>
 /// <param name="oProperty">Property object which is used for getting value,name so
that the value will be retained
 ///  if you are loading the action after setting the property value.</param>
 /// <param name="oWebDesigner">ProcessDesigner control to which the control used in
the custom property is added.
 ///</param>
public void Draw(Action oAction, Property oProperty, ProcessDesigner oWebDesigner)
{
string output;
ProcessDesignerAdapter pda = new ProcessDesignerAdapter(oWebDesigner);
//To get the workflow related Querystring Parameters.This method is used in ASPX
property types which is getting opened from the Process Designer.
string s = pda.GetUrlParameters(false);
String URL = oWebDesigner.GetTemplateRelativeUrl("CustomAspx.aspx") + "?ActionName="
+
Workflow.NET.CommonFunctions.HandleSplCharactersForLocalization(System.Web.HttpUtilit
y.UrlEncode(oAction.Name)) + "&selPropertyName=" +
Workflow.NET.CommonFunctions.HandleSplCharactersForLocalization(System.Web.HttpUtilit
y.UrlEncode(oProperty.Name)) + "&" + s;

output = "<TABLE class='propertiesboxinputstable' cellpadding=0 cellspacing=0
width=100%>";
output += "<TR>";
Workflow.NET.SampleAspxProperty.SampleCustomASPXPropertyType AspxPropertyType =
((Workflow.NET.SampleAspxProperty.SampleCustomASPXPropertyType)oProperty.PropertyHand
ler);

string propValue = AspxPropertyType.Value;
if (!string.IsNullOrEmpty(propValue))
output += "<TD>" +
CommonFunctions.AddElipses(System.Web.HttpUtility.HtmlEncode(propValue), 20) +
"</TD>";
else
output += "<TD>" +
CommonFunctions.AddElipses(oWebDesigner.ActionsDefinition.GlobalResourceSet.GetString
("CustomASPXPropertyValue"), 20) + "</TD>";

output += "<TD width=20><input type=button text=W class=propertiesboxinputbutton
style=\"width:15;height:15\"  onclick=\"window.open('" + URL +
"','AspxProperty','height=395,width=790,resizable=no');\"></TD>";

output += "</TR>";
output += "</TABLE>";
oWebDesigner.Controls.Add(new LiteralControl(output));

}

 public PropertyPage CreatePropertyPageInstance()
 {
 return null;
 }
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
public bool HandlePostBack(Action oAction, Property oProperty, ProcessDesigner
oWebDesigner, ref string ErrorString)
{
bool valid = false;
ErrorString = "";
if (oProperty.PropertyHandler.isValidValue(oProperty.Value.ToString(), ref
ErrorString))
{
valid = true;
}
return valid;
}

#endregion
}
}
```

6. Build the Project.
7. Add the dll of the Project to the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPM.NET\bin* and the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPM.NET\Web\EnterpriseConsole\bin*
8. Create an ASPX Page **"CustomASPX.aspx"** for showing the UI for the property.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="CustomASPX.aspx.cs"
Inherits="BPMUITemplates_Default_ProcessDesigner_CustomASPX" %>

<html>

<head id="Head1" runat="server">

<title><%=sWorkflow%>-<%=actionname%>-<%=propertyname%></title>

<link href="<%=ProcessDesignerControl.GetCommonRelativeUrl("stylesheet/
global.css")%>" rel="stylesheet" type="text/css" />

</head>

<body>

<form id="form1" runat="server">

<br />

<div>

<table border="0" cellpadding="4" cellspacing="1" height="10%" width="100%"
class='tablebg'>

<tr>

<td class="header2" colspan=2>
<font class="subtitle">Custom ASPX Property</font>
</td>

</tr>

<tr bgcolor=white>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<td style="width: 187px">

<asp:Label ID="Label1" runat="server" Text="Label" CssClass="lefttdbg"><font
class="subtitle">Select the value:</font></asp:Label>

</td>

<td>
<asp:DropDownList ID="DropDownList1" runat="server"
CssClass="inputselect"></asp:DropDownList><br />
</td>

</tr>

<tr bgcolor=white>

<td style="width: 187px">

<asp:Label ID="Label2" runat="server" Text="Label" CssClass="lefttdbg"><font
class="subtitle">Enter Value:</font> </asp:Label>

</td>

<td>

<asp:TextBox ID="TextBox1" runat="server" CssClass="inputtext"></asp:TextBox>

 </td>

</tr>

<tr bgcolor=white>

<td colspan=2 align=center>

<asp:Button ID="Button1" runat="server" Text="Submit" CssClass="inputbutton"
OnClick="Button1_Click"/>

</td>

</tr>

</table>

<br />

</div>

</form>

</body>

</html>
```

9. Create a Codebehind file for Custom aspx page CustomASPX.aspx.cs

```
using System;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using Workflow.NET;
using Workflow.NET.Web.Designer;
public partial class BPMUITemplates_Default_ProcessDesigner_CustomASPX :
System.Web.UI.Page
{

protected Property selProperty = null;

protected ProcessDesigner ProcessDesignerControl;
protected ProcessDesignerAdapter pda;
protected string actionname = "";
protected string propertyname = "";
protected string sWorkflow = "";


Workflow.NET.SampleAspxProperty.SampleCustomASPXPropertyType aspxProperty = null;

/// <summary>
/// Page Load Method.
/// Reading the Querystring parameters passed from the Draw Method of the Custom ASPX
Property.
/// Filling of Drop down which is used in the Custom Aspx page.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void Page_Load(object sender, EventArgs e)
{

if (!Page.IsPostBack)
{

actionname = this.Context.Request["ActionName"];
propertyname = this.Context.Request["selPropertyName"];


this.ViewState["actionname"] = actionname;
this.ViewState["selpropertyname"] = propertyname;


}

else
{
actionname = this.ViewState["actionname"].ToString();
propertyname = this.ViewState["selpropertyname"].ToString();

}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
pda = new ProcessDesignerAdapter();

//To get the Process Designer details  corresponding to the current application
name,workflow name. This can be used in the ASPX pages which is getting opened from
the Process designer say menu .In the Code behind of ASPX page you will require
workflow related details like Application Name,Workflow Name and which is the current
action.
pda.LoadControlInstance(false);
ProcessDesignerControl = pda.ProcessDesignerControl;
sWorkflow = pda.ProcessDesignerControl.WorkflowName;
//Getting the Action Object from the Process Designer Control process definition
object which contains
//the details definition of the workflow by passing the action name
Workflow.NET.Action selAction =
ProcessDesignerControl.ProcessDefinition.Actions[actionname];
//Getting the property details based on the property name passed from the action
object
selProperty = selAction.Properties[propertyname];
aspxProperty =
(Workflow.NET.SampleAspxProperty.SampleCustomASPXPropertyType)selProperty.PropertyHan
dler;

if (!Page.IsPostBack)
{

if (!string.IsNullOrEmpty(aspxProperty.TextBoxValue))
 TextBox1.Text = aspxProperty.TextBoxValue;

DropDownList1.Items.Add(new ListItem("Orange", "Orange"));
DropDownList1.Items.Add(new ListItem("Yellow", "Yellow"));
DropDownList1.Items.Add(new ListItem("Green", "Green"));
DropDownList1.Items.Add(new ListItem("Blue", "Blue"));
DropDownList1.Items.Add(new ListItem("White", "White"));

if (!string.IsNullOrEmpty(aspxProperty.DropDownValue))
{
ListItem selectedItem = DropDownList1.Items.FindByValue(aspxProperty.DropDownValue);
selectedItem.Selected = true;

}
}

}

/// <summary>
/// On click Event handling of button
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void Button1_Click(object sender, EventArgs e)
{
//Assigning to the Custom Property drop down value property with the selected
//dropdown value shown in the custom ASPX page UI
aspxProperty.DropDownValue = DropDownList1.SelectedItem.Value;
//Assigning to the Custom Property text value with the text box value shown in the
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
custom ASPX page UI
aspxProperty.TextBoxValue = TextBox1.Text;

//Writing the property values in the Xml format as defined per the WriteXML method of
the Custom
//Aspx property type
ProcessDesignerControl.WriteFile();
//To Refresh the Selected action so that the updated value appears in the property
and closing the window which shows the custom aspx page UI.
Response.Write("<script language='javascript'>
window.setTimeout('window.opener.refreshselectedaction()',100);window.opener.parent._
skProcessDesigner.RefreshProperties();window.setTimeout('window.close()',100);
</script>");

}

}
```

10. Copy the aspx page (CustomASPX.aspx and code behind) that is created to the web application BPMUITemplates\Default\ProcessDesigner folder. The sample is built to load the Custom ASPX page from this particular folder.
11. Update the property in Actions.XML under <properties> tag. Add the following tag in actions.xml under <properties> tag.

```
<propertytype name="SampleAspxProperty"
classname="Workflow.NET.SampleAspxProperty.SampleCustomASPXPropertyType"
assemblyname="Bin\SampleCustomAspxProperty.dll"
webuiclassname="Workflow.NET.SampleAspxProperty.SampleCustomASPXPropertyWebUI"
webuiassemblyname="Bin\SampleCustomAspxProperty.dll"/>
```

Now the property can be included in any custom activity under <properties> tag of Custom Activity.

For example:

```
<action name="CustomeActivity" displayname="Custom Activity">
<description>Custom Activity.</description>
<image resourcename="dummyaction.png"></image>
<handler classname="CustomeActivity.Class1" assembly="bin\
CustomeActivity.dll"></handler>
<properties>
<property name="CustomActivity Property" type="aspxproperty"></property>
</properties>
</action>
```

Custom Property

Using the AVEVA Work Tasks defined interfaces, custom property can be created that can be implemented by any custom activity. This custom property will be available in the Activity Properties of the custom activity implementing this.

Procedure

Dll Reference

```
System.Web
Workflow.NET.NET2
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Workflow.NET.Web.Designer.NET2
Namespace Used
System
System.IO
System.Xml
System.Xml.XPath
System.Collections.Generic
System.Text
Workflow.NET
Workflow.NET.Engine
Workflow.NET.Interfaces
Workflow.NET.Web.Designer.Interfaces
```

Note: When you package a workflow with custom activity, if you need to correct the custom activity in the target repository, follow the steps mentioned in "*Resolving Custom Activities*" and "*Correcting Custom Activities*". For more information, see Resolve Custom Activities and Correct Custom Activities.

To create Custom Property

1. Go to Start->Visual Studio 2005->New->Projects->Class Library->SampleProperty. This step will create a Class Library project in the path specified by the name SampleProperty.
2. Create two class files under the project SampleProperty.cs and SamplePropertyWebUI.cs
3. There are two interfaces of AVEVA Work Tasks which will be used while developing a custom property. One is IPropertyTypeWebUI interface and the other is IPropertyType2 interface.
4. The IPropertyTypeWebUI interface is used to render the property in the Process Designer.

   Note: Add a reference to Skelta.Bpm.Quickflow.Bridge.dll.
5. The IPropertyType2 interface is used for setting and getting the details of the property. IPropertytype2 interface internally inherits the IpropertyType interface.

   SampleProperty.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using System.Xml.XPath;
using System.IO;
using Workflow.NET;
using Workflow.NET.Engine;
using Workflow.NET.Interfaces;


namespace Workflow.NET.SampleProperty
{
/// <summary>
 /// Class for Custom Property which implements the IPropertyType2 interface and also
Inherits the ExpressionHandler Class
/// </summary>
public class SampleProperty : ExpressionHandler, IPropertyType2
{

string _value;
string _defaultvalue;
bool _mandatory;
string _name;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
string _choicestring;
InitializationContext _contextInfo;
/// <summary>
/// Initializes a new instance of SampleProperty class.
/// </summary>
public SampleProperty()
{

}
#region IPropertyType2 Members



 /// <summary>
/// The class  InitializationContext provides the current running instance details of
workflow at /// runtime. Used to resolve variable or contents used during runtime.
 (e.g :  /// ContextInfo.CurrentContext.ExecutionID) and also design time details.
/// This class is used in property for accessing the design time details
/// like applicationname,workflowname,database details,Workflow.Net.config object.
 /// </summary>
 InitializationContext IPropertyType2.ContextInfo
{
get
{
return _contextInfo;
}

set
{
_contextInfo = value;
}
}

#endregion

#region IPropertyType Members

/// <summary>
/// Clearing the value of the custom property.
/// </summary>
void IPropertyType.Clear()
{
_value = "";
if (_defaultvalue != null)
if (_defaultvalue != "")
_value = _defaultvalue;
}

/// <summary>
/// Clones the property instance.
/// Improper implementation of clone method could result in an unexpected value
display.
/// We have to make sure the required properties  are set from the clone method.
/// </summary>
/// <returns></returns>
IPropertyType IPropertyType.Clone()
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
SampleProperty oSampleProperty = new SampleProperty();
oSampleProperty._mandatory = _mandatory;
oSampleProperty._defaultvalue = _defaultvalue;
oSampleProperty._name = _name;
oSampleProperty._value = _value;
oSampleProperty._choicestring = _choicestring;
return oSampleProperty;
}

/// <summary>
/// Name of the Custom Property
/// </summary>
string IPropertyType.Name
{
get
{
return _name;
}

set
{
_name = value;
}
}

/// <summary>
/// This is specific to this custom property which is used
/// to fill the predefined values for the property.
/// </summary>
public string ChoiceString
{
get
{
return _choicestring;
}
}


/// <summary>

/// If there is some definition which needs to be passed and read for initializing
the property.
/// The definition is defined along with the property in the actions.xml.

/// (for ex: In the property below (sample) configuration, the nodes choice is the
definition for the property and which is used for filling the property default
values.
///</summary>
/*
<property name="Sample Property" type="sample" defaultvalue="N">
 <choice>Yes;Yes</choice>
 <choice>No;No</choice>
 </property>
*/
void IPropertyType.ReadDefinitionXml(System.Xml.XmlTextReader xtr)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
_defaultvalue = xtr.GetAttribute("defaultvalue");
string mandatory = xtr.GetAttribute("mandatory");
if (mandatory != null)
{
if (mandatory.ToUpperInvariant() == "TRUE")
_mandatory = true;
else
_mandatory = false;
}

_choicestring = "";

while (!xtr.EOF)
{
xtr.MoveToElement();
if (xtr.NodeType == XmlNodeType.EndElement && xtr.Name == "property") break;
if (xtr.NodeType == XmlNodeType.Element)
{
if (xtr.Name == "choice")
{
string _choicestringdisp = xtr.GetAttribute("displayas");
string _choicestringValue = xtr.ReadElementString();

if (_choicestringdisp == null || _choicestringdisp == "")
_choicestringdisp = _choicestringValue;

_choicestring += (_choicestring == "" ? "" : ",") + _choicestringdisp + ";" +
_choicestringValue;
}
}
xtr.Read();
}

}


/// <summary>
 /// The ReadXml method is called whenever the action which is having the property is
loaded from process designer.
/// The ReadXml method is used to read the Xml data persisted in the process
definition.
 /// ReadExpressionXml method reads the defined expression for the property from the
/// Xml format and stored in the property Codeclassname and Expressionstring while
loading the action.
/// </summary>

void IPropertyType.ReadXml(System.Xml.XmlTextReader xtr)
{
if (!ReadExpressionXml(xtr))
_value = xtr.ReadElementString();
}
/// <summary>
/// Gets a string that represents the property. Used Internally while rendering
report.
/// </summary>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// <returns></returns>
public override string ToString()
{
return Value;
}

/// <summary>
/// Gets a string that represents the property. Used internally while rendering
report.
/// </summary>
/// <param name="XPath">The xpath expression for getting a specific string.</param>
/// <returns>A string that represents the property.</returns>

public virtual string ToString(string XPath)
{
string strValue = "";
string strXml = ToString(true);
XPathDocument xpathDoc = new XPathDocument(new StringReader(strXml));
XPathNavigator xpathNavigator = xpathDoc.CreateNavigator();
XPathNodeIterator iterator = xpathNavigator.Select(XPath);
while (iterator.MoveNext())
{
strValue += iterator.Current.Value;
}
return strValue;
}
/// <summary>
/// Gets a string that represents the property. Used internally while rendering
report.
/// </summary>
/// <param name="ReturnXml">Determines whether the string should be returned in XML
format.</param>
 /// <returns>A string that represents the property.</returns>
public virtual string ToString(bool ReturnXml)
{
string strValue = "";
if (ReturnXml)
{
MemoryStream stream = new MemoryStream();
StreamWriter writer = new StreamWriter(stream);
XmlTextWriter xmlWriter = new XmlTextWriter(writer);
xmlWriter.WriteStartElement("property");
xmlWriter.WriteAttributeString("name", _name);
xmlWriter.WriteAttributeString("value", Value);
xmlWriter.WriteAttributeString("type", "SampleProperty");
xmlWriter.WriteEndElement();
xmlWriter.Flush();
stream.Position = 0;
StreamReader reader = new StreamReader(stream);
strValue = reader.ReadToEnd();
stream.Close();
}
else
{
strValue = ToString();
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
return strValue;
}


/// <summary>
///  This property is used to set  or get the Value for the custom property.
///  Using this property the value of the property can be accessed at runtime.
 ///  This method is used to evaluate the expression value defined in the property.
 ///  We would have defined variable or content or xml variables or hard coded value
in the expression builder
 /// for a property in an action.
 /// GetProcessedValue method is used to evaluate these variable or content or
xmlvariable or hard coded value
/// and assigned as the value to the property.
/// </summary>
public string Value
{
 get
{
if (_value == null || _value == "")
{
if (_defaultvalue != null && _defaultvalue != "")
_value = _defaultvalue;
}

if (this.IsExpression)
{
if (_contextInfo != null)
{
if (_contextInfo.CurrentContext != null)
return this.GetProcessedValue(_contextInfo.CurrentContext).ToString();
else
return _value;
}
else
return _value;
}
else
return _value;
}

set
{
string ErrorString = "";
if (((IPropertyType)this).isValidValue(value, ref ErrorString))
_value = value;
else
throw new System.Exception(ErrorString);

 }
}

/// <summary>
/// This method is used to write the property value data in the Process Definition in
the XML format.
/// This method will be called when the activity in which this property is used is
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
rendered.
/// This method is used to persist data which is saved after rendering the property
in process designer
/// The value persisted here is read back from the ReadXml method of the
IPropertytype interface.
/// WriteExpressionXml method writes the defined expression for the property in  Xml
format.
/// This XML contains the details of the expression defined and associated class name
 /// of the compiled code.
/// </summary>

void IPropertyType.WriteXml(System.Xml.XmlTextWriter xtw)
{
if (!WriteExpressionXml(xtw))
  xtw.WriteCData(_value);
}

 /// <summary>
/// Used to check the whether the property value is valid or not.
 /// Gets called while process definition is validated on deploy of the workflow.
 /// IsExpression property is used to check Whether any expression
/// is defined for a property for an action.
/// </summary>
/// <param name="Value"></param>
/// <param name="ErrorString"></param>
/// <returns></returns>
bool IPropertyType.isValidValue(string Value, ref string ErrorString)
{
bool valid = false;
if (this.IsExpression)
{
return true;
 }
if (_mandatory)
{
if (Value != null)
{
if (Value != "" && Value != "0")
valid = true;
}
if (!valid)
{
if (ErrorString != null)
{
SkeltaResourceSetManager resourceManager = new SkeltaResourceSetManager();
ErrorString = resourceManager.GlobalResourceSet.GetString("Value cannot be left
empty") + " [" + resourceManager.GlobalResourceSet.GetString(_name) + "]";
}
_value = "";
}
}
else
valid = true;
return valid;
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
#endregion
}
}
```

SamplePropertyWebUI.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Web.UI;
using System.Web.UI.WebControls;
using Workflow.NET;
using Workflow.NET.Web.Designer.Interfaces;

namespace Workflow.NET.SampleProperty
{
/// <summary>
/// Class which implements IPropertyTypeWebUI interface for rendering the custom
property
/// in the process designer.
/// </summary>
public class SamplePropertyWebUI :IPropertyTypeWebUI
{

/// <summary>
/// Initializes a new instance of SamplePropertyWebUI class.
/// </summary>
public SamplePropertyWebUI()
{

}
#region IPropertyTypeWebUI Members


/// <summary>
///  This method is used to render the UI for the Property (The right pane which gets
displayed /// for an activity to display it's collection  of properties) in the
process designer.
/// </summary>

/// <param name="oAction">Action class details to which this custom property
belongs.</param>
/// <param name="oProperty">Property object which is used for getting value,name so
that the value will be retained
///  if you are loading the action after setting the property value.</param>
/// <param name="oWebDesigner">ProcessDesigner control to which the control used in
the custom property is added.
/// here in this custom property dropdownlist control is added.</param>
public void Draw(Action oAction, Property oProperty,
Workflow.NET.Web.Designer.ProcessDesigner oWebDesigner)
{
string output;
Workflow.NET.SampleProperty.SampleProperty oSampleProperty =
 (Workflow.NET.SampleProperty.SampleProperty)oProperty.PropertyHandler;


output = "<TABLE class='propertiesboxinputstable' cellpadding=0 cellspacing=0
width=100%>";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
output += "<TR><TD>";
oWebDesigner.Controls.Add(new LiteralControl(output));

string[] Choices = oSampleProperty.ChoiceString.Split(new char[] { ',' });
DropDownList oChoice = new DropDownList();
oChoice.CssClass = "propertiesboxinputs";
oChoice.ID = "ActionProperty" + oAction.Name + oProperty.Name + "SampleProperty";
oChoice.Items.Add(new
ListItem(oWebDesigner.ActionsDefinition.GlobalResourceSet.GetString("Select Option"),
"0"));
foreach (string sItem in Choices)
{
string[] valueArr = sItem.Split(';');
oChoice.Items.Add(new
ListItem(oWebDesigner.ActionsDefinition.GlobalResourceSet.GetString(valueArr[0]),
valueArr[1]));
}
bool selected = false;
if (oProperty.Value != null)
{

ListItem oSelectedItem = oChoice.Items.FindByValue((string)oProperty.Value);
if (oSelectedItem != null)
{
oSelectedItem.Selected = true;
selected = true;
}
}
oWebDesigner.Controls.Add(oChoice);
output = "</TD></TR></TABLE>";
oWebDesigner.Controls.Add(new LiteralControl(output));

}


/// <summary>
/// This is to handle the Custom Property details after clicking on Save button from
the Activity
/// which is using this custom property in the Process designer.
///
/// </summary>
/// <param name="oAction">Action class details to which this custom property
belongs.</param>
/// <param name="oProperty">Property object which is used for getting value,name so
that the value will be retained</param>
/// <param name="oWebDesigner">Getting the Control used for the custom property from
the ProcessDesigner control.In this custom
/// property we have used dropdown list control.</param>
/// <param name="ErrorString">Checking for error in the input for the custom
property.</param>
/// <returns>true if the input is valid; else false</returns>

public bool HandlePostBack(Action oAction, Property
oProperty, Workflow.NET.Web.Designer.ProcessDesigner oWebDesigner, ref string
ErrorString)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
bool valid = false;
ErrorString = "";
DropDownList oChoice = (DropDownList)oWebDesigner.FindControl("ActionProperty" +
oAction.Name + oProperty.Name + "SampleProperty");
if (oChoice != null)
{
if (oProperty.PropertyHandler.isValidValue(oChoice.SelectedItem.Value, ref
ErrorString))
{
oProperty.Value = oChoice.SelectedItem.Value;
valid = true;
}
}

return valid;
}

/// <summary>
/// This method is used where in we need to render as html.
/// </summary>
/// <returns></returns>
public Workflow.NET.Web.Designer.PropertyPage CreatePropertyPageInstance()
{
return null;
}

#endregion
}
}
```

6. Build the Project.
7. Place the dll in the AVEVA Work Tasks Installed path\BPM.NET bin directory.
8. Configuration of Custom Property in the Actions.XML file (which will be located in WorkflowElements\Default\en-us\Actions\XML). We need to add the custom property under the properties tag with the classname,assemblyname,webuiclassname and webuiassemblyname for the custom property.

```
<propertytype name="SampleProperty"
classname="Workflow.NET.SampleProperty.SampleProperty"
assemblyname="bin\SampleProperty.dll"
webuiclassname="Workflow.NET.SampleProperty.SamplePropertyWebUI"
webuiassemblyname="bin\SampleProperty.dll"/>
```

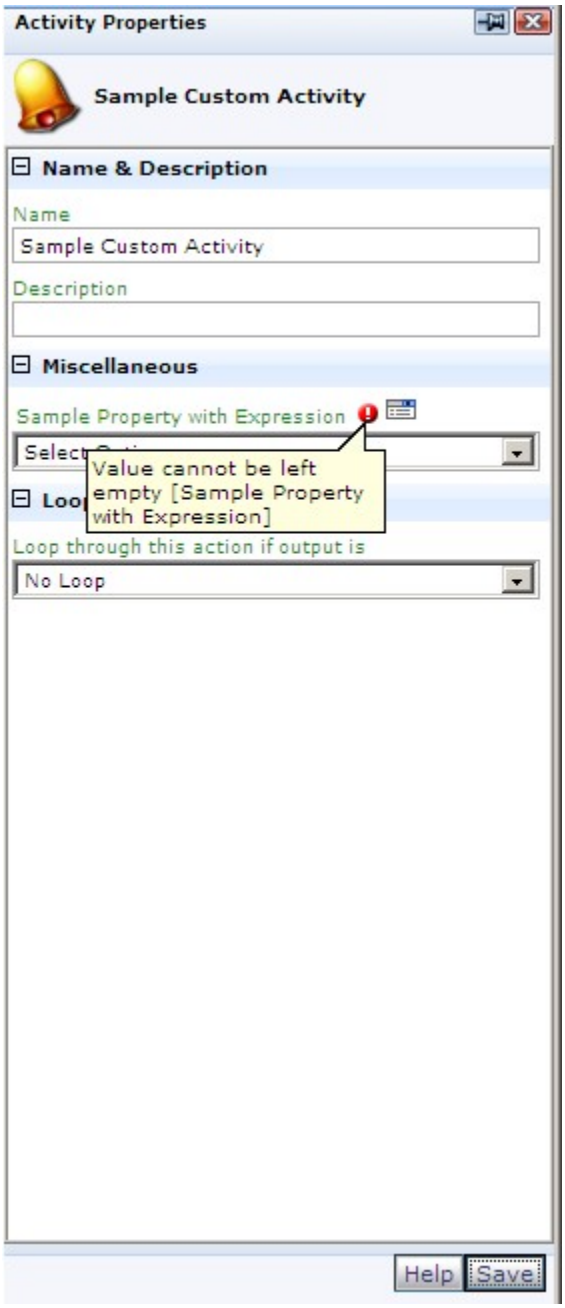
9. Configuration of Custom Property for the Custom Engine Activity (Specified in the next topic) in the Actions.XML file.

```
<action name="CustomActivity" displayname="Custom Activity">
 <description>Custom activity. </description>
 <image resourcename="Customac.png"></image>
 <handler classname="SampleCustomAction.CustomAction" assembly="bin\
SampleCustomAction.dll"></handler>
 <properties>
 <property name="Sample Property with Expression" type="SampleProperty"
mandatory="true">
 <choice>Yes;Yes</choice>
 <choice>No;No</choice>
 </property>
 </properties>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

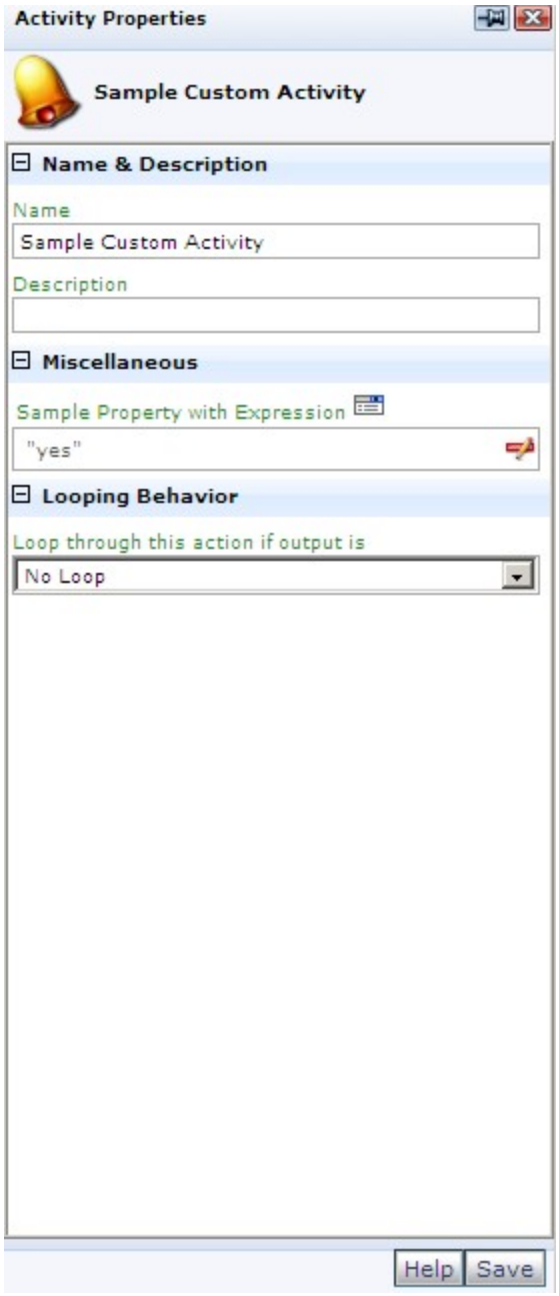
```
    <return>
    <value helpstring="Success">Success</value>
    <value helpstring="UnSuccess">UnSuccess</value>
    </return>
   </action>
```

10. After the configuration of the custom property in the Actions.xml the Property will appear in the Process Designer for the specified custom activity as below. See that there is a mandatory icon that gets displayed when it is viewed from the Process Designer. This indicates that without providing the value to this property we cannot publish the workflow. The property is made mandatory using the attribute mandatory="true".



The following figure shows the property value set through the Expression Editor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Custom Engine Activity

AVEVA Work Tasks provides an interface called IActionRun which can be implemented in the class file of a custom activity to develop a AVEVA Work Tasks engine activity.
The Engine Activity gets executed in Runtime by the Workflow Engine exe. No human intervention is required while executing the Engine Activity.

Procedure

Dll Reference

```
Workflow.NET.NET2
```

SampleProperty

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Namespace Used

```
System
System.Collections.Generic
System.Text
Workflow.NET
Workflow.NET.Interfaces
Workflow.NET.Engine
Workflow.NET.Engine.Interfaces
```

To create Custom Activity

1. Go to Start->Visual Studio->New->Projects->Class Library->SampleCustomAction. This step will create a Class Library project in the path specified by the name SampleCustomAction.
2. Add the following references:

   Workflow.NET.NET2.dll from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\bin*

   SampleProperty which is created in the previous topic of "Custom Property" from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\bin*
3. Create a class file CustomAction.cs under the project.
4. Copy the following code in CustomAction.cs and build the project.

   CustomAction.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using Workflow.NET;
using Workflow.NET.Interfaces;
using Workflow.NET.Engine;
using Workflow.NET.Engine.Interfaces;
namespace SampleCustomAction
{
public class CustomAction :IActionRun
{
#region IActionRun Members

/// <summary>
/// This is maintained for backward Compatibility
/// </summary>
/// <returns></returns>
public IActivityWebUIHandler GetActivityWebUIHandler()
{
throw new Exception("The method or operation is not implemented.");
}


//// <summary>
/// The AVEVA Work Tasks workflow engine exe service calls this method to execute the
action.
/// </summary>
/// <param name="ExecutionID">Execution Id for the workflow for the current running
workflow instance.</param>
/// <param name="ExecutionDetailsID">Execution details id for the current running
action for the particular running workflow instance.</param>
/// <param name="CurrentContext">Workflow engine context which contains the details
of the Workflow.</param>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/// <param name="CurrentAction">Current action object, This object has entire details
like properties collection used in this action. We can access the property value from
the properties collection using the CurrentAction object.</param>
/// <param name="InlinkOutput">This is the Previous Action's output which leads to
this action.</param>
/// <param name="Retrying">Indicates whether attempts will be made to reexecute this
action. Will always be false.</param>
/// <param name="Output">
/// Output of the current action. This is the return value of the current action.
/// The actions that have conditional routing must assign a value to this parameter.
/// The workflow engine routes the workflow execution based on this output.
/// </param>
/// <returns>The Return Value of the Run method is ActionResult. There are two modes
of Action Execution one is Completed  and the other one is Sleep mode. If the Return
value is ActionResult.Completed then the Engine will alert the specified output  and
the status of the Action will be completed in database. If the Return value is
ActionResult. Sleep then the Engine will alert the specified output and the Action
status will not be completed in the database.</returns>
/// <remarks>
/// The workflow engine executes an action by calling this method for the action.
/// Workflow.NET provides a set of inbuilt actions. The list of actions can be
/// extended for an application by developing the custom actions that implement the
/// IActionRun interface. The custom action should to be added to the
/// Actions.xml in the workflow elements directory for the application, to be able
/// to use it in the process design.
/// </remarks>
public ActionResult Run(int ExecutionID, int ExecutionDetailsID, Context
CurrentContext, Workflow.NET.Action CurrentAction, string InlinkOutput, bool
Retrying, out string Output)
{
string customActivityPropertyValue = "";

//First Check for the Existence of the property in the Current Action this check is
must
//for backward compatibility reasons and for any change in Actions.xml

if (CurrentAction.Properties.Contains("Sample Property with Expression"))
{
Property prop = (Property)CurrentAction.Properties["Sample Property with
Expression"];


Workflow.NET.SampleProperty.SampleProperty sampleProperty =
(Workflow.NET.SampleProperty.SampleProperty)prop.PropertyHandler;

//Gets the value entered for the property
customActivityPropertyValue = sampleProperty.Value;

}
//Displays the property value information in the Logger

CurrentContext.log.LogInformation("@#@ The Value of the sample property set in the
custom activity is :" + customActivityPropertyValue);


//Output of the current action. This is the return value of the current action.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//Suppose if you want to have conditional routing must assign a different value to
this Output parameter.

if (customActivityPropertyValue.ToUpperInvariant() == "YES")
Output = "Success";
else
Output = "UnSuccess";

return ActionResult.Completed;

}

#endregion
}
}
```

5. Place the dll in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasksbin* folder.
6. Open the **Actions.xml** file from *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Actions\XML* and configure for the **Custom Activity** under **Custom Activities** category.
   It is recommended to save the Custom Activity definition as a separate XML file. This ensures that when you log on to the Enterprise Console after an upgrade, the existing Custom Activity definition is retained. You can use the following sample code as a reference while creating the XML file.

```
<?xml version="1.0" encoding="utf-8" ?>
<actionsdata>
<version number="3.0"/>
<category name= "My Custom Activities" sortorder="10"  image="custom.gif">
<action name="AbortWorkfow">
<description>Used to abort a workflow</description>
<image resourcename="icon-Control-Runtime-Workflow.png"></image>
<handler classname="CustomAction.AbortWorkflow" assembly="C:\ProgramFiles\Skelta\
BPM.NET\Bin\CustomActions\CustomAction.dll"></handler>
<properties>
<property name="Application" displayname="Repository Name" type="workflowmemo"
mandatory="true" helpstring="Name of the Repository."> </property>
<property name="Workflow" displayname="Workflow Name" type="string" mandatory="true"
helpstring="Name of the Workflow."> </property>
<property name="ExecutionID" displayname="Execution ID" type="workflowmemo"
mandatory="true" helpstring="Execution ID of the workflow instance that needs to be
aborted."> </property>
</properties>
<return>
<value helpstring="Workflow Aborted">Workflow Aborted</value>
<value helpstring="Workflow cannot be aborted">Workflow cannot be aborted</value>
<value helpstring="Couldn't Abort Workflow">Couldn't Abort Workflow</value>
</return>
</action>
</category>
</actionsdata>
```

   While configuring the Custom Activity, we need to specify the image filename of PNG type of size 40x40 for the custom activity which should be in the following folders:

   [AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Actions\Images\Normal\Large,
   [AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\WorkflowElements\Default\en-US\Actions\Images\Selected\Large, and

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\ActivityImages

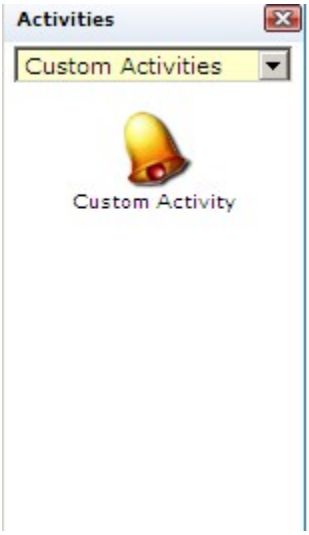
```
<action name="CustomActivity" displayname="Custom Activity">
<description>Custom activity. </description>
<image resourcename="bell.png"></image>
<handler classname="SampleCustomAction.CustomAction" assembly="bin\
SampleCustomAction.dll"></handler>
<properties>
 <property name="Sample Property with Expression" type="SampleProperty"
mandatory="true">
 <choice>Yes;Yes</choice>
 <choice>No;No</choice>
 </property>
</properties>
<return>
 <value helpstring="Success">Success</value>
 <value helpstring="UnSuccess">UnSuccess</value>
 </return>
</action>
```

Note: Suppose while configuring the Custom Activities in the Actions.XML file if you do not want to use any properties within the Custom activity, it is mandatory to specify the empty properties tag, as shown in the sample code below.

```
<action name="CustomActivity" displayname="Custom Activity">
<description>Custom activity. </description>
<image resourcename="bell.png"></image>
<handler classname="SampleCustomAction.CustomAction" assembly="bin\
SampleCustomAction.dll"></handler>
<properties></properties>
<return>
 <value helpstring="Success">Success</value>
 <value helpstring="UnSuccess">UnSuccess</value>
 </return>
</action>
```

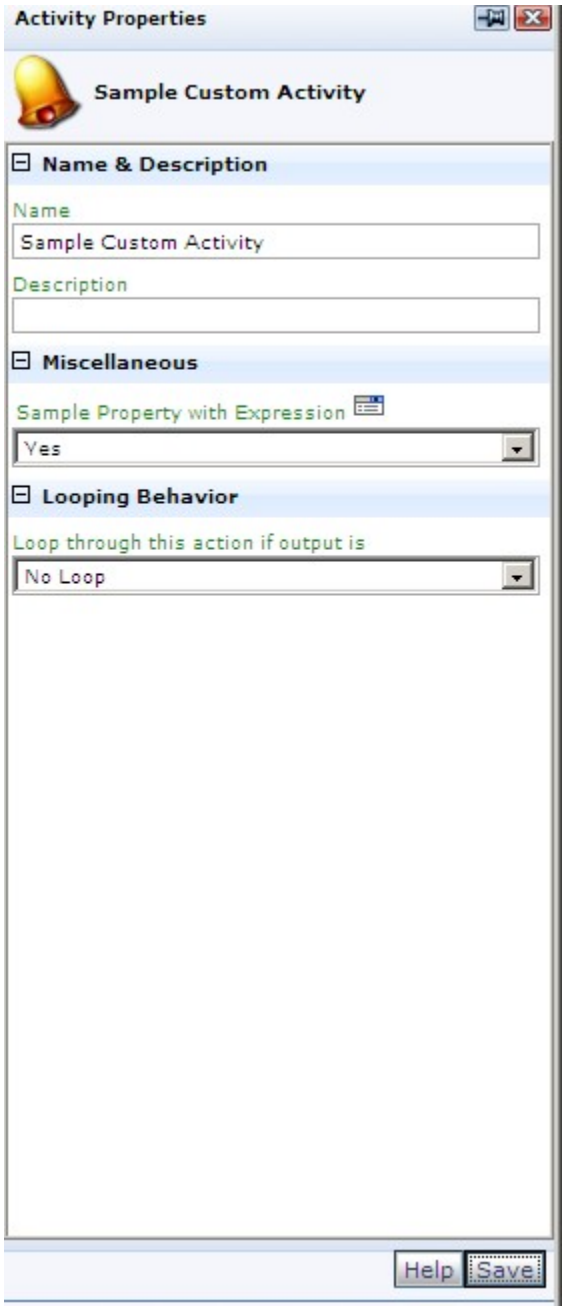
Note: Make sure to set the class name and assembly correctly for the Custom Activity. Classname="Namespace.Classname" and assembly="assembly name that you will get after building the Project".

7. Open the Process Designer and the activity should get displayed under the Custom Activities Category.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

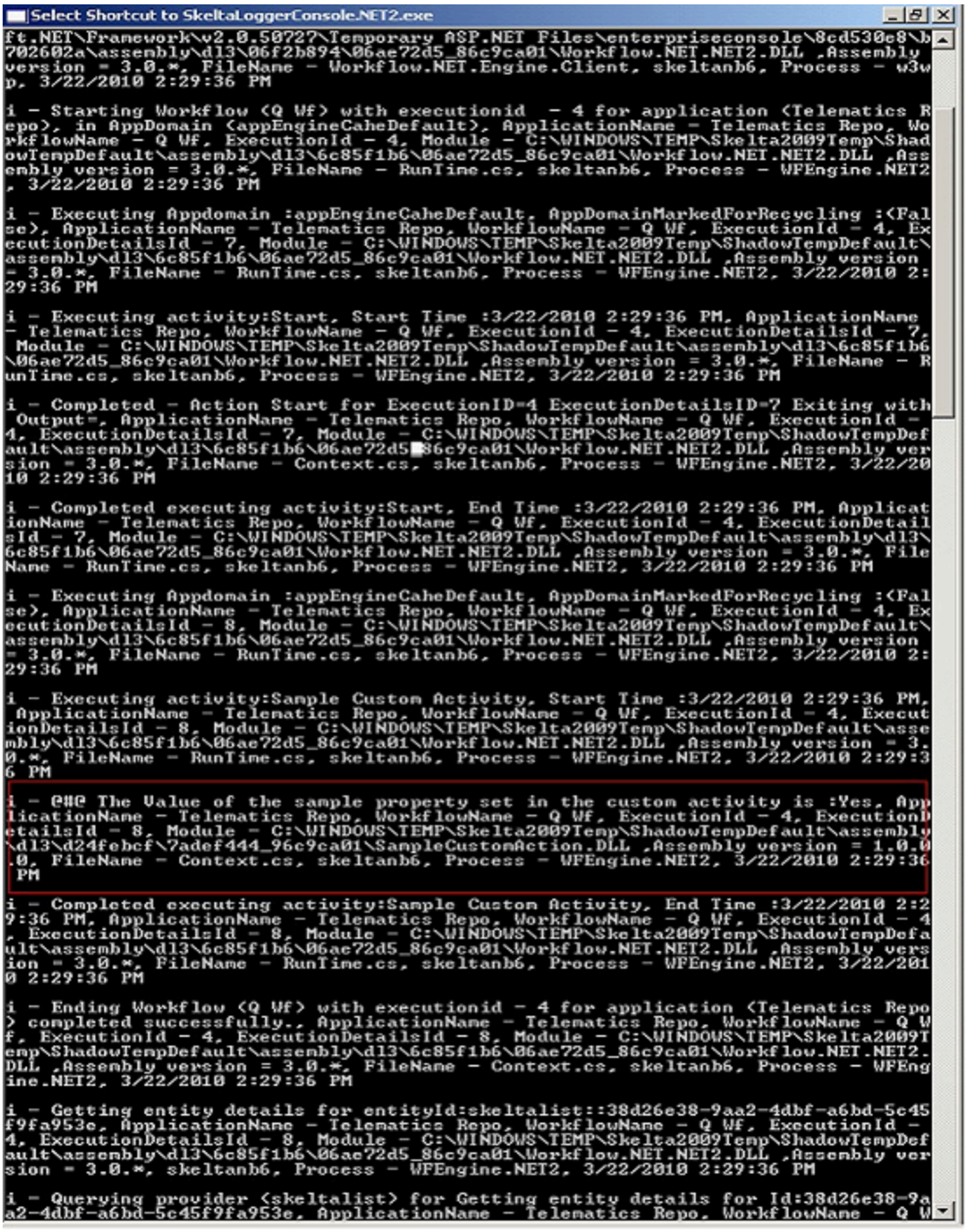
8. Now you can drag and drop the activity to the Process Designer and set the property value and deploy the workflow.



9. Once you Execute a Workflow using the Custom activity you can see the set value of Property in the Logger Console after the custom activity gets executed.

Output

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Configure ASPX Form as Work Item Detail View Page

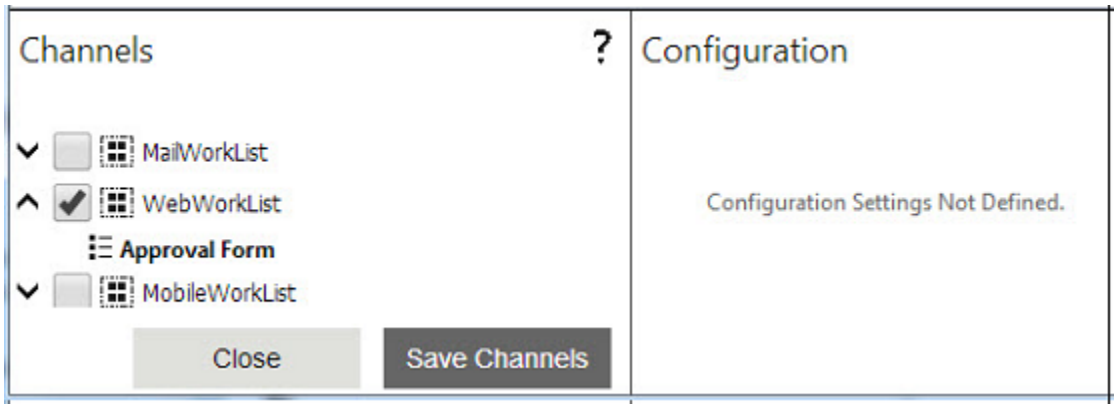
You can configure an ASPX form as the Detailed View page of the Work Item List Control. This topic describes the steps to configure the ASPX form.

To Configure the ASPX Form as a Detailed View Page

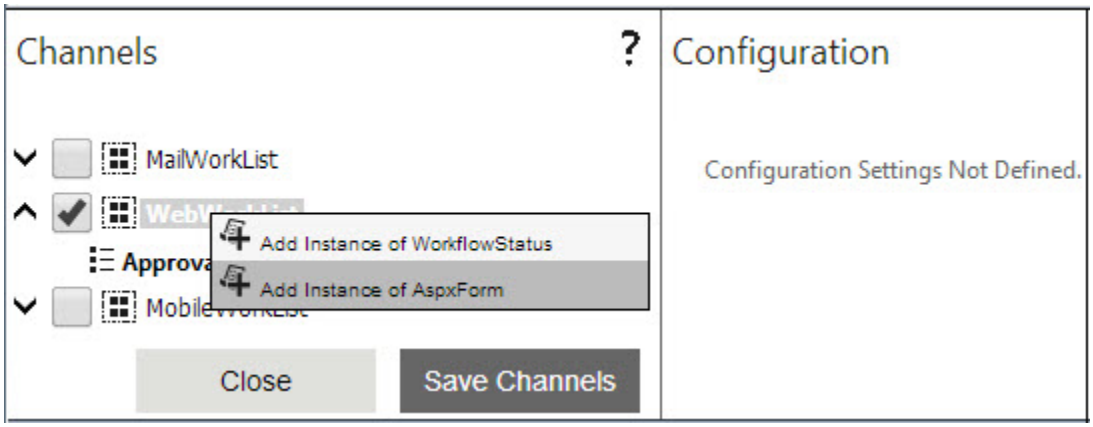
1. Select the **Human Activity**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

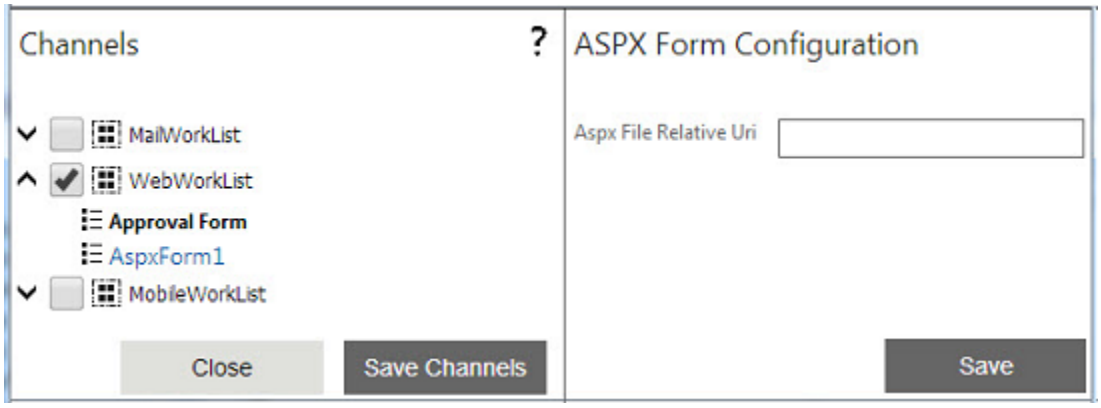
2. Click the **Delivery Channels** Property. A new window opens, as shown below.



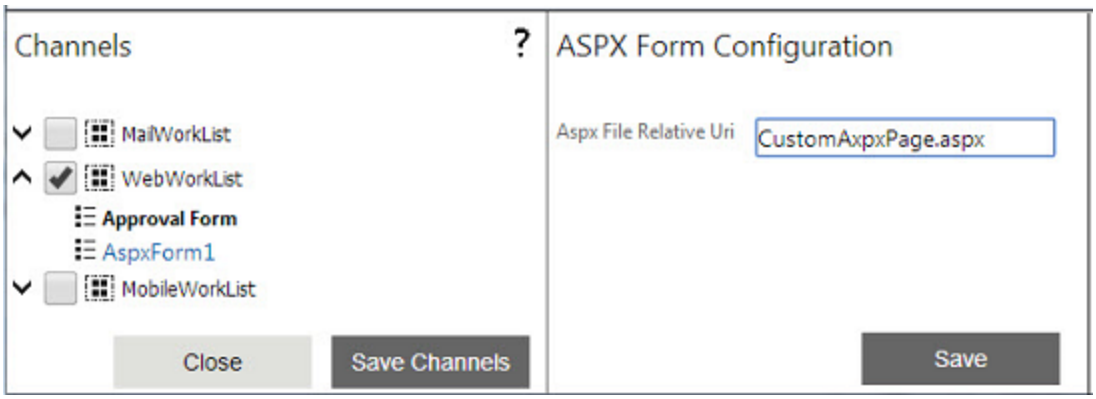
3. Right-click the WebWorkList to display the context menu with the **Add Instance of Aspx form** option, as shown below.



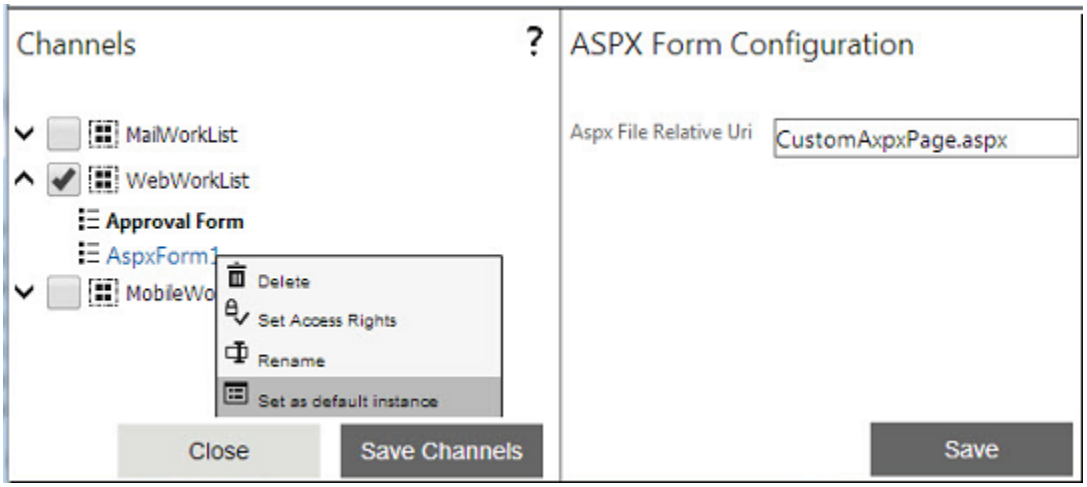
4. Click the Add Instance of Aspx form option. In the configuration settings fields that are displayed, specify the URL of the ASPX form, as shown below. You can specify the Relative or absolute Url in the text box.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



5. If you are accessing the Work Item List Control from the EC Site, then the CustomAspxPageDefault.aspx page should reside in the BPMUITemplates\Default\Repository\Site folder.
6. Set the Aspxform1 as the default instance by right-clicking Aspxform1 and selecting the **Set as default instance** option from the context menu.



7. Click the Save and Save Channels buttons.
8. Following are the steps and code samples for the Custom Aspx form to appear as Work Item detail view.
9. Create the CustomAspxPageDefault.aspx and CustomAspxPageDefault.aspx.cs in the BPMUITemplates\Default\Repository\Site folder.
10. In the CustomAspxPageDefault.aspx.cs file we will get the work item details using the Skelta.HWS.WorkListChannel.Web.WebWorkItemAdapter object. The Url respective to Work Item controls are secured. The Secured Url is getting build when we select a row from the Work Item list control using the WebWorkItemAdapter object. In the Custom Aspx page, we need to access the work item related details from the Secured url. The Processing of Secured url is achieved through the WebWorkItemAdapter object.

CustomAspxPageDefault.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="CustomAspxPageDefault.aspx.cs"
Inherits="CustomASPXForm" %>
<html>
<head runat="server">
 <title>Untitled Page</title>
<link rel="stylesheet" type="text/css" href="<%=_cssPath+"Common/StyleSheet/Global.css"%>"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/>
</head>
<body leftmargin="0" topmargin="0">
<script>
//For calling the Javascript from the Workitem control
function RefreshWorkItemList()
 {
if(typeof(self.opener) != 'undefined')
{
 self.opener.RefreshGridfromdetwindow();
 window.close();
}
else
{
top.RefreshGridfromdetwindow();
}
 }
```

```
</script>
<form id="form1" runat="server">
    <table width="100%" border="0" cellpadding="0" cellspacing="0">
    <tr>
<td>
 <table width="100%"  border="0" cellpadding="4" cellspacing="1" class='tablebg'>

<tr>
 <td width="32%" class="lefttdbg"><font class="subtitle">Application</font></td>
 <td width="68%" class="righttdbg"><font class="description"><%=_application%></font></td>
</tr>
```

```
<tr>
 <td width="32%" class="lefttdbg"><font class="subtitle">Workflow</font></td>
 <td width="68%" class="righttdbg"><font class="description"><%=_workflow%></font></td>
</tr>
<tr>
 <td width="32%" class="lefttdbg"><font class="subtitle">Workflow Instance Id</font></td>
 <td width="68%" class="righttdbg"><font class="description"><%=_instanceid%></font>
 </td>
</tr>

<tr>
 <td width="32%" class="lefttdbg"><font class="subtitle">WorkItem Id</font></td>
 <td width="68%" class="righttdbg"><font class="description"><%=_workitemId%></font></td>
</tr>
<tr>
 <td width="32%" class="lefttdbg"><font class="subtitle">Activity</font></td>
 <td width="68%" class="righttdbg"><font class="description"><%=_activityname%></font></td>
</tr>

<tr>
 <td width="32%" class="lefttdbg"><font class="subtitle">Subject</font></td>
 <td width="68%" class="righttdbg"><font
class="description"><%=_workitemSubject%></font></td>
</tr>
 </table>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 </td>
</tr>
 </table>
<br />
 <table width="100%"  border="0" cellpadding="4" cellspacing="1" class='tablebg'>
 <tr>
<td  width="32%" class="lefttdbg"><font class="subtitle">Activity Output</font></td>
<td  width="68%" class="righttdbg"><asp:DropDownList ID="ddlStatus" runat="server"
CssClass="inputselect" >
</asp:DropDownList>
</td>
</tr>
```

```
<tr>
<td align="right" colspan="2" >
<asp:Button ID="btnSubmit" Text="Submit" runat="server" class="inputbutton"/>
</td>
</tr>
</table>
     </form>
</body>
</html>
```

CustomAspxPageDefault.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class CustomASPXForm : System.Web.UI.Page
{
```

```
Skelta.HWS.WorkItem wItem;
protected Guid _workitemId = Guid.Empty;
protected string _workitemSubject = "";
protected string _activityname = "";
protected string _application = "";
protected string _workflow="";
protected string _instanceid="";
protected string _cssPath = "";
protected void Page_Load(object sender, EventArgs e)
{
```

```
//This is for getting the common folder relative path
_cssPath = Convert.ToString(Workflow.NET.TemplateExpressionBuilder.GetUrl(""));
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// The Url respective to Workitem controls are secured.
// The Secured Url is getting build when we select a row from the Workitem list control
 using the WebWorkItemAdapter object.
// In the Custom Aspx page we need to access the workitem related details from the Secured
url.
// The Processing of Secured url is achieved through the WebWorkItemAdapter object.
```

```
Skelta.HWS.WorkListChannel.Web.WebWorkItemAdapter adapter = new
Skelta.HWS.WorkListChannel.Web.WebWorkItemAdapter();

//ProcessWebChannelRequest Method is used to Decrypt the Securedurl passed to the Detail
Page and
//get the details related to Workitem like
 Applicationname,workflowname,executionid,currentworkitem etc
adapter.ProcessWebChannelRequest();

//adapter.CurrentWorkItem will give you the entire selected Workitem object details.
wItem = adapter.CurrentWorkItem;

//adapter.ApplicationName will give you the application name.
_application = adapter.ApplicationName;

//adapter.WorkflowName will give you the workflow name.
_workflow = adapter.WorkflowName;
```

```
//This is getting the Workitem id from the Workitem object
_workitemId = wItem.Id;

//adapter.CurrentWorkItem.Subject will give you the value of the Subject defined for the
activity
_workitemSubject = adapter.CurrentWorkItem.Subject;

//adapter.CurrentWorkItem.Subject will give you the value of the activity display name
_activityname = adapter.CurrentWorkItem.CurrentActivity.ActionDisplayName;

//adapter.CurrentWorkItem.ExecutionId will give you the Workflow Execution Id
_instanceid = adapter.CurrentWorkItem.ExecutionId.ToString();
```

```
//To Get the Possible outputs defined for the selected activity
System.Collections.Generic.Dictionary<string, string> GetStatusList = new
System.Collections.Generic.Dictionary<string, string>();
GetStatusList =
((Skelta.HWS.IHWSActivityProvider)adapter.CurrentWorkItem.HWSActivity.CurrentActivityHandle
r).GetPossibleStatusList();

foreach (string kvp in GetStatusList.Keys)
{
ddlStatus.Items.Add(new ListItem(kvp,GetStatusList[kvp]));

}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
btnSubmit.Click += new EventHandler(btnSubmit_Click);

}

/// <summary>
/// Event for handling the Submit button
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
protected void btnSubmit_Click(object sender, EventArgs e)
{
string status = ddlStatus.SelectedItem.Value;

//Submitting the Workitem
wItem.Submit("WebWorkList", status , "", "");

//Refreshing the Workitem grid after submitting the workitem
Page.ClientScript.RegisterClientScriptBlock(this.GetType(), "RefreshGrid", "<script
language='javascript'>RefreshWorkItemList();</script>");

}


}
```

Now, when you click any row in the Work items list, you will find this aspx page being displayed as the detailed view.

Get Client Credential Token

Use `GetDefaultClientCredentialToken()` method to return the client credential token using custom scope.

Syntax

```
var token = TokenServiceCommonFunctions.GetDefaultClientCredentialToken(applicationName);
```

Parameters

applicationName: Application Name

Return Value

This method returns token value of string datatype.

Example

```
//Get the default client credential token with custom scope.
var token = TokenServiceCommonFunctions.GetDefaultClientCredentialToken("applicationName");
```

Note: Use this function in the custom activity if the access token is required.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

IPropertyRoutingCondition

This interface has to be implemented by the custom property to return the dynamic routing condition values from the property.

GetRoutingConditionValues method will be available in the custom propety class which implements IPropertyRoutingCondition which returns SortedList which will be seen in the Process Designer when we link the custom action which uses this property to another Activity.

We have to define the Link tag under the Custom Activity Return tag Section.

```
<link property="PropertyNameAttribute" >;
```

Say we want to dynamically route the values based on the custom property named "SampleProperty", then we have to give the link tag under the Custom Activity Return tag Section like below.

```
<link property="SampleProperty" >;
```

Lists

This section provides information on workflow lists and creating a list item.

Workflow List

The Workflow List is a list of workflows and folders which contains workflows within it. The Workflow List provides a central interface that enables categorization of workflows.

Add Workflow Item to Workflow List

The list internally uses Form engine infrastructure. During list creation, the list columns are defined as form fields. When adding an item, values have to be set for these fields. The following code example demonstrates how to add a new work flow item to workflow list.

PROCEDURE

Dll REFERENCE

```
Workflow.NET.NET2
Skelta.forms.core
```

NAMESPACE USED

```
Workflow.NET
Skelta.forms.core
```

Try the following code in a console application by adding the above references and name spaces:

```
//Creating ListDefinition for specified Repository and List
//Here Repository="Repo1" and ListName="Workflow"
ListDefinition _listDef = ListDefinition.GetList(new
Skelta.Core.ApplicationObject(_applicationName), _listName);

//Here specify your repository name and retain the second parameter "Workflow"
//Creating a new listItem for the given list definition.
Skelta.Repository.List.ListItem newListItem = new Skelta.Repository.List.ListItem(listdef);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//Assigning the Title and Description value for the list item.
((ListTextDataItem)newListItem.ListForm.Records[0].FindControlByID("_sys_wf_title")).Value
= "Workflow Name";

((ListTextDataItem)newListItem.ListForm.Records[0].FindControlByID("_sys_wf_desc")).Value =
"Description text";

//Accessing the ListTableForm in the Workflow list definition
ListTableForm listTableForm =
((ListTableForm)newListItem.ListForm.Records[0].FindControlByID("_sys_wfdef_form"));

//Setting the version value
((ListTextDataItem)listTableForm.Records[0].FindControlByID("_sys_wfdef_filename")).Value =
"1";

//For Workflow item, the Itemtype should be 1
newListItem.ItemType = 1;

//Accessing User context object created while logging in to Enterprise Console site
UserContext UserCtx = new UserContext();

newListItem.LoggedInUserId = UserCtx.LoggedInUserId;

// Note:User context will be available only when the user logins in the  enterprise
console.Otherwise // the LoggedInUserID has to be set manually
//Example: newListItem.LoggedInUserId = "activedirectory::John"

//For saving the item under a folder or workflow set the following property
//[ newListItem.ParentItemId = Guid of the Parent workflow folder/item ]

//Saving the list item
newListItem.Save();
```

Note: To create a workflow the itemtype should be 1 and to create a folder the itemtype should be 0.

Get All the Workflows Present in the Selected Repository

PROCEDURE

Dll REFERENCE

```
Workflow.Net.NET2
Skelta.Forms.Core
```

NAMESPACE USED

```
Workflow.NET
Skelta.Repository.List
Skelta.Forms.Core.Controls
```

Get all the Workflow Names

The code below demonstrates how to get all the workflow names present in the specified repository.

```
//Namespaces
using Skelta.Repository.List;
using Skelta.Forms.Core.Controls;
using Workflow.NET;
//Class level Variables
string _applicationName = "MyRepo1";  //Repository Name
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
string _listName = "Workflow"; //This value is fixed for workflow list
string UserId="skeltalist::1AA9AA25-92F8-4635-839B-CB029EB95B2F";
//1AA9AA25-92F8-4635-839B-CB029EB95B2F is userguid of the user admin in AVEVA Work Tasks
list
Guid parentItemId=Guid.Empty;  //This is the Id of Parent workflow
string[] wfarray;  //An array of string consisting of all the workflow names

//Creating a hashtable for storing column names
 //AVEVA Work Tasks stores table information as virtual columns, so we are using Hashtable
here

Hashtable _colNames=new Hashtable();

//Create an object of ListDefinition class (_listDef) by passing //Repository name and List
name(Workflow) or Guid as parameters
ListDefinition _listDef = ListDefinition.GetList (new
Skelta.Core.ApplicationObject(_applicationName),
_listName);

//Creating ListItemCollection object for the workflow
ListItemCollection _listItemCollection=new ListItemCollection(_listDef);

//Create WorkflowListColumns for all the workflow(ListItems) in the Workflow list
 //SortedDictionary is used to map the virtual columns and physical columns
SortedDictionary<string,Skelta.Forms.Core.Controls.BaseDataControl> fields=_
listItemCollection.GetRecordsFieldMapping();
 if(_colNames.Count>0)
 {
 _colNames.Clear();
 }
foreach(KeyValuePair<string, Skelta.Forms.Core.Controls.BaseDataControl> wfInfo in fields)
{
 _colNames.Add(wfInfo.Value.Name.ToString(),wfInfo.Key.ToString());
}

//Calling GetRecordsForConsume API to get all workflow(ListItems)
DataTable WfItems=_ listItemCollection.GetRecordsForConsume(parentItemId, UserId);
int itemCount=0;
wfarray=new string[WfItems.Rows.Count];
foreach(DataRow dr in WfItems.Rows)
{
 wfarray[itemCount]=dr[_colNames["Title"].ToString()].ToString();
 itemCount++;
}
```

Get all the Workflow with description

```
The code below demonstrates how to get all the workflow names with descriptions present in
the specified repository.
//Namespaces
using Skelta.Repository.List;
using Skelta.Forms.Core.Controls;
using Workflow.NET;

//Class level Variables
string _applicationName = "MyRepo1";  //Repository Name
string _listName = "Workflow"; //This value is fixed for workflow list
string UserId="skeltalist::1AA9AA25-92F8-4635-839B-CB029EB95B2F";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//1AA9AA25-92F8-4635-839B-CB029EB95B2F is userguid of the user admin in AVEVA Work Tasks
list
Guid parentItemId=Guid.Empty;  //This is the Id of Parent workflow
string workflowDescription="";  //All workflow with Name and Description

//Creating a hashtable for storing column names
 //AVEVA Work Tasks stores table information as virtual columns, so we are using Hashtable
here
Hashtable _colNames=new Hashtable();

//Create an object of ListDefinition class (_listDef) by passing //Repository name and List
name(Workflow) or Guid as parameters
ListDefinition _listDef=new ListDefinition(new
Skelta.Core.ApplicationObject(_applicationName), _listName);

//Creating ListItemCollection object for the workflow
ListItemCollection _listItemCollection=new ListItemCollection(_listDef);

//Create WorkflowListColumns for all the workflow(ListItems) in the Workflow list
 //SortedDictionary is used to map the virtual columns and physical columns

SortedDictionary<string,Skelta.Forms.Core.Controls.BaseDataControl> fields=_
listItemCollection.GetRecordsFieldMapping();
 if(_colNames.Count>0)
 {
 _colNames.Clear();

 }
foreach(KeyValuePair<string, Skelta.Forms.Core.Controls.BaseDataControl> wfInfo in fields)
{
 _colNames.Add(wfInfo.Value.Name.ToString(),wfInfo.Key.ToString());
}

//Calling GetRecordsForConsume API to get all workflow(ListItems)
DataTable WfItems=_ listItemCollection.GetRecordsForConsume(parentItemId, UserId);
int itemCount=0;
StringBuilder strBuilder=new StringBuilder();
strBuilder.Append("<Workflows>");
foreach(DataRow dr in WfItems.Rows)
{
 strBuilder.Append("<workflow
name=\""+dr[_colNames["Title"].ToString()].ToString()+"\">"+dr[_colNames["Description"].ToS
tring()].ToString()+"</workflow>");
 itemCount++;
}
strBuilder.Append("</Workflows>");
//Returns all the workflow with Name and Description
workflowDescription=strBuilder.ToString();
```

Get All the Versions of a Workflow

PROCEDURE

Dll REFERENCE

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Workflow.Net.NET2
Skelta.Forms.Core
```

NAMESPACE USED

```
Workflow.NET
Skelta.Repository.List
Skelta.Forms.Core.Controls
```

The code below demonstrates how to get all the versions for a specified workflow.

```
//Namespaces
using Skelta.Repository.List;
using Skelta.Forms.Core.Controls;
using Workflow.NET;

//Class level Variables
string _applicationName = "MyRepo1";  //Repository Name

string Workflow="TestWf";  //Workflow name for which all version //information is required
string[] versions;  //Returns all the versions for a specified  workflow

Skelta.Repository.WorkflowList.WorkflowDataHandler wdh=new
Skelta.Repository.WorkflowList.WorkflowDataHandler(_applicationName);

//Pass the workflow name to get all the versions for this workflow
ListItemVersion[] workflowVersions=wdh.GetAllVersions(Workflow);
Versions=new string[workflowVersions.Length];

int itemCount=0;
foreach(ListItemVersion version in workflowVersions)
{
 versions[itemCount]=version.VersionStamp;
itemCount++;
}
```

Get All the Published Versions of a Workflow

PROCEDURE

Dll REFERENCE

```
Workflow.Net.NET2
Skelta.Forms.Core
```

NAMESPACE USED

```
Workflow.NET
Skelta.Repository.List
Skelta.Forms.Core.Controls
```

The code below demonstrates how to get all the published versions for a specified workflow.

```
//Namespaces
using Skelta.Repository.List;
using Skelta.Forms.Core.Controls;
using Workflow.NET;

//Class level Variables
string _applicationName = "MyRepo1";  //Repository Name
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
string _listName = "Workflow"; //This value is fixed for workflow list
string Workflow="TestWf";  //Workflow name for which all published version information is
required

//Create an object of ListDefinition class (_listDef) by passing //Repository name and List
name(Workflow) or Guid as parameters
ListDefinition _listDef=new ListDefinition.GetList(new
Skelta.Core.ApplicationObject(_applicationName), _listName);


//Id of the ListItem to import the workflow
 //This Id is obtained from the table SKEWorkflow, value fo the column Id corresponding to
your Repository Name(Application) and Workflow Name(Title)
Guid WfId = new Guid("65caf315-889e-4c46-b8d4-ea9b0e09013b");

//ListItem object for the specified workflow by passing ListDefinition object(_listDef) and
ListItem id(WFId)
Skelta.Repository.List.ListItem WFlistItem = new Skelta.Repository.List.ListItem(_listDef,
WFId);

//Get all the Published versions information for a specified workflow
ListItemVersion[] versionColl= WFlistItem.GetAllPublishedVersionsInformation();

//Creating a List for saving all the published version information for a specified workflow
List<ListItemVersion> versionList=new List<ListItemVersion>(versionColl);
```

Add Folder Item to Workflow List

The list internally uses Form engine infrastructure. During list creation, the list columns are defined as form fields. When adding an item, values have to be set for these fields.

For example: The following code example demonstrates how to add a new work flow item to Workflow list. For folder item, the Itemtype value should be 0.

PROCEDURE

Dll REFERENCE

```
Workflow.NET.NET2
Skelta.forms.core
```

NAMESPACE USED

```
Workflow.NET
Skelta.Forms.Core.Controls
```

Try the following code in a console application by adding the above references and name spaces:

```
//Creating ListDefinition for specified Repository and List
//Here Application="Repo1" and List="Workflow"
ListDefinition listdef = ListDefinition.GetList(new
Skelta.Core.ApplicationObject("Repository2"), "Workflow");

//Creating a new listItem for the given listdefinition.
Skelta.Repository.List.ListItem newListItem = new Skelta.Repository.List.ListItem(listdef);

//Assigning the Title and description for the listitem.
 ((ListTextDataItem)newListItem.ListForm.Records[0].FindControlByID("_sys_wf_title")).Value
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
= "Folder Name";
 ((ListTextDataItem)newListItem.ListForm.Records[0].FindControlByID("_sys_wf_desc")).Value
= "Description text";

//For "Folder" item, Itemtype should be 0
newListItem.ItemType = 0;

//Accessing User context created while logging in to Enterprise Console site
UserContext UserCtx = new UserContext();
newListItem.LoggedInUserId = UserCtx.LoggedInUserId;

//Note:User context will be available only when the user logins in the enterprise console.
Otherwise //the LoggedInUserID has to be set manually
//Example: newListItem.LoggedInUserId = "activedirectory::John"

//For saving the item under a folder or workflow set the following property
//newListItem.ParentItemId = Guid of the Parent workflow folder/item

//Saving the information in the database.
newListItem.Save();
```

Edit Existing Folder Item in Workflow List

The following code sample demonstrates how to edit the existing folder details available in the Workflow list.

PROCEDURE

Dll REFERENCE

```
Workflow.NET.NET2
Skelta.forms.core
```

NAMESPACE USED

```
Workflow.NET
Skelta.Forms.Core.Controls
Skelta.Repository.List
Skelta.Entity
```

Try the following code in a console application by adding the above references and name spaces:

```
//Creating ListDefinition for specified Repository and List
//Here Application="Telematics Repo" and List="Workflow"
ListDefinition listdef = ListDefinition.GetList(new
Skelta.Core.ApplicationObject("Telematics Repo"), "Workflow");
//Creating the User Context
UserContext UserCtx = new UserContext(new Guid("38D26E38-9AA2-4DBF-A6BD-5C45F9FA953E"),new
Skelta.Core.ApplicationObject("Telematics Repo"),"skeltalist") ;
Skelta.Repository.List.ListItemCollection newListItemCollection = new
Skelta.Repository.List.ListItemCollection(listdef);
DataTable dtAllRecords = new DataTable();
//Test is the folder name
dtAllRecords = newListItemCollection.GetRecords("Title", "Test",UserCtx.LoggedInUserId);
Guid folderGuid = new Guid(dtAllRecords.Rows[0]["Id"].ToString());
Skelta.Repository.List.ListItem newListItem = new Skelta.Repository.List.ListItem(listdef,
folderGuid);
//Assigning the Title and description for the listitem
((ListTextDataItem)newListItem.ListForm.Records[0].FindControlByID("_sys_wf_title")).Value
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
= "Testing";
((ListTextDataItem)newListItem.ListForm.Records[0].FindControlByID("_sys_wf_desc")).Value
="Description text";
//Saving the information in the database.
newListItem.Save();
```

Import the Workflow from Workflow Definition Xml

AVEVA Work Tasks provides a feature to import the workflow from the definition XML in the Process Designer. This can also be done using the AVEVA Work Tasks APIs programmatically. The following code sample demonstrates how to import the workflow.

PROCEDURE

Dll REFERENCE

```
Workflow.Net.NET2
Skelta.Forms.Core
```

NAMESPACE USED

```
Workflow.NET
Skelta.Repository.List
Skelta.Forms.Core.Controls
```

Try the following code in a console application by adding the above references and name spaces:

```
//Namespaces
using Skelta.Repository.List;
using Skelta.Forms.Core.Controls;
using Workflow.NET;

//Class level Variables

//Repository Name
string _applicationName = "MyRepo1";

//This value is fixed for workflow lists
string _listName = "Workflow";
ListDefinition _listDef = ListDefinition.GetList(new
Skelta.Core.ApplicationObject(_applicationName), _listName);

//specify path of exported workflow Xml file.
string strData = System.IO.File.ReadAllText (@"M:\SkeltaDev3.0\xml.xml");

//Id of ListItem to import the workflow
//This id is obtained from the table SKEWorkflow, value of the column Id corresponding to
your
//repository name(Application) and workflow (Title).
Guid gid = new Guid("65caf315-889e-4c46-b8d4-ea9b0e09013b");
Skelta.Repository.List.ListItem ListItemForImport = new
Skelta.Repository.List.ListItem(_listDef, gid);
ListTableForm listTableForm1 = ((ListTableForm)
ListItemForImport.ListForm.Records[0].FindControlByID("_sys_wfdef_form"));
string fileName =
(string)((ListTextDataItem)listTableForm1.Records[0].FindControlByID("_sys_wfdef_filename")
).Value;
string workflowname = ListItemForImport.Title;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Workflow.NET.Interfaces.IDataHandler DBHandler = null;
Config ConfigObj = new Config(_applicationName);
Workflow.NET.Storage.Staging stg = new Workflow.NET.Storage.Staging(DBHandler);
stg.Update("MyRepo1", workflowname, fileName, strData);
ListItemForImport.LoggedInUserId = "skeltalist::1AA9AA25-92F8-4635-839B-CB029EB95B2F";

// If the provider is Active Directory the format is  "activedirectory::user id"
//1AA9AA25-92F8-4635-839B-CB029EB95B2F is userguid of the user admin in skelta list
//setting logged-In userid is mandatory
ListItemForImport.ChangeStatusAndUpdate(ListItemVersionStatus.Published);
```

Create New Version

The following code sample demonstrates the creation of a new version for a workflow.

PROCEDURE

Dll REFERENCE

```
Workflow.Net.NET2
Skelta.Forms.Core
```

NAMESPACE USED

```
Workflow.NET
Skelta.Repository.List
Skelta.Forms.Core.Controls
```

Try the following code in a console application by adding the above references and name spaces:

```
//Namespaces
using Skelta.Repository.List;
using Skelta.Forms.Core.Controls;
using Workflow.NET;

//Class level Variables
//Repository Name
string _applicationName = "MyRepo1";
//This value is fixed for workflow lists
string _listName = "Workflow";
ListDefinition _listDef = ListDefinition.GetList(new
Skelta.Core.ApplicationObject(_applicationName), _listName);
//specify path of exported Workflow xml file.
string strData = System.IO.File.ReadAllText(@"c:\ADSQLRep_TestWF_1.xml");

//Id of ListItem to which import the workflow
//This id is obtained from the table SKEWorkflow, value of the column Id corresponding to
your //repository name(Application) and workflow (Title).
Guid gid = new Guid("6ffa4b1f-9151-4158-b8b3-6733cf2f63ce");
Skelta.Repository.List.ListItem ListItemChangeVersion = new
Skelta.Repository.List.ListItem(_listDef, gid);
ListTableForm listTableForm1 =
((ListTableForm)ListItemChangeVersion.ListForm.Records[0].FindControlByID("_sys_wfdef_form"
));
string fileName =
(string)((ListTextDataItem)listTableForm1.Records[0].FindControlByID("_sys_wfdef_filename")
).Value;
string workflowname = ListItemChangeVersion.Title;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Workflow.NET.Interfaces.IDataHandler DBHandler = null;
Config ConfigObj = new Config(_applicationName);
Workflow.NET.Storage.Staging stg = new Workflow.NET.Storage.Staging(DBHandler);

//Repository name
stg.Update("MyRepo1", workflowname, fileName, strData);
ListItemChangeVersion.LoggedInUserId = " skeltalist::1AA9AA25-92F8-4635-839B-CB029EB95B2F";

// If the provider is Active Directory the format is "activedirectory::user id"
//1AA9AA25-92F8-4635-839B-CB029EB95B2F is userguid of the user admin in skelta list
//setting logged-In userid is mandatory
ListItemChangeVersion.SaveAsNextMajorVersion(); // Can Create new version
 ListItemChangeVersion.ChangeStatusAndUpdate(ListItemVersionStatus.Published);
```

Create ListItem

The following code sample demonstrates the creation of a list item in the Workflows list.

PROCEDURE

Dll REFERENCE

```
Skelta.Forms.Core.dll
Workflow.Net.NET2.dll
```

NAMESPACE USED

```
Skelta.Repository.List
Skelta.Forms.Core.Controls
Workflow.NET
```

Create a console Application and execute the following code:

To create a listitem, follow the steps below:

1. Create an object (_listDef) of **ListDefinition** Class by passing application name and listname or Guid.
2. Create an object (_listItem) of **ListItem** class by passing ListDefinition object (_listDef).
3. Find the control named "_sys_wf_form" in the ListItem object (_listItem) and set to **ListMainForm** object (_listMainForm). Do the proper casting.
4. Set the "_sys_wf_title", _sys_wf_desc" controls value properties of **ListMainForm** object (_listMainForm). Do the proper castings.
5. Find the control named "_sys_wfdef_form" in the **ListItem** object(_listItem) and set to ListTableForm object (_listTableForm) . do the proper casting.
6. Set the "_sys_wfdef_filename","_sys_wfdef_productionid" controls value properties of **ListTableForm** object (_listTableForm). Do the proper castings.
7. Set the "LoggedInUserId" property of ListItem object (_listItem) to logged-In userid.
8. Finally call the save method of **ListItem** object (_listItem).

```
//Namespaces
using Skelta.Repository.List;
using Skelta.Forms.Core.Controls;
using Workflow.NET;
//Class level Variables
// Repository Name
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
string _applicationName = "MyRepo1";
// This value is fixed for workflow lists
string _listName = "Workflow";
ListDefinition _listDef;
_listDef = ListDefinition.GetList(new
Skelta.Core.ApplicationObject(_applicationName), _listName);
Skelta.Repository.List.ListItem _listItem = new
Skelta.Repository.List.ListItem(_listDef);

ListMainForm _listMainForm =
((ListMainForm)_listItem.ListForm.Records[0].FindControlByID("_sys_wf_form"));
((ListTextDataItem)_listMainForm.FindControlByID("_sys_wf_title")).Value = "TestWF";
((ListTextDataItem)_listMainForm.FindControlByID("_sys_wf_desc")).Value = "Test
Workflow";

ListTableForm _listTableForm =
((ListTableForm)_listItem.ListForm.Records[0].FindControlByID("_sys_wfdef_form"));
((ListTextDataItem)_listTableForm.Records[0].FindControlByID("_sys_wfdef_filename")).
Value = "1";
((ListIntDataItem)_listTableForm.Records[0].FindControlByID("_sys_wfdef_productionid"
)).Value = "0";
_listItem.LoggedInUserId = "skeltalist::1AA9AA25-92F8-4635-839B-CB029EB95B2F";

//1AA9AA25-92F8-4635-839B-CB029EB95B2F is userguid of the user admin in skelta list
 //setting logged-In userid is mandatory
_listItem.Save();
```

Save ListItem

Save a ListItem using list API.

The list API makes use of the following DLL references, namespaces, classes, and methods.

DLL Reference	Namespace
• Workflow.NET.NET2 • Skelta.Forms.Core	• System • Skelta.Repository.List

Class
• Skelta.Repository.List.ListDefinition • Skelta.Repository.List.ListItem

Methods
• `ListItem.Save()`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Methods
This method is used to save the ListItem.

Scenario 1

For the **Employee** list mapped to an **Emp** table with the fields such as **Title**, **Name**, **City**, **State**, **Pincode**, and **Age**, save the ListItem by running the following code.

```
using System;
using Workflow.NET;
using Workflow.NET.Engine;
using Skelta.Forms.Core.Controls;
using Skelta.Repository.List;
public class WorkflowScript0144412702584f2a8852a95a0a727737
{
public string Run(int ExecutionId, int ExecutionDetailsId, Workflow.NET.Engine.Context ctx,
Workflow.NET.Action action, string inlink)
{
string applicationname = ctx.ApplicationName;
string listname = "Employee";
ListDefinition lstdef = new ListDefinition(new
Skelta.Core.ApplicationObject(applicationname), listname);
Skelta.Repository.List.ListItem listItem = new Skelta.Repository.List.ListItem(lstdef);
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("Title"))).Value = "T3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("Name"))).Value = "A3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("City"))).Value = "C3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("State"))).Value = "S3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("PinCode"))).Value =
"P3";
((ListIntDataItem)(listItem.ListForm.Records[0].FindControlByName("Age"))).Value = "20";
listItem.Save();
return "saved";
}
}
```

Scenario 2

If we add a new column to the **Emp** Table say **Department**, then we need to call the `ClearTableNameKeyFromTableCollection` API before calling the `listitem.save()`.

```
Skelta.Core.DS.SingleTableObject.ClearTableNameKeyFromTableCollection(ctx.DataHandler.DataSourceTyp
applicationname, "EmpTable");
```

```
using System;
using Workflow.NET;
using Workflow.NET.Engine;
using Skelta.Forms.Core.Controls;
using Skelta.Repository.List;
public class WorkflowScript0144412702584f2a8852a95a0a727737
{
public string Run(int ExecutionId, int ExecutionDetailsId, Workflow.NET.Engine.Context ctx,
Workflow.NET.Action action, string inlink)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
string applicationname = ctx.ApplicationName;
string listname = "Employee";
Skelta.Core.DS.SingleTableObject.ClearTableNameKeyFromTableCollection(ctx.DataHandler.DataS
ourceType, applicationname, "Emp");
ListDefinition lstdef = new ListDefinition(new
Skelta.Core.ApplicationObject(applicationname), listname);
Skelta.Repository.List.ListItem listItem = new Skelta.Repository.List.ListItem(lstdef);
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("Title"))).Value = "T3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("Name"))).Value = "A3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("City"))).Value = "C3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("State"))).Value = "S3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("PinCode"))).Value =
"P3";
((ListIntDataItem)(listItem.ListForm.Records[0].FindControlByName("Age"))).Value = "20";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("Department "))).Value =
"R&D";
listItem.Save();
return "saved";
}
}
```

Data Source

You can create a data source in the following ways:

- Implementing IDBConnectionString Interface
- Using ConnectionString
- Using Assembly File
- Subscribing AssemblyResolve Event while Using APIs for Database-related Activities

Implement IDBConnectionString Interface

The IDBConnectionString Interface can be used to secure the connection string configuration specified in theFarm Configuration Wizard's Database Settings screen. Custom data can be passed to the interface method by setting the inner text for the data source node.

Using this, the connection string can be moved from the configuration file to the DLL in order to enhance security. When creating a data source in Central Configuration, select the option "Connection Provider" and specify the dll generated in the Assembly Path field.

PROCEDURE

Dll REFERENCE

```
System
System.Data
Workflow.NET.NET2
```

NAMESPACE USED

```
System.Collections.Generic
System.Text
Workflow.NET
Workflow.NET.Storage
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Workflow.NET.Interfaces
Workflow.NET.Engine
Workflow.NET.Engine.Interfaces
```

A sample implementation code is shown below.

```
public class GetConnectionString:IDBConnectionString
{
public GetConnectionString()
{

}
public GetConnectionString(string ParamAttributeFromConfigFile)
{

}

#region IDBConnectionString Members
  /// <summary>
/// This method gets called from skelta when a connection string is expected for DB
operation.
/// </summary>
/// <param name="AppName">Application Name.</param>
/// <param name="InnerTextValueFromDataSourceNode">This parameter is maintained for
backward compatibility and does not provide any value </param>
/// <returns></returns>

public string GetConString(string AppName,object InnerTextValueFromDataSourceNode)
{
return"server=localhost;database=skelta;uid=sa;pwd=;Pooling=False;";
}
#endregion
}
```

Create Data Source Using ConnectionString

The following example demonstrates the creation of data source using connection string. The data source created will be listed in the enterprise console and the repository can be created using the newly created data source.

PROCEDURE

Dll REFERENCE

```
Workflow.NET.NET2
Skelta.Configurations.DB
```

NAMESPACE USED

```
Workflow.NET
Skelta.Configurations.DB.Interfaces
Skelta.Configurations.DB
Skelta.FarmManager
```

Try the following code in a console application by adding the above references and name spaces:

```
using System;
using Skelta.Configurations.DB.Interfaces;
using Skelta.Configurations.DB;
using Workflow.NET;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using Skelta.FarmManager;
namespace CreateDataSource
{
/// <summary>
/// Sample Class to create Data Source using Connection String
/// </summary>
class CreateDataSourceUsingConnectionString
{
static void Main(string[] args)
{
Test();
}

static void Test()
{
try
{
//Set Connection String
string connectionString = "Data Source=localhost;Integrated Security=True;";

//Provide a new name for the Database
string dataBase = "SampleDataBase";

//Creating a Data Source Object with name dataSource
DataSource dataSource = new DataSource();

//Set Name for the Data Source
dataSource.Name = "SampleDataSource";

//Check whether the Data Source Name Exists in Farm Database
DataSourceCollection datasourceCollec = new DataSourceCollection();
if (datasourceCollec.Contains(dataSource.Name))
{
datasourceCollec = null;
throw new Exception("Data Source \"" + dataSource.Name + "\" already exists in Farm
Database");
}

//Set Description for Data Source
dataSource.Description = "SampleDataSource";

//Set SourceType for Data Source(SQL=SQL Server)
dataSource.SourceType = "SQL Server";

//Set ConnectionStringXML for the Data Source
dataSource.ConnectionStringXML = "<datasource >" + connectionString + "Database=" +
dataBase + "</datasource >";

//Set DataBase handler
IDBCore coreHandler = DBCoreHandler.GetHandler(connectionString, "SQL Server");

//Create a new DataBase with Product Tables
coreHandler.CreateDatabaseWithReadCommittedSnapshot(dataBase);
Console.WriteLine("Database \"" + dataBase + "\" was successfully created");
//Run the Product Script to create Product Tables in DataBase
coreHandler.RunScript(Config.GetConfigFilePath() +
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"DatabaseScripts\\SQLServer\\Workflow.NET.sql");


//Create a Database Archive with Product Tables
coreHandler = DBCoreHandler.GetHandler(connectionString, "SQL Server");
coreHandler.CreateDatabaseWithReadCommittedSnapshot(dataBase + "_BI");
Console.WriteLine("Database Archive \"" + (dataBase.ToString() + "_BI") + "\" was
successfully created");
//Run the Product Script to create Product Tables in DataBase Archive
coreHandler.RunScript(Config.GetConfigFilePath() +
"DatabaseScripts\\SQLServer\\Workflow.NET.sql");

//Save the Data Source
dataSource.Save();
Console.WriteLine("DataSource \"" + dataSource.Name + "\" was successfully created in the
Farm Database");
}
catch (Exception e)
{
Console.WriteLine(e.Message);
}
}
}
}
```

Create Data Source Using Assembly File

The following example demonstrates the creation of data source by fetching the connection string from the assembly file. Here the connection string is fetched from the connection class by providing the proper path in the class CreateDataSourceUsingAssembly.

PROCEDURE

Dll REFERENCE

```
Workflow.NET.NET2.dll
```

NAMESPACE USED

```
Workflow.NET
```

Create a ClassLibrary Project by name ConnectionClass and build it. Use the ConnectionClass.dll in the next Sample code.

```
using System;
using Workflow.NET.Interfaces;
namespace ClassLibrary1
{
/// <summary>
/// Sample Class to retrieve Connection String
/// </summary>
public class ConnectionClass : IDBConnectionString
{
public string GetConnectionString(string ApplicationName, object Parameter)
{
// Database Name is not required.New database will be created in the below code sample
return "Data Source=localhost;Integrated Security=True;Pooling=False;";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
}
}
```

Give the path of the ConnectionClass.dll, created by the above sample code for restarting AVEVA Work Tasks services as a parameter to the Method LoadAssembly () in the below code.

IDBConnectionString

PROCEDURE

Dll REFERENCE

```
Workflow.NET.NET2.dll
Skelta.Configurations.DB.dll
```

NAMESPACE USED

```
Skelta.Configurations.DB.Interfaces
Skelta.Configurations.DB
Workflow.NET
Skelta.FarmManager
```

Add the reference of the ConnectionClass.dll that is created above.

Try the following code in a console application by adding the above references and name spaces:

```
using System;
using Skelta.Configurations.DB.Interfaces;
using Skelta.Configurations.DB;
using Workflow.NET;
using Skelta.FarmManager;
namespace CreateDataSource
{
/// <summary>
/// Sample Class to create Data Source using Assembly file.
/// Add Reference ConnectionClass.dll to this code before running it.
/// </summary>
class CreateDataSourceUsingAssembly
{
static void Main(string[] args)
{
Test();
}

static void Test()
{
try
{
string _ClassName = "";
string connectionString = "";

//Provide the new  Database name
string dataBase = "SampleDataBase";

//Creating a Data Source Object with name dataSource
DataSource dataSource = new DataSource();

//Provide the new Name for the Data Source
dataSource.Name = "SampleDataSource";

//Check whether the Data Source Name Exists in Farm Database
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
DataSourceCollection datasourceCollec = new DataSourceCollection();
if (datasourceCollec.Contains(dataSource.Name))
{
datasourceCollec = null;
throw new Exception("Data Source \"" + dataSource.Name + "\" already exists in Farm
Database");
}

//Set Description for Data Source
dataSource.Description = "SampleDataSource";

//Set SourceType for Data Source(SQL=SQL Server)
dataSource.SourceType = "SQL Server";

//Get Connection String from an Assembly

//*****Give the proper path of the ConnectionClass.dll for LoadAssembly Method********
\\ Provide the path of the Connection class dll that you created in the third code sample
Implementation of IDBConnectionString
System.Reflection.Assembly getAssembly =
Workflow.NET.CommonFunctions.LoadAssembly("C:\\ConnectionClass\\bin\\Debug\\ConnectionClass
.dll");
foreach (Type counterType in getAssembly.GetTypes())
{
if (counterType.IsClass)
{
if (Workflow.NET.CommonFunctions.IsInterfaceImplementedByType(counterType,
typeof(Workflow.NET.Interfaces.IDBConnectionString)))
{
_ClassName = counterType.ToString();
break;
}
}
}
connectionString = Workflow.NET.CommonFunctions.GetConnectionString("Central Config",
_ClassName, Environment.CurrentDirectory.ToString()+"\\ConnectionClass.dll", "", "", null);

//Set ConnectionStringXML for the Data Source
dataSource.ConnectionStringXML = "<datasource >" + connectionString + "Database=" +
dataBase + "</datasource >";

//Set DataBase handler
IDBCore coreHandler = DBCoreHandler.GetHandler(connectionString, "SQL Server");

//Create a new DataBase with Product Tables
coreHandler.CreateDatabaseWithReadCommittedSnapshot(dataBase);
Console.WriteLine("Database \"" + dataBase + "\" was successfully created");
coreHandler.RunScript(Config.GetConfigFilePath() +
"DatabaseScripts\\SQLServer\\Workflow.NET.sql");
//Run the Product Script to create Product Tables in DataBase
coreHandler.RunScript(Config.GetConfigFilePath() +
"DatabaseScripts\\SQLServer\\Workflow.NET.sql");


//Create a Database Archive with Product Tables
coreHandler = DBCoreHandler.GetHandler(connectionString, "SQL Server");
```

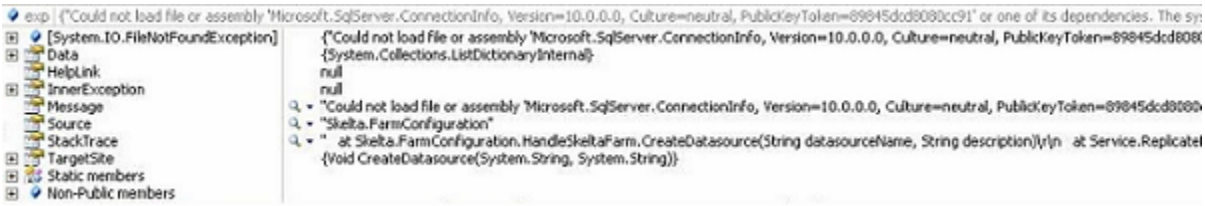
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
coreHandler.CreateDatabaseWithReadCommittedSnapshot(dataBase + "_BI");
Console.WriteLine("Database Archive \"" + (dataBase.ToString() + "_BI") + "\" was
successfully created");
//Run the Product Script to create Product Tables in DataBase Archive
coreHandler.RunScript(Config.GetConfigFilePath() +
"DatabaseScripts\\SQLServer\\Workflow.NET.sql");

//Save the Data Source
dataSource.Save();
Console.WriteLine("DataSource \"" + dataSource.Name + "\" was successfully created in the
Farm Database");
}
catch (Exception e)
{
Console.WriteLine(e.Message);
}
}
}
}
```

Subscribe AssemblyResolve Event while Using APIs for Database-related Activities

If any of the applications use AVEVA Work Tasks APIs for doing some database related activities, then the following code has to be used before calling the API.

Note: If the following code is not added then the following error message is displayed.



Code

```
AppDomain.CurrentDomain.AssemblyResolve += delegate(object sender, ResolveEventArgs e)
{
AssemblyName requestedName = new AssemblyName(e.Name);
if (requestedName.Name.Contains("Microsoft.SqlServer.ConnectionInfo"))
{
if (requestedName.Version.ToString() == "9.0.242.0")
{
throw new Exception("Can not load 9.0.242.0 dll too");
}
else
{
AssemblyName loadThis = new AssemblyName("Microsoft.SqlServer.ConnectionInfo");
loadThis.Version = new Version("9.0.242.0");
loadThis.CultureInfo = requestedName.CultureInfo;
loadThis.SetPublicKeyToken(requestedName.GetPublicKeyToken());
return Assembly.Load(loadThis);
}
}
else
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (requestedName.Name.Contains("Microsoft.SqlServer.Smo"))
{
if (requestedName.Version.ToString() == "9.0.242.0")
{
throw new Exception("Can not load 9.0.242.0 dll too");
}
else
{
AssemblyName loadThis = new AssemblyName("Microsoft.SqlServer.Smo");
loadThis.Version = new Version("9.0.242.0");
loadThis.CultureInfo = requestedName.CultureInfo;
loadThis.SetPublicKeyToken(requestedName.GetPublicKeyToken());
return Assembly.Load(loadThis);
}
}
else
return null;
}
};
```

Example:

If the below API is used to create a data source, then the above code sample has to be used before calling the **CreateDatasource() method.**

Using Assembly resolve event

```
AppDomain.CurrentDomain.AssemblyResolve += delegate(object sender, ResolveEventArgs e)
{
AssemblyName requestedName = new AssemblyName(e.Name);
if (requestedName.Name.Contains("Microsoft.SqlServer.ConnectionInfo"))
{
if (requestedName.Version.ToString() == "9.0.242.0")
{
throw new Exception("Can not load 9.0.242.0 dll too");
}
else
{
AssemblyName loadThis = new AssemblyName("Microsoft.SqlServer.ConnectionInfo");
loadThis.Version = new Version("9.0.242.0");
loadThis.CultureInfo = requestedName.CultureInfo;
loadThis.SetPublicKeyToken(requestedName.GetPublicKeyToken());
return Assembly.Load(loadThis);
}
}
else
{
if (requestedName.Name.Contains("Microsoft.SqlServer.Smo"))
{
if (requestedName.Version.ToString() == "9.0.242.0")
{
throw new Exception("Can not load 9.0.242.0 dll too");
}
else
{
AssemblyName loadThis = new AssemblyName("Microsoft.SqlServer.Smo");
loadThis.Version = new Version("9.0.242.0");
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
loadThis.CultureInfo = requestedName.CultureInfo;
loadThis.SetPublicKeyToken(requestedName.GetPublicKeyToken());
return Assembly.Load(loadThis);
}
}
else
return null;
}
};
HandleSkeltaFarm handlerSkeltaFarm = new HandleSkeltaFarm();
Skelta.FarmConfiguration.HandleSkeltaFarm.ConnectionStringDetails connectStrDetails = new
HandleSkeltaFarm.ConnectionStringDetails();
connectStrDetails.DataSource = "serverName";// Server name
connectStrDetails.DatabaseType = "SQL Server";
connectStrDetails.IntegratedSecurity = true;// true if Windows authentication is used
connectStrDetails.DatabaseName = "DataBaseName";// Database name
handlerSkeltaFarm.PrepareConnectionString(connectStrDetails);
handlerSkeltaFarm.CreateDatasource("DataSourcename", "Datasource description");
```

Without using Assembly resolve event

```
HandleSkeltaFarm handlerSkeltaFarm = new HandleSkeltaFarm();
Skelta.FarmConfiguration.HandleSkeltaFarm.ConnectionStringDetails connectStrDetails = new
HandleSkeltaFarm.ConnectionStringDetails();
connectStrDetails.DataSource = "serverName";// Server name
connectStrDetails.DatabaseType = "SQL Server";
connectStrDetails.IntegratedSecurity = true;// true if Windows authentication is used
connectStrDetails.DatabaseName = "DataBaseName";// Databse name
handlerSkeltaFarm.PrepareConnectionString(connectStrDetails);
handlerSkeltaFarm.CreateDatasource("DataSourcename", "Datasource description");
```

XML Variable API

ASSEMBLY REFERENCES (dlls)

```
Skelta.Forms.Core
Skelta.Enterprise
System
System.Data
System.Web.Services
System.Xml
Workflow.NET.NET2
Workflow.NET.Web.Designer.NET2
```

NAMESPACE USED

```
Workflow.NET
Workflow.NET.Engine
Workflow.NET.Interfaces
Workflow.NET.Web.Designer
Workflow.NET.Engine.Interfaces
Skelta.Repository.List
Skelta.Repository
Skelta.Forms.Core.Controls
Skelta.Forms
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Skelta.Enterprise.XmlStorageType
```

Mapping the value of XML Variable

If the defined XML variable's internal storage is of type Variable, following code has to be included while invoking the workflow.

```
Workflow.NET.VariablesCollection varCol = new Workflow.NET.VariablesCollection();

string xstring =
"<Employees><Employee><Name>John</Name><Department>Development</Department></Employee></Emp
loyees>";

// EmployeeXmlVariable is the name of the XmlVariable defined in the workflow
varCol.Add("EmployeeXmlVariable", "string", xstring);

//Initializes a new instance of the Client class with the specified application and
workflow names
Client objSKClient = new Client("MyRepository","MyWorkflow");

//xmlData can be any Content data that is being passed to the workflow
string xmlData = "<Employee><Name>John</Name></Employee>";


//Executes a workflow with a version number, user id, custom XML content and variable
collection object; xmlData is the content data
objSKClient.Execute(1,"activedirectory::Bob",xmlData,varCol);

// If the defined Xml variable's internal storage is of type: Content , XmlVariable will
take the data from the passed content provided the schema of the Content variable be the
same as passed from here.
```

Updating XML variables

The XML variables defined for a workflow can be updated by getting the current context of the work item and referring to the XML variables, _WorkFlowContext.XmlVariables.Values will give the XML variables defined in the workflow.

```
Workflow.NET.VariablesCollection varCol = new Workflow.NET.VariablesCollection();
string xstring =
"<Employees><Employee><Name>John</Name><Department>Development</Department></Employee></Emp
loyees>";

// EmployeeXmlVariable is the name of the XmlVariable defined in the workflow
varCol.Add("EmployeeXmlVariable", "string", xstring);

//Initializes a new instance of the Client class with the specified application and
workflow names
Client objSKClient = new Client("MyRepository","MyWorkflow");

//xmlData can be any Content data that is being passed to the workflow
string xmlData = "<Employee><Name>John</Name></Employee>";


//Executes a workflow with a version number, user id, custom XML content and variable
collection object; xmlData is the content data
objSKClient.Execute(1,"activedirectory::Bob",xmlData,varCol);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// If the defined Xml variable's internal storage is of type: Content , XmlVariable will
take the data from the passed content provided the schema of the Content variable be the
same as passed from here.
```

Updating the value of the node

The value of a node in the XML can be updated. Here, we will update the "Name" node value of the Employee XML using the following code.

```
//_WorkFlowContext is the runtime context object
Workflow.NET.Engine.Context _WorkFlowContext;
int executionId = 123;

//Gets the workflow Context for the specified execution id where executionId is the
Execution Id (type int) of the workflow.
_WorkFlowContext = Workflow.NET.Engine.Context.GetContext(executionId, "MyRepository");


_WorkFlowContext.XmlVariables["EmployeeXmlVariable"].RawXml =
"<Employees><Employee><Name>Kevin</Name><Department>Testing</Department></Employee></Employ
ees>";

//Updating the value of the node
object[] Submittedvalues = new object[1];
//object[1] refers to the node Name
 Submittedvalues[0] = "Ken";
 _WorkFlowContext.XmlVariables["EmployeeXmlVariable"].SetNodeValue("//Employee/Name",
Submittedvalues);
//EmployeeXmlVariable is the name of the XML variable defined in the workflow
//_WorkFlowContext is the runtime context object
```

XML String(value of the XML variable):
<Employees><Employee><Name>Kevin</Name><Department>Testing</Department></Employee</Employees>

Output(Name of the Employee in the variable has been updated):

<Employees><Employee><Name>Ken</Name><Department>Testing</Department></Employee></Employees>

Updating the Root node

To update the root node, the **XML nodelist** has to be build and the following code has to be executed.

```
 XML String (value of the XML variable):
<Employees><Employee><Name>Kevin</Name><Department>Testing</Department></Employee</Employee
s>
Output (Root node of the XML variable has been changed as shown below):
<Employees><Employee><Name>John</Name><Department>Development</Department></Employee></Empl
oyees>
Getting the value of a node of a XML Variable
To get the value of a node, the following code has to be executed.
//_WorkFlowContext is the runtime context object
Workflow.NET.Engine.Context _WorkFlowContext;
int executionId = 123;

//Gets the workflow Context for the specified execution id where executionId is the
Execution Id (type int) of the workflow.
_WorkFlowContext = Workflow.NET.Engine.Context.GetContext(executionId, "MyRepository");

//XML string, using which the root node of the XMLvariable has to be updated
string xstring =
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"<Employees><Employee><Name>John</Name><Department>Development</Department></Employee></Emp
loyees>";
//Creating the XML nodelist
XmlDocument xdoc = new XmlDocument();
xdoc.LoadXml(xstring);
XmlNodeList xnodelist = xdoc.SelectNodes("//Employees/Employee");
//Defining the object array
object[] submittedvalues = new object[xnodelist.Count];
int i = 0;
//Fill the object array
foreach (XmlNode xnode in xnodelist)
{
 submittedvalues[i] = xnode;
i = i + 1;
 }

//Sets the updated value to the root node. Updates the EmployeeXmlVariable's (the name of
the XMLvariable which is defined in the workflow)root node with the data specified above (
xstring )

_WorkFlowContext.XmlVariables["EmployeeXmlVariable"].SetNodeValue("//Employees/Employee",
submittedvalues);
//_WorkFlowContext is the runtime context object

//_WorkFlowContext is the runtime context object
Workflow.NET.Engine.Context _WorkFlowContext;
int executionId = 123;

//Gets the workflow Context for the specified execution id where executionId is the
Execution Id (type int) of the workflow.
_WorkFlowContext = Workflow.NET.Engine.Context.GetContext(executionId, "MyRepository");

//Getting the value of the Name node from the Employee Xml
Object[] values =
_WorkFlowContext.XmlVariables["EmployeeXmlVariable"].GetNodeValues("//Employees/Employee/
Name");

//If there are many Xml Variables we can loop through the Xml variable collection to get
the required Xml Variable
//foreach (Workflow.NET.XmlVariable xVariable in //_WorkFlowContext.XmlVariables.Values)
// {
 //Getting the node values of a Xml variable
// if(xVaraiable.Name == "EmployeeXmlVariable")
// {
 // GetNodeValues returns an object array
// Object[] values = xVariable.GetNodeValues("//Employees/Employee/Name");
// break;
// }
//}
```

XML String(value of the XML variable):
<Employees><Employee><Name>Kevin</Name><Department>Testing</Department></Employee><Employee><Name>J

Output: values[0]: Kevin

values[1]: John

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Getting the XMLNodeList from XML Variable

```
//_WorkFlowContext is the runtime context object
Workflow.NET.Engine.Context _WorkFlowContext;
int executionId = 123;

//Gets the workflow Context for the specified execution id where executionId is the
Execution Id (type int) of the workflow.
_WorkFlowContext = Workflow.NET.Engine.Context.GetContext(executionId, "MyRepository");

//Getting XmlNodeList from Xml Variable
XmlNodeList nodeList=
_WorkFlowContext.XmlVariables["EmployeeXmlVariable"].GetXmlNodes("//Employees/Employee");

//if there are more no. of Xml Variables we can loop through, verify and get the Xml
NodeList
//foreach (Workflow.NET.XmlVariable xVariable in _WorkFlowContext.XmlVariables.Values)
//{
//  Getting the nodes of a Xml variable
//  if(xVariable.Name == "EmployeeXmlVariable")
// {
// XmlNodeList nodelist = //xVaraiable.GetXmlNodes(("//Employees/Employee");
// break;
// }
//}
```

<Employees><Employee><Name>Kevin</Name><Department>Testing</Department></Employee><Employee><Name>J

Output: Gives us the Employee collection(XML Node List)

XmlNode1:Employee – Kevin

XmlNode2:Employee – John

Create Web Service Using Windows Application

This example illustrates how to create a Web Service in a Windows application.

PROCEDURE

Dll REFERENCE

```
Workflow.NET.NET2
Skelta.HWS
```

NAMESPACE USED

```
System
System.Data
System.Web
System.Web.Services
System.Web.Services.Protocols
Workflow.NET
Skelta.HWS
Skelta.Core
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. In Visual Studio, open ASP.NET Web Service.
2. Add reference "Skelta.HWS.dll" and "WorkFlow.Net.Net2.dll".
3. Write the code in Appcode\Service.cs

```
using System;
using System.Data;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using Workflow.NET;
using Skelta.HWS;
using Skelta.Core;
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Service : System.Web.Services.WebService
{
ApplicationObject application = null;
WorkflowObject workflow = null;
Actor actor = null;
public Service ()
 {

//Uncomment the following line if using designed components
//InitializeComponent();
application = new ApplicationObject("WebServiceRep");
workflow = new WorkflowObject("WebServiceWF", application);
}

[WebMethod]
public string HelloWorld()
{
return "Hello World";
}

[WebMethod]
public DataSet GetWorkItemList(string realActorId)
{
WorkItemCollection workItemCollection = new WorkItemCollection(application, workflow,
new
Actor(application, realActorId), false, "WebWorkList");
return workItemCollection.GetRecords();
}

[WebMethod]
public DataSet GetWorkItemQueueList(Guid queueId, string realActorId)
{
WorkItemCollection workItemCollection = new WorkItemCollection(application, workflow,
queueId, new Actor(application, realActorId));
return workItemCollection.GetRecords();
}

[WebMethod]
public void SubmitWorkItem(Guid workitemId, string statusSubmitted)
{
WorkItem workItem = new WorkItem(workflow, workitemId);
workItem.Submit("WebWorkList", statusSubmitted, "", "");
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}

[WebMethod]
public string[] GetWorkItemOutputs(Guid workitemId)
{
WorkItem workItem = new WorkItem(workflow, workitemId);
System.Collections.Generic.Dictionary<string, string>  outputs =
((Skelta.HWS.IHWSActivityProvider)workItem.HWSActivity.CurrentActivity.ActionHandler)
.GetPossibleStatusList();
string[] statusList = new string[outputs.Count];
outputs.Keys.CopyTo(statusList, 0);
return statusList;
}

[WebMethod]
public void PickNextItem(Guid queueId, string realActorId)
{
Skelta.HWS.Queue.Queue queue = new Skelta.HWS.Queue.Queue(application, queueId);
Actor actor  = new Actor(application, realActorId);
Skelta.HWS.WorkItem workitem = queue.GetNextItem(queueId, actor);

workitem.PickItem(actor, true);
}

}
```

4. Design a simple workflow. Publish it and associate with an event. Note that the workflow name should be the same as mentioned above.
5. To check whether the web service you created is working or not, right-click "Service.asmx" in the Solution Explorer, and click "View in Browser". A web page should appear.
6. Use the web service in the Window Application Form. Create a Window Application in Visual Studio. For example, WebServiceApplication1.
7. Design the Form. Apply the Proper Naming Convention. Add Reference of "System.web.Services."
8. Go to Visual Studio Command Prompt.

Navigate to the Windows Application Directory in the Command Prompt. Type the following command.

wsdl /language:CS /n:"Microsoft.SqlServer.ReportingService2005"

http://localhost:1283/SupportWf/Service.asmx?WSDL(this should be URL of the Web Service)

This step is to create a code file of the Web Service. Make use of the web service class, create an object of it and start using its methods.

1. Type the code behind for the Form.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace WebServiceApplication1
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
public partial class Form1 : Form
{
 SupportWF sf = new SupportWF();//creating an object of Web Service Class
 string realActorId = "skeltalist::fbb172c5-5daf-4e5d-bcff-6500b95004f1";
 Guid workitemId = Guid.Empty;
 Guid queueId = new Guid("dad821c0-eec7-464d-879c-14b76d3e4a55");
public Form1()
{
InitializeComponent();
}

private void GetWorkItems_Click(object sender, EventArgs e)
{
DataSet ds = sf.GetWorkItemList(realActorId);
workitemId = Guid.Empty;

dataGridView1.DataSource = ds.Tables["CollectionBaseTable"];

dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

}

void dataGridView1_RowEnter(object sender,
System.Windows.Forms.DataGridViewCellEventArgs e)
{
comboBox1.Items.Clear();
workitemId = new Guid(((DataGridView)sender).Rows[0].Cells[0].Value.ToString());
string[] outPuts = sf.GetWorkItemOutputs(workitemId);
foreach (string strValue in outPuts)
{
if (!comboBox1.Items.Contains(strValue))
comboBox1.Items.Add(strValue);
}
}
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{

}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{

}

private void GetQueueItems_Click(object sender, EventArgs e)
{
DataSet ds = sf.GetWorkItemQueueList(queueId, realActorId);
workitemId = Guid.Empty;
dataGridView1.DataSource = ds.Tables["CollectionBaseTable"];
dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;

comboBox1.Items.Clear();

}

private void PickQueueItem_Click(object sender, EventArgs e)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
sf.PickNextItem(queueId, realActorId);

}

private void Submit_Click(object sender, EventArgs e)
{
string comboBoxValue = comboBox1.Text;
sf.SubmitWorkItem(workitemId, comboBoxValue);
}
}
}
```

2. Run the application. Click **GetWorkItems**, it will display the current item of the Workitem List. (Approval Activity).
3. Approval Activity has two outputs, Approved or Rejected. So in the Combo Box, it will show two WorkItem tasks. Select any WorkItem Task (Approved or Rejected) and click **Submit**. The task gets submitted to the WorkItem.
4. Click **GetQueueItems** button. The current item in the Queue will be displayed.
5. Click **PickQueueItem**, select any Queue item and click **PickQueueItem**. It will pick the item from the Queue.

SOA

SOA consists of SOA repository and SOA consumption using AVEVA Work Tasks Expression Editor and SOA Activities. SOA repository contains the registered SOA Folders and SOA assemblies. Assemblies can be registered from the GAC (Global Assembly Cache) or from the SOA folders. Once an assembly is registered it will be available for consumption in AVEVA Work Tasks.

To use it in a workflow, you need to instantiate the required class in the Start Activity. The required constructor parameter, if any, can be set while instantiating the class. Public Methods available in the class could be accessed in expression builder properties of available actions.

Examples:

- How to Use SOA in a Simple Method?
- How to Use a Web Service or Remote Service using SOA Infrastructure?
- How to Use an Application that Interacts with the SQL Server?
- How to Use Static Members of the Class?
- How to Pass the Object as Parameter to the Constructor?
- How to Use Scripting in Expression Editor?
- How to Use Expression in Constructor Parameter?
- Execute Methods with Out Parameters in SOA
- Retrieve Multiple Records from Database Using SOA

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

How to Use SOA in a Simple Method?

This example illustrates the use of SOA in a simple method in a .NET framework assembly

PROCEDURE

NAMESPACE USED

```
System
System.Collections.Generic
System.Text
```

To create a New Project

1. Launch **Microsoft Visual Studio**. The Visual Studio Start Page opens.
2. On the **File** menu, point to **New**, and then select **Project**. The New Project dialog box appears.
3. In the right pane, select **Class Library**.
4. Create your .net Class Library. For example, in the **Name** box, type MySkeltaTest.
5. Set the path to save the project. For example, in the **Location** drop-down, set the location as M:\Work.
6. In the **Solution Name** box, type the directory name. By default, it displays the directory name as the specified project name.
7. Click **OK**. The MySkeltaTest project gets created in the path M:\Work.

   Sample code to create the class

```
using System;
using System.Collections.Generic;
using System.Text;
namespace MySkeltaTest
{
public class MyTest
{
private int id;
public MyTest()
{
}
public MyTest(int inde)
{
id = inde;
}
public string GetName()
{
return "Welcome to Skelta SOA";
}
public string GetNextName()
{
return "Wonderful feature";
}
public string Unique_Name(int iID)
{
if (iID == 1return "Use WEB Service";
}
else if (iID == 2)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
return "Access SQL Server data";
}
else if (iID == 1)
{
return "Access COM object";
}
else
return "What not";
}
}
}
```

8. Build the created project. The dll gets created in M:\Work\MySkeltaTest\MySkeltaTest\bin\Debug folder.

   Adding this folder in the SOA Folder option:

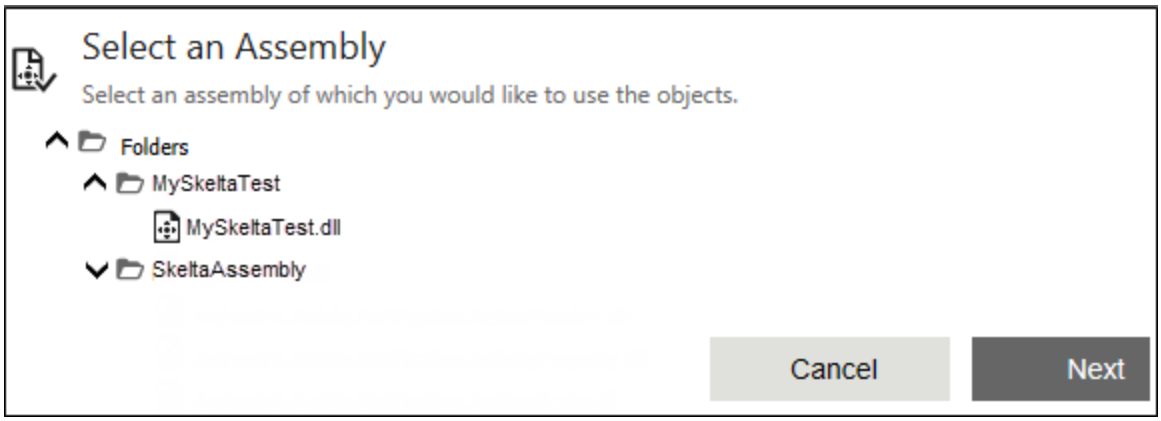
9. On the Windows Start menu, point to **All Programs**, point to **Wonderware**, point to **AVEVA Work Tasks**, and then click **Central Configuration**. The Central Configuration page appears.
10. Click Repository List. The Repository List page appears.
11. In the Site Path column, click the desired link to log on to the desired repository. You would be redirected to the Enterprise Portal.
12. Select the required **Authentication Provider**.
13. Log on to the **Enterprise Console**.
14. In the **Enterprise Console**, click the **Menu** button, click Settings, and then click SOA Folder. The Folder List page appears.
15. On the action bar, click **New**. The **Folder Configuration**- New wizard appears.
16. In the **Title** box, type a name for the folder, and in the **Description** box, type a description for the folder.
17. In the **Physical Path** box, type the physical path. For example, M:\Work\MySkeltaTest\MySkeltaTest\bin\Debug.
18. Click **View Assemblies**. All the assemblies available in the given path will be listed in the List of Assemblies section.
19. Click **Save**. The SOA folder is created.

    Adding Assembly into SOA Assembly Repository

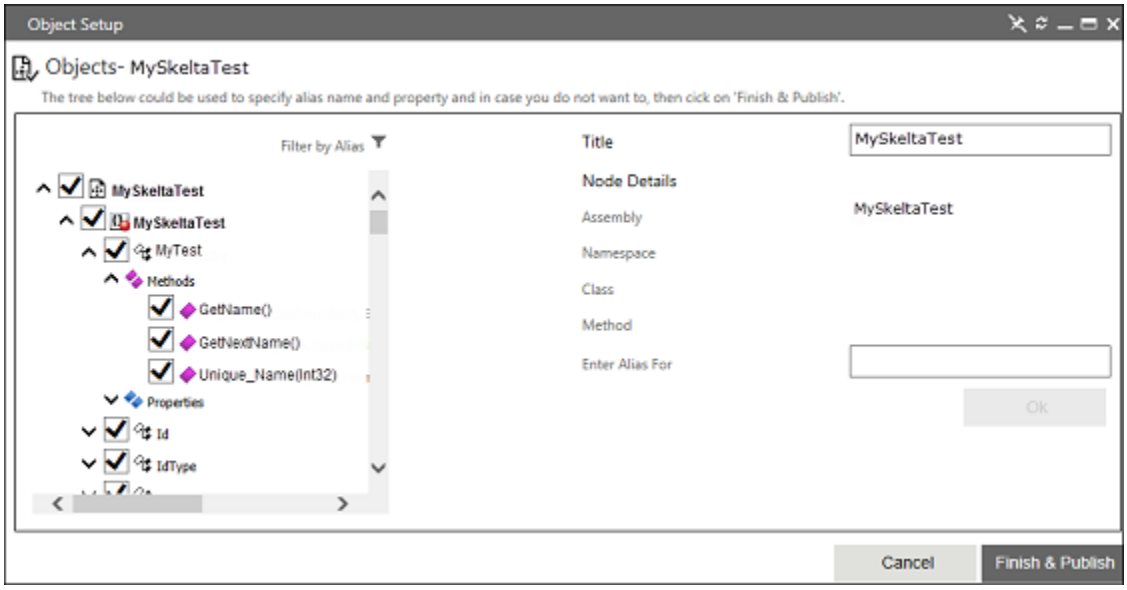
20. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **SOA Assembly**. The **Assembly List** page appears.
21. On the action bar, click **New**, and then click **From Assembly Folder** to select an assembly from a registered folder. The **Select an Assembly** page appears.
22. Select the assembly. The assembly must be a .NET Framework assembly.

    After you select an assembly, you can browse through the namespaces, classes, and methods of the assembly. You can also provide an alias for the selected methods.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



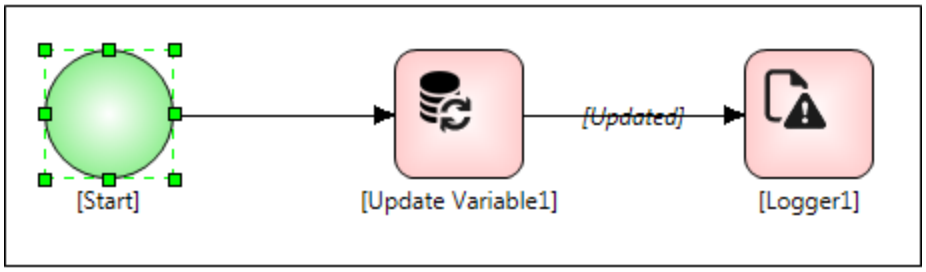
23. Click **Next**. The **Object Setup** page appears.
24. Select methods to provide an alias.



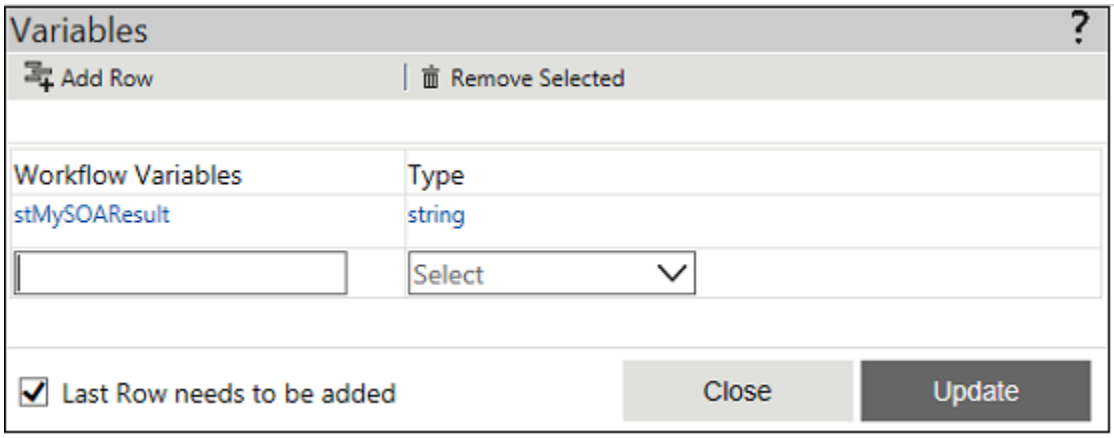
Note: By default, all the classes and methods are checked. If you do not want to use any particular class or method, then uncheck the class or method. However, the unchecked class will not be displayed in the Start Activity of the SOA Object Instance property. The unchecked class will be displayed in the Expression Editor of any activity, but the data will not be available.

25. Click **Finish** & **Publish**.
26. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**. The Workflows page appears.
27. Design a workflow. The **Process Designer** appears.
28. In the **Activity Properties** of the Start activity, select SOA Object Instances property to instantiate a class for the specific purpose. The **SOA Object Instance** page appears.
29. Select the appropriate constructor for the specified class.
30. Click **Save** to save the object instance.
31. Design a simple workflow that has two activities, Update **Variable Activity** and **Logger Activity**. You need the Update activity to update a variable, and the Logger activity to display a value.

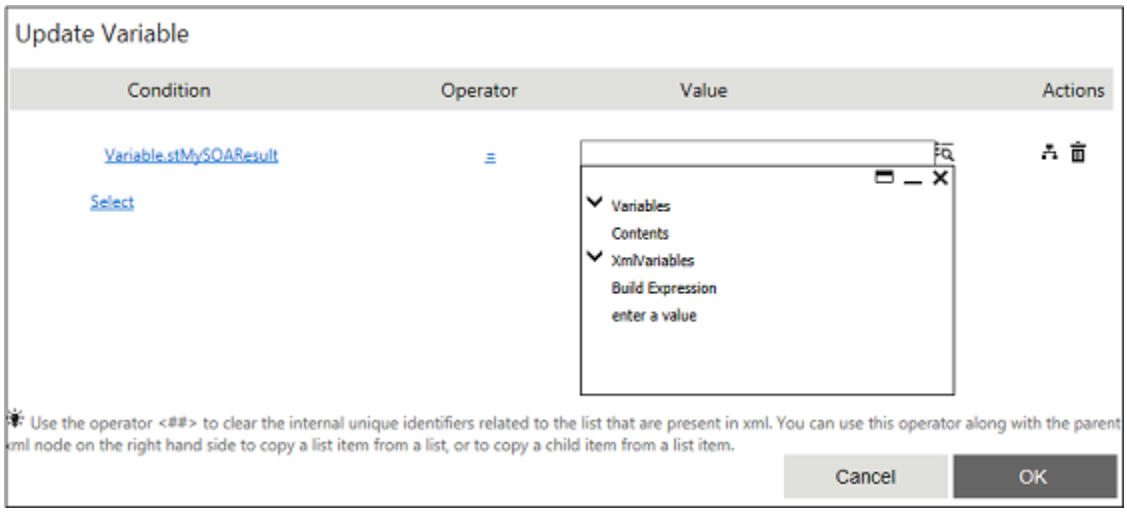
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



32. In the **Activity Properties** of the Start activity, click the **Variable** property to create a variable of type string. For example, you can name the variable as stMySOAResult.

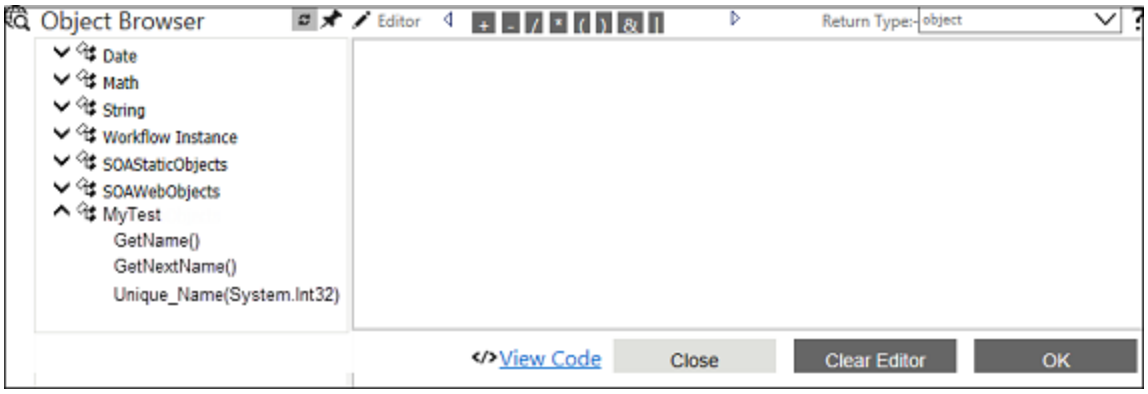


33. Click **Update**.
34. In the **Activity Properties of the Update Variable** activity, select **Set Variable** property. The **Update Variable** page appears.
35. In the **Condition** column, select the variable.

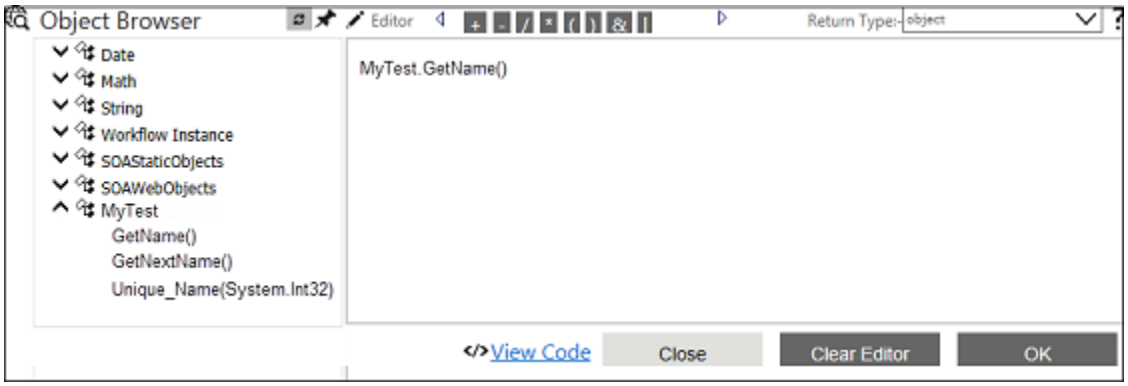


36. In the **Value** column, select **Build Expression**.

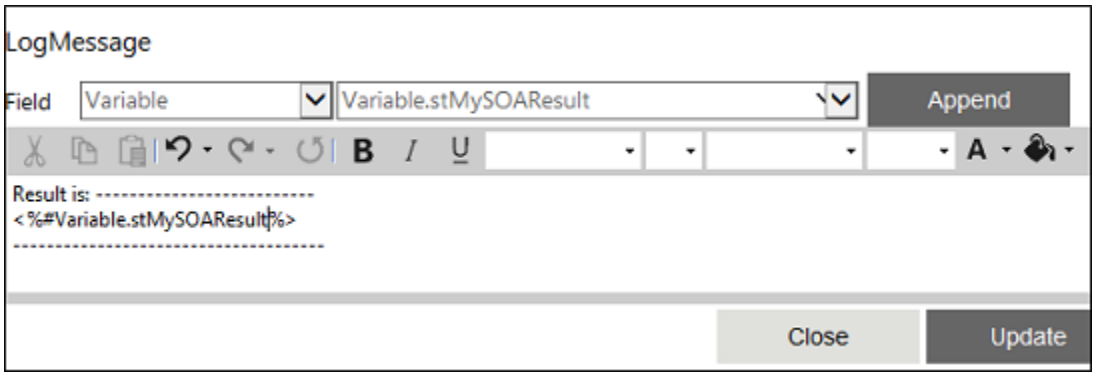
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



37. Use the instance method. For example, My Test.



38. In the Activity Properties of the Logger activity, select the LogMessage property. The **LogMessage** page appears.



39. Type the log message to be displayed.
40. Click **Update**. The result appears in the **Logger Console**.

How to Use a Web Service or Remote Service using SOA Infrastructure?

This example illustrates how to use a web service or remote service using SOA infrastructure. The example is a very simple one which can be modified to have a more practical scenario.

PROCEDURE

NAMESPACE USED

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
System
System.Web
System.Web.Services
System.Web.Services.Protocols
```

To create a New Web Service

1. On the **Windows Start** menu, point to **All Programs**, and then click **Microsoft Visual Studio**. The Visual Studio Start Page appears.
2. In the **File** menu, point to **New**, and then click **Web Site**. The **New Web Site** dialog box appears.
3. In the **Templates** pane, select **ASP.NET Web Service**. This is selected by default, and you can retain the default.
4. Enter a name for the website by changing the default name in the **Location** text box, for example, M:\Work\WebSite8. You can retain the default selection of File System in the Location drop down.
5. Click **OK**.

   Sample Code for creating Web service using System;

```
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
[WebService(Namespace = "http://skeltaSoa.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Service : System.Web.Services.WebService
{
public Service ()
{

//Uncomment the following line if using designed components
//InitializeComponent();
}

[WebMethod]
public string SkeltaTest()
{
return "This is my SketlaTest";
}

}
```

To create a Web Site

Complete the following steps to create a web site to host the created web service:

1. Open IIS Manager.
   - For Windows Server 2012, on the Start page click the Server Manager tile, and then click OK. In the Server Manager, click the Tools menu, and then click Internet Information Services (IIS) Manager.
   - For Windows 8.1, on the Start page type Control Panel, and then click the Control Panel icon in the search results. On the Control Panel screen, click System and Security, click Administrative Tools, and then click Internet Information Services (IIS) Manager.
2. In the Connections pane, right-click the Sites node in the tree, and then click Add Website.
3. In the Add Website dialog box, enter a friendly name for your website in the Site name box.
4. If you want to select a different application pool than the one listed in the Application Pool box, click Select.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

In the Select Application Pool dialog box, select an application pool from the Application Pool list and then click OK.

5. In the Physical path box, enter the physical path of the website's folder, or click the browse button to navigate the file system to find the folder.
6. If the physical path that you entered in the previous step is to a remote share, click Connect as to specify credentials that have permission to access the path. If you do not use specific credentials, select the Application user (pass-through authentication) option in the Connect As dialog box.
7. Select the protocol for the website from the Type list.
8. If you must specify a static IP address for the website (by default, this is set to All Unassigned), enter the IP address in the IP address box.
9. Enter a port number in the Port text box.
10. Optionally, enter a host header name for the website in the Host Header box.
11. If you do not have to make any changes to the site, and you want the website to be immediately available, select the Start Web site immediately check box.
12. Click OK.

Creating Proxy dll

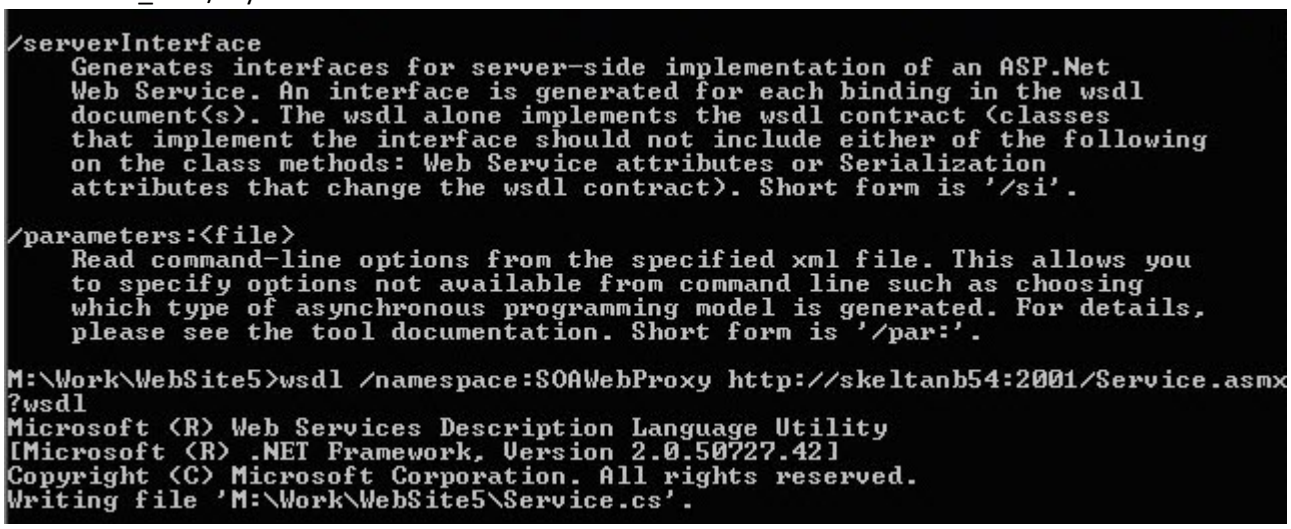
We have to create a Proxy dll since the proxy does all the marshalling and serialization/deserialization of parameters and returns to/from a web service.

Creating proxy dll using WSDL

Create the proxy class of the web service using the WSDL tool. Use the wsdl.exe utility to generate code for XML web service clients and XML web service using ASP.NET from WSDL contract files, XSD schemas, and discomap discovery documents. This tool can be used in conjunction with disco.exe.

In Visual Studio Command Prompt, run the following command:

C:\Program Files\Microsoft Visual Studio\VC> wsdl /namespace:SkeltaSOAWebProxy http://localhost:1508/WebService_New/MyWebService.asmx?wsdl



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To create proxy dll using CSC command

1. Create DLL using CSC command.
2. Create a new project 'SOAWebServiceUse'.
3. Right-click 'SOAWebServiceUse' project which created and click Add Reference.
4. Add reference 'SkeltaSOAService.Dll' to use.

Place this compiled proxy class in the dll and refer the same from where you want to access the Web Service.

Create a .net Assembly to consume the Web Service method.

```
using System;
using System.Collections.Generic;
using System.Text;
using SOAWebProxy;

namespace SOAWebSeriveUsage
{
public class SOAWebSeriveUsage
{
public SOAWebSeriveUsage() { }
public string GetWebServiceData()
{
SOAWebProxy.Service ser = new SOAWebProxy.Service();
return ser.SkeltaTest();
}
}
}
```

Do not forget to add the Proxy.dll in the reference.

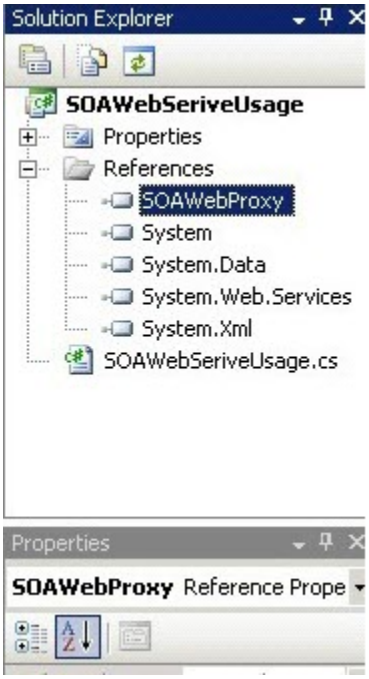


Figure - Solution Explorer

Adding New SOA Folder

The very basic component of SOA repository is the Folder. Basically you need to register a folder to consume the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

assemblies from the folder. Add the reference folder into the SOA Folders.

Adding new folder in the SOA Folder option:

1. Go to **Start > All Programs > AVEVA > Work Tasks > Central Configuration**.
2. Go to the **Repository List**.
3. Click Site Path from which you would like to login to the repository. This redirects to the Enterprise Portal.
4. Click arrow button to enter into the **Enterprise Console**.
5. In the **Enterprise Console**, enter the **User Name** and **Password**. Click the icon next to the Password field to login.
6. From the **Enterprise Console** menu, select **Tools > SOA Assembly**.
7. In the ribbon bar, click **New Folder**.
8. In the **Title** text box, enter the folder name as 'MySkeltaTest'.
9. In the **Description** text area, enter a description for the folder.
10. In the **Physical Path** text box, enter the physical path to verify the presence of the assembly using the view assembly option.
11. Click **Verify Assemblies**.
12. Click **Save**.

Adding Assembly

There are two options to add assembly into the SOA repository i.e. **From GAC or From Assembly Folder**.

To add an assembly in the SOA repository, choose the **SOA > Assemblies** option from the Enterprise Console.

Note:

- If the web service is not available, the system shows the warning message:
“Unable to access < Web service url>. The web service call will be ignored for further processing. Exception message ...".
The methods consumed in the Workflow will not be executed.

- After upgrading the product, validation may not happen when you click on **View Code** or **OK** button in the Expression Editor. If you get an "Access Denied" message in the Logger Console, delete the SOA Web service dll from the Windows Temp folder and access it again.

- If the SOA Execute activity configured with SOA Web service is not working, you need to delete that activity and reconfigure it.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

How to Use an Application that Interacts with the SQL Server?

This example illustrates how to use an application which interacts with SQL server.

PROCEDURE

NAMESPACE USED

```
System
System.Collections.Generic
System.Text
System.Data.sqlClient
System.Data
using System;
using System.Collections.Generic;
using System.Text;
using System.Data.SqlClient;
using System.Data;
namespace HumanResource
{
public class EmployeeMaster
{
private string _Server;
public string ServerName
{
get { return _Server; }
set { _Server = value; }
}
private string _User;
public string User
{
get { return _User; }
set { _User = value; }
}
private string _Password;
public string Password
{
get { return _Password; }
set { _Password = value; }
}
private string _ID;
public string EmployeeID
{
get { return _ID; }
set { _ID = value; }
}
private string _Name;
public string Name
{
get { return _Name; }
set { _Name = value; }
}
private string _Address;
public string Address
{
get { return _Address; }
set { _Address = value; }
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
static SqlConnection objConnect;
static string sConnection = "";
public EmployeeMaster(string sServer, bool bWindowsAuthent, string sUser, string sPasswd)
{
string sdatabaseName = "HRMS";
if (sServer == string.Empty)
sServer = "skeltanb54";
if ((sUser == string.Empty) || (sPasswd == string.Empty))
{
bWindowsAuthent = true;
sConnection = "Data Source=" + sServer + "; Database=" + sdatabaseName + ";
Trusted_Connection = true" ;
}
else
{
sConnection = "Data Source=" + sServer + "; Database=" + sdatabaseName + ";UID=" + sUser +
"; PWD=" + sPasswd;
}
//objConnect = new SqlConnection(sConnection);
//objConnect.Open();
}
~EmployeeMaster()
{
if (objConnect != null)
objConnect = null;
}
public SqlConnection GetDataBaseConnection()
{
try
{
if (objConnect == null)
{
objConnect = new SqlConnection(sConnection);
objConnect.Open();
}
else
{
if ( objConnect.State == ConnectionState.Closed)
objConnect.Open();
}
}
catch (Exception ex)
{
throw ex;
}
return objConnect;
}
public void CloseDatabaseConnectio()
{
objConnect.Close();
objConnect = null;
}
public string GetEmployeeName(string id)
{
try
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
string sSQL = "select * from Employees where employeeID = '" + id + "'";
GetDataBaseConnection();
SqlDataAdapter adp = new SqlDataAdapter(sSQL, objConnect);

DataSet ds = new DataSet();
adp.Fill(ds);
_Name = ds.Tables[0].Rows[0]["Name"].ToString();
_ID = ds.Tables[0].Rows[0]["EmployeeID"].ToString();
_Address = ds.Tables[0].Rows[0]["Address"].ToString();
return ds.Tables[0].Rows[0]["Name"].ToString();
}
catch (Exception ex)
{
throw ex;
}

}
public void GetEmployeeDetails(string id)
{
try
{
string sSQL = "select * from Employees where employeeID = '" + id + "'";
GetDataBaseConnection();
SqlDataAdapter adp = new SqlDataAdapter(sSQL, objConnect);

DataSet ds = new DataSet();
adp.Fill(ds);
_Name = ds.Tables[0].Rows[0]["Name"].ToString();
_ID = ds.Tables[0].Rows[0]["EmployeeID"].ToString();
_Address = ds.Tables[0].Rows[0]["Address"].ToString();
}
catch (Exception ex)
{
throw ex;
}
}
public string GetConnectEmployeeName( SqlConnection SqlConn, string id)
{
try
{
string sSQL = "select * from Employees where employeeID = '" + id + "'";
SqlDataAdapter adp = new SqlDataAdapter(sSQL, SqlConn);

DataSet ds = new DataSet();
adp.Fill(ds);
_Name = ds.Tables[0].Rows[0]["Name"].ToString();
ID = ds.Tables[0].Rows[0]["EmployeeID"].ToString();
_Address = ds.Tables[0].Rows[0]["Address"].ToString();
return ds.Tables[0].Rows[0]["Name"].ToString();
}
catch (Exception ex)
{
throw ex;
}
```

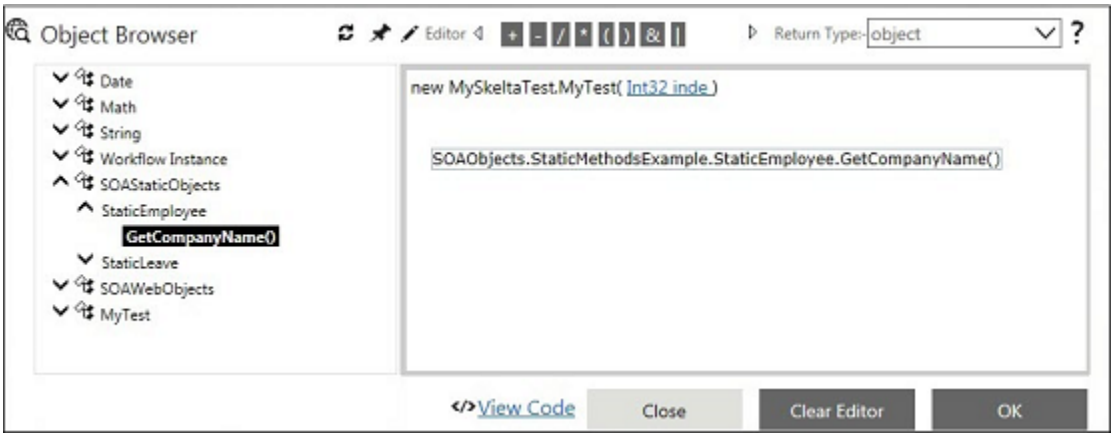
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
}
}


Table Structure

Employees – Main table to store employee details

CREATE TABLE [dbo].[Employees](
[ID] [int] IDENTITY(1,1) NOT NULL,
[EmployeeID] [nchar](10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Name] [nvarchar](50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
[Address] [nvarchar](max) COLLATE SQL_Latin1_General_CP1_CI_AS NULL,
[DOJ] [datetime] NULL
) ON [PRIMARY]


Table to store Leave Balance

CREATE TABLE [dbo].[LeaveBalance](
[ID] [int] NULL,
[CL_Balance] [numeric](18, 2) NULL,
[SL_Balance] [numeric](18, 2) NULL
) ON [PRIMARY]
```

How to Use Static Members of the Class?

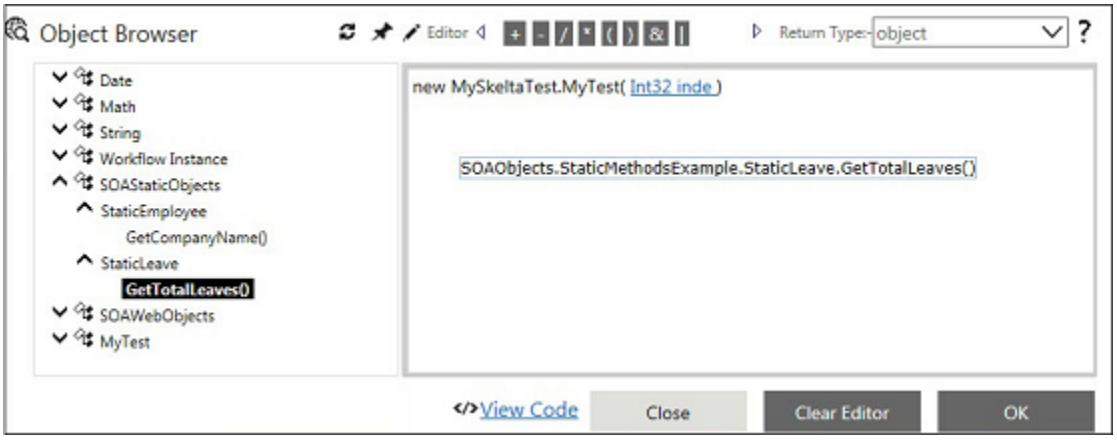
This example shows how to use static members of the class in SOA Property Editor/Expression Editor. To use Static members, there is no need to create the instance of the class. After registering an assembly into the SOA Assembly list, all the static members of the registered class will get loaded into the SOA Property Editor/ Expression editor and can make use of the same for passing the parameter.

Static methods of a class in SOA Property Editor



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Static methods of a class in Expression Editor:

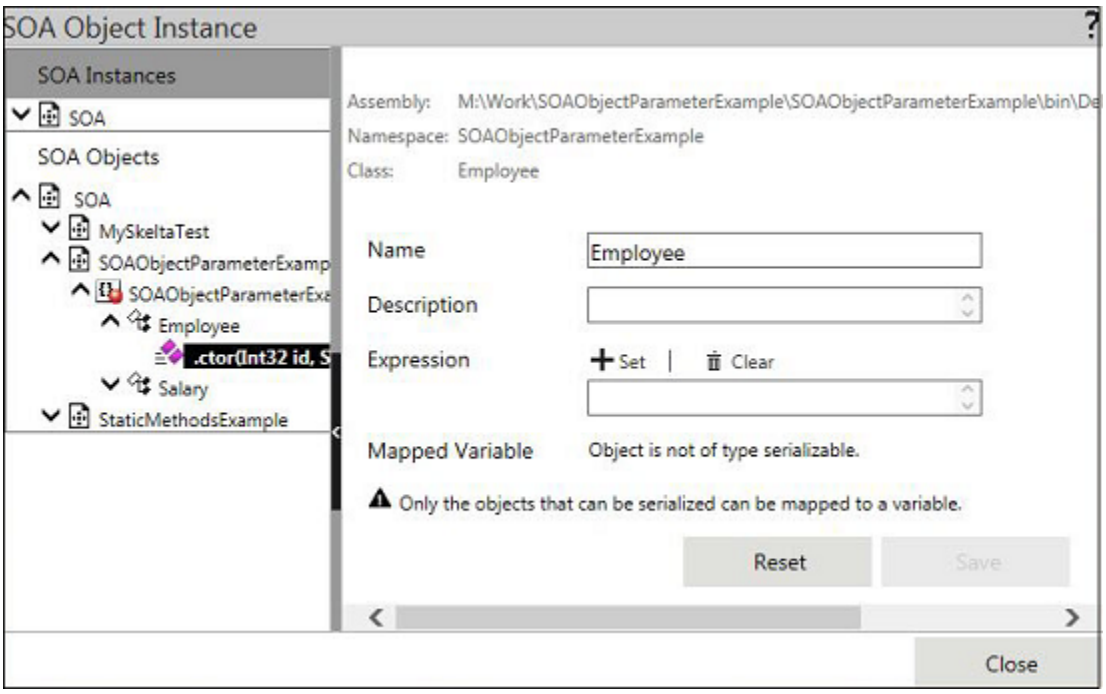


How to Pass the Object as Parameter to the Constructor?

This example shows how to use the object as parameter to the constructor parameter.

For example, consider 'Employee' and 'Salary' to be two classes of an assembly. Employee object can be created by passing the employee information as parameter to the constructor. If 'Salary' object takes 'Employee' object as parameter, then it can be created by making use of the previously created instance 'Employee'.

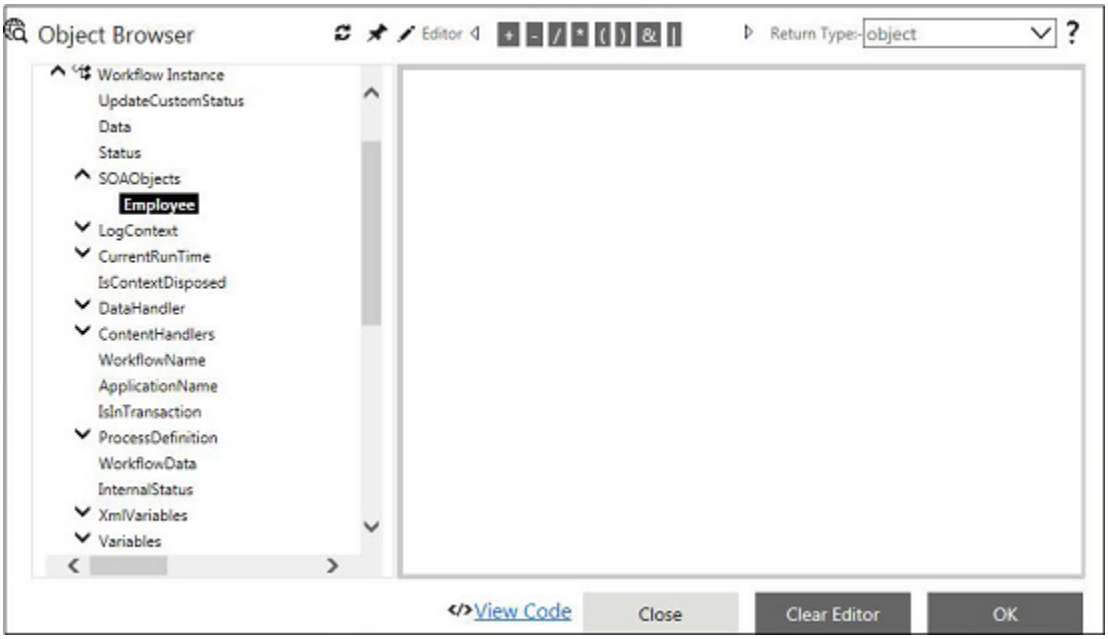
Step1: Create an instance of a class by using SOA Property Editor.



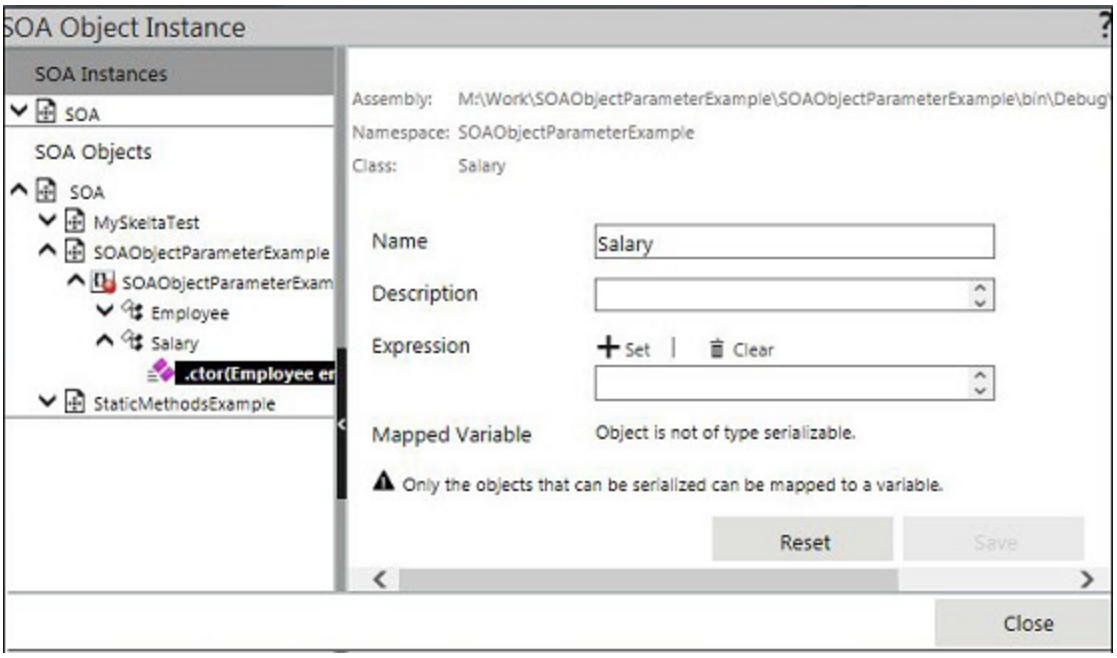
Step 2: All saved instances will be listed in Object Browser/Workflow Instances/SOA Objects.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Step 3: Click and drop as a parameter of the constructor of another instance which takes the object as parameter.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Note: There is no need to type cast the parameter value as specific object when we use the object instance as parameter. This is because the compiler takes care of type casting to specific type of object.

How to Use Scripting in Expression Editor?

In some scenarios, users want to execute a series of expressions together and return the value of the last statement. This can be achieved by separating the statements by semi colon (;).

The below example shows how one can execute multiple statements and return the value of the last statement.

If the user wants to execute one or more expressions in SOA Expression Editor, then the statements can be separated by using semicolon (;).

Example:

In the below example, there are three statements.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. int a = 10;
2. Variable.varResult = a;
3. Employee.GetEmployeeName();

Here, the execution sequence takes place from the first statement.

Step Details:

1. Variable 'a' with integer type will be created with the value 10.
2. AVEVA Work Tasks Variable 'Variable.varResult' will be updated with the value of 'a' that is 10.
3. SOA Expression 'Employee.GetEmployeeName();' will be executed and value will be returned.

Note: By using this feature, there is no need to use the 'Update Variable' activity for simple data updation to any AVEVA Work Tasks Variables.

How to Use Expression in Constructor Parameter?

This example shows how to pass expression as parameter to the constructor parameter.

.ctor (int EmployeeID, string EmployeeName, string Department, string Designation, int Age, double Salary)

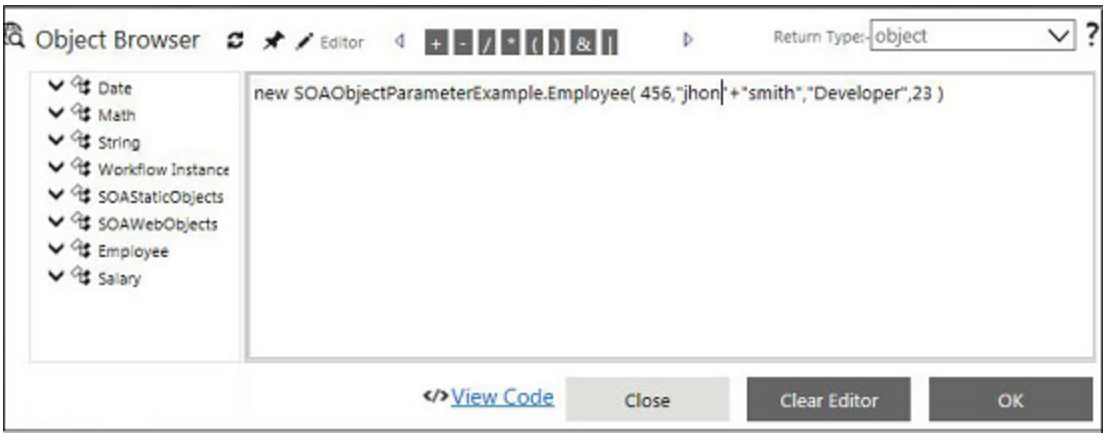
The above constructor takes 6 parameters.

Constructor parameters can be passed like this.

.ctor (5, "ABC" + "XYZ", "R & D", "Analyst", 30, 50000)

Here, the second parameter is passed as the expression: **"ABC" + "XYZ"** which is evaluated at the time of execution and updated with the value "ABCXYZ."

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



SOA Object Support

The aim of the Object Support in AVEVA Work Tasks Object Access (SOA) is to enhance the capability of executing and persisting real time entities in long running workflows.

Features Supported for SOA

Features supported

- Support for variables of type Object to hold any real time entity that can be serialized i.e. marked as 'Serializable'.
- Support for mapping the variable to a constructor in the SOA property editor. This maps the variable to the constructor and during execution the variable gets updated with the object.
- Persistence of the real time entity or object in long running workflows.
- Changing the property of an object stored in the variable at runtime. This increases the flexibility in handling the object.

Scenario

Telematics organization is one of the leading automobile manufacturing companies. It needs to buy spare parts for manufacturing automobile from various spare parts manufacturers. This happens at all stages of manufacturing. It requires approval from the concerned manager.

Telematics organization needs to automate the process of buying Radiators. In this scenario, Radiator as a product comes under the Product entity.

The above scenario can be achieved by using AVEVA Work Tasks Object Access (SOA) to build the 'Product' entity and hold it in the long running process. Persistence of the 'Product' entity in the workflow enhances its capability to handle the complex objects in long running workflows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

How to Create a Class for Handling the 'Product' Object?

The scenario explained in the previous page can be designed by following the steps given in the examples in the next few pages.

To create a Class for Handling the 'Product' Object

Note: You can make use of any existing class to build the scenario.

'Product' class: Product information.

'OrderLineCalculator' class: Operations performed on 'Product' object.

Note: Only the objects that can be serialized can be persisted in long running workflows. Objects that need to be stored in the AVEVA Work Tasks variables should be marked as 'Serializable'.

```
using System;
using System.Collections.Generic;
using System.Text;
namespace Product
{
/// <summary>
/// Product Class
/// </summary>
[Serializable]
public class Product
{
string _code = string.Empty;
string _name = string.Empty;
string _description = string.Empty;
double _unitPrice = 0.0;

/// <summary>
/// Product code
/// </summary>
public string Code
{
get { return _code; }
set { _code = value; }
}

/// <summary>
/// Product name
/// </summary>
public string Name
{
get { return _name; }
set { _name = value; }
}

/// <summary>
/// Product Description
/// </summary>
public string Description
{
get { return _description; }
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
set { _description = value; }
}

/// <summary>
/// Price
/// </summary>
public double UnitPrice
{
get { return _unitPrice; }
set { _unitPrice = value; }
}
public Product(string code, string name, string description, double price)
{
this._code = code;
this._name = name;
this._description = description;
this._unitPrice = price;
}
}

/// <summary>
/// Order Class
/// </summary>
public class OrderLineCalculate
{
public OrderLineCalculate() { }

/// <summary>
/// Get Total amount to be paid
/// </summary>
/// <param name="product">Product</param>
/// <param name="quantity">Quantity</param>
/// <returns>Total Price to be paid</returns>
public double GetAmount(Product product, int quantity)
{
return (product.UnitPrice * quantity);
}
}
}
```

1. Using the above code creates an assembly. For e.g. Product.dll
2. Place the assembly in the SOA folder, the folder from which the SOA assemblies are registered. For e.g. M:\SOA\Assemblies\Product.dll.
3. Register the folder (M:\SOA\Assemblies) in the AVEVA Work Tasks Object Access folder.
4. Register the 'Product' assembly [Product.dll] in SOA Assemblies from the SOA folder.
5. The registered assembly will get displayed in the SOA assembly list.

How to Create an Object Instance?

The below example shows how the user can create an Object Instance.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To create an Object Instance

Instance creation is required to invoke the members of the object.

1. Create an instance of 'Product' object. This can be done in the SOA property editor of the **Start Activity**.

   Instantiating SOA Objects

   Instantiating objects can be done by making use of the constructors of the object. The SOA property editor allows to create an object instance by using constructors.

   Instantiating and persisting the object instance can be done in different ways. This is explained below.

   a. **Creating an instance of the object by using 'Constructors'**

      Description: In the above example, we have 'Product' object and in order to use the Product object in AVEVA Work Tasks workflows, we need to create an object instance of the 'Product'. Creating object instance can be done in the SOA property editor by using the constructor defined for the 'Product' object.

      Steps:

      a. Click the constructor defined for 'Product' object.
      b. Click 'Set' to set the expression/define the constructor parameters. It will open the Expression Editor for setting the expression.
      c. Pass the appropriate parameters to the constructor expression.
      d. Save the expression.

      By creating an object instance, it is possible to build an expression with the methods and properties of the 'Product' class. In the workflow, whenever an expression with the 'Product' object is encountered, the workflow engine will create an instance of the 'Product' object and resolve the expression. Instance creation happens in all the places wherever the expression is associated with the 'Product' object. In this way, the expression with methods and properties of an object gets resolved.

   b. **Creating an instance of the object using 'Constructors' and mapping the object to a variable**

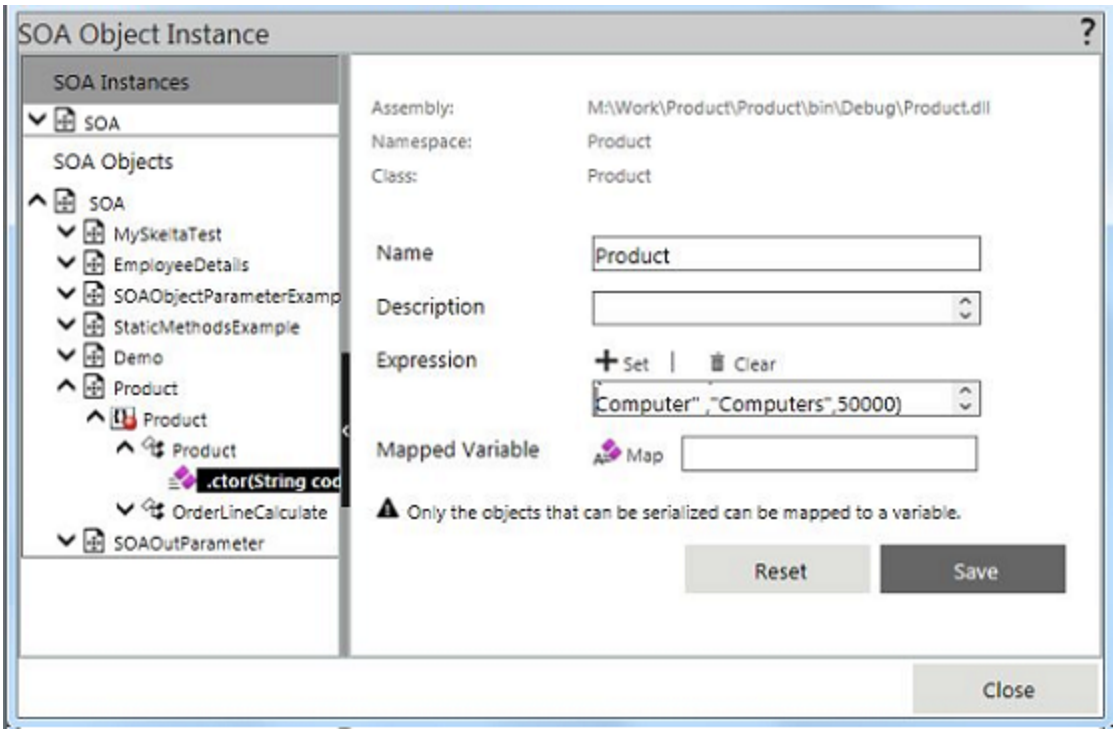
      Description: Considering the first approach, in some scenarios there may be a requirement of storing the object and persist it in the long running workflows.

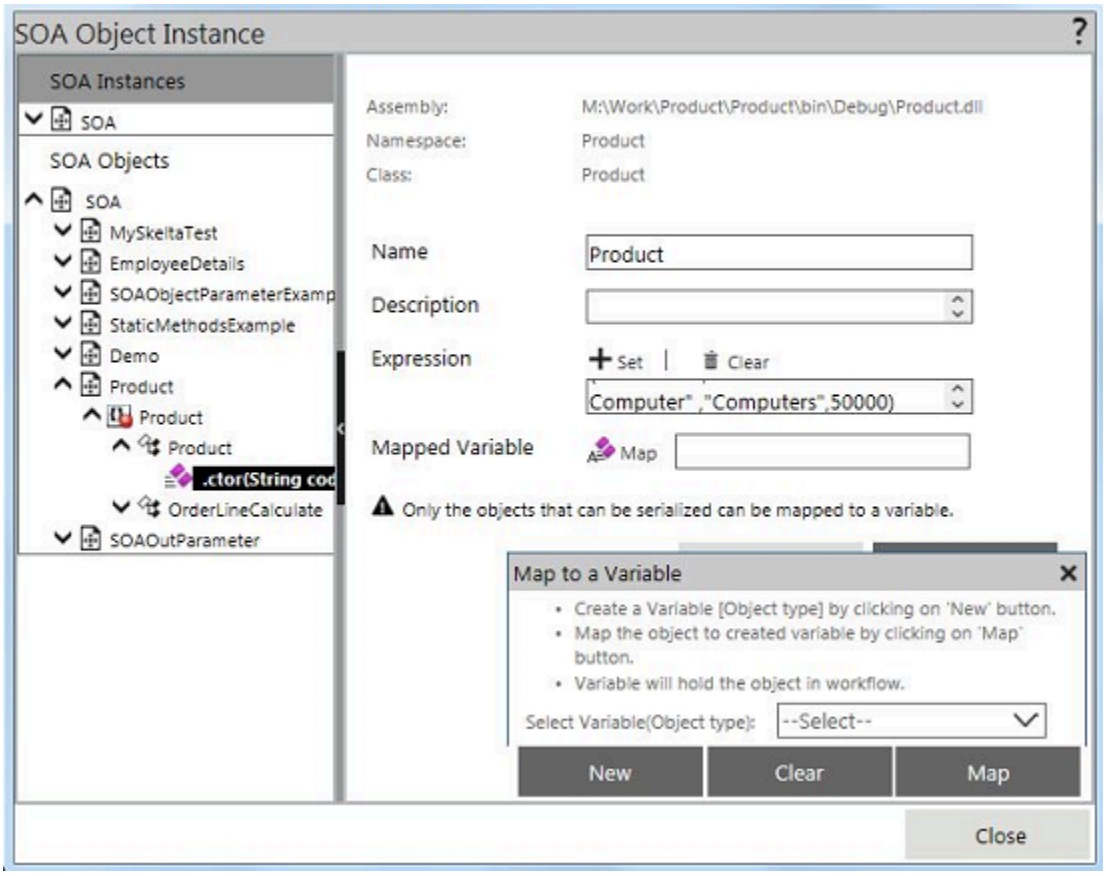
      SOA object mapping provides an option to store object state in a variable. This increases the flexibility in handling the complex object in long running workflows.

      Mapping an object to a variable will persist the object state in the workflow. This happens for the first time whenever an object is instantiated; when you invoke the object after that, it will return the object information which is stored in the variable. This will not only hasten the process of resolving the expression built with the object mapped to a variable, but also persist the object state in long running workflows.

      Mapping an object to a variable can be done at the time of configuring the instance in the 'SOA Property' editor. This approach is explained in depth in the next topic.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



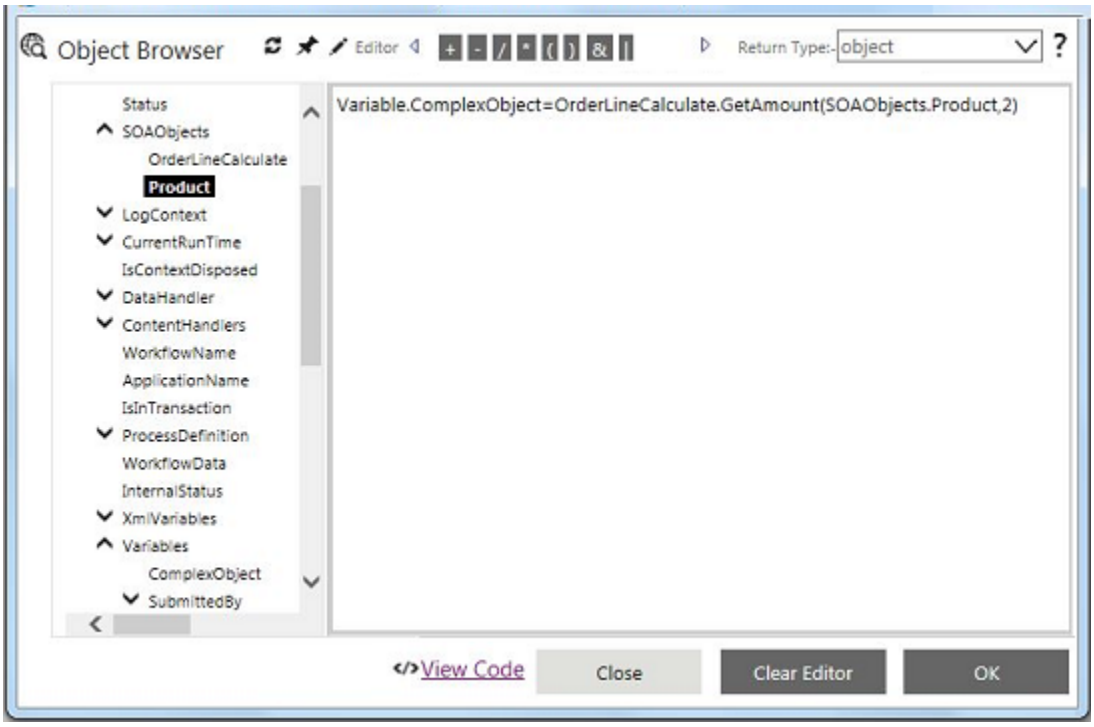


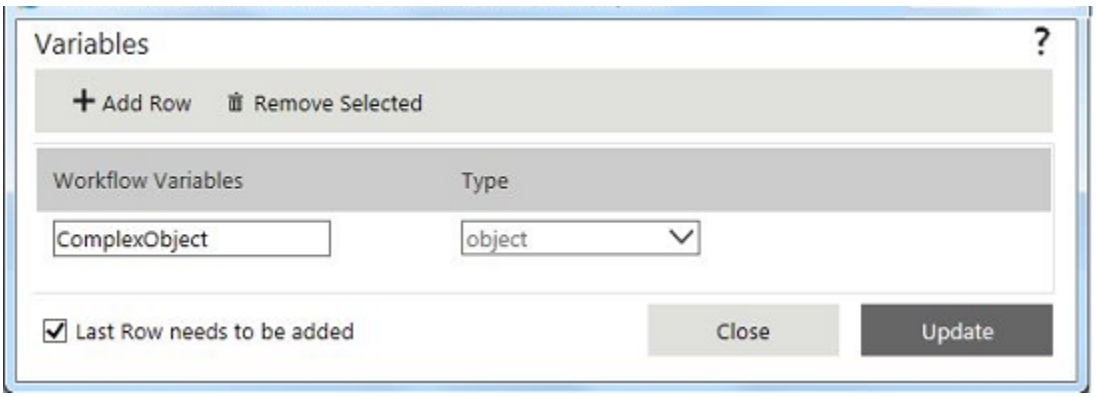
Note: Only objects that are Serializable can be mapped to a variable/persist in the workflow. Variable mapping should be used only when an object needs to be persisted in long running workflows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

c. **Mapped Variables**

Description: Any object can be directly mapped to a variable without any constructor. In some scenarios, there may be a requirement to store a complex object in a variable. For e.g.: the 'Employee' object returned from a web service.





This can be done by simply assigning the object to a variable in the SOA Expression Editor.

How to Create a Variable?

The below example shows how the user can create a Variable.

To create a Variable

1. Create a variable of type 'Object' in the Start Activity, for e.g. 'ProductObject'. This variable will hold the 'Product' object in the process. This variable is also used for object mapping. Mapping of an object to a variable will store the object state in the variable. This will enhance the capability of persisting the real time entities in the long running workflow.

   The following steps will explain how to create an instance of object (Approach (a)), and how to map an

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

object to a variable (Approach (b)).

'Product' object with all the available constructors will be displayed in the 'SOA Objects'. Objects that can be serialized will have the option 'Map to Variable'.

Note: 'Mapped variable' feature gets enabled only for objects that are marked as 'Serializable'.

2. Create a 'Product' object instance by selecting a constructor and pass the appropriate parameters to it.
3. Click 'Set' to set the expression to the constructor.
4. Pass the valid parameters to the constructor. E.g.: "Rad01","Radiator","Spare Part Automobile", 20000

   Map the variable to the object. The Mapped variable will get automatically filled with the real time entity whenever the object gets instantiated. This happens for the first time, and further invoking the object will get the object state stored in the variable.
5. Click 'Map to Variable' to map a variable to the instance. Only object type variables can be mapped to SOA Objects. A pop-up with 'Map to Variable' will get displayed.
6. Select the variable to be mapped and click 'Map'. This will map the variable to the SOA Object. It is possible to create a variable of type object by clicking 'New' and mapping the variables.
7. The Mapped variable will get displayed in the 'Mapped Variable' field.
8. After the variable is mapped, click 'Save' to save the instance.
9. Create an instance of 'OrderLineCalculator' class. This object will be used for calculating the total amount.

How To

In AVEVA Work Tasks, learn how to:

- Add users to an active directory
- Create custom wizards
- Return values from the engine to the client
- Implement the interface to get Action and Workflow status change information from the engine
- Get the username with the domain name format

Add Users to Active Directory

To add users to the **Active Directory**, do the following:

1. Select **Administrative Tools** from the **Control Panel** and point to **Active Directory Users and Computers**.
2. The **Active Directory Users and Computers** window opens.
3. Right click **Users** and point to **New** to add a new user. New **Object - User** wizard opens.
4. Enter the details of the new user.
5. Click **Next** or **Cancel** to abort. **New Object - User** password wizard opens.
6. Enter the password for the new user and deselect the check boxes.
7. Click **Next** or **Cancel** to abort. **New Object - User** final wizard opens.
8. Click **Finish**.
9. Double-click the new user added. **Administrator Properties** window opens.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

10. Click **Member Of** tab from the administrator properties window.
11. Click **Add**. The **Select Groups** window opens.
12. Enter Administrator and click **Check Names**.
13. Click **OK**.

    Note: Users who have administrator privileges can log into the Enterprise Console.

14. Make sure that all the users have logged into the Enterprise Console site for the concerned repository at least once.
15. Click **Organization** tab from the administrator properties window.
16. Enter the **Title** and click **Change**.
17. Enter a name and click **Check Names**.
18. Click **OK**.

Create Custom Wizards

AVEVA Work Tasks provides a set of interfaces and their supporting classes and members that can be implemented in the class of a custom activity to create a custom wizard. The following procedure creates an Email Wizard that is available in the Process Designer in the menu bar as well as under the Wizard category in the activities tool box.

PROCEDURE

Dll REFERENCE

```
System
System.Data
System.XML
System.DirectoryServices
Workflow.NET.NET2
```

NAMESPACE USED

```
System
System.Collections.Generic
System.Text
Skelta.Wizards.Core.Interfaces
Workflow.NET
Skelta.Core
System.ComponentModel
Skelta.Wizards.Core
System.XML
```

The following steps and code illustrate the creation of a custom email wizard. The custom email wizard is listed in the AVEVA Work Tasks Wizards window.

Step 1:

1. Create a new project of type class library and name as CustomWizards.
2. Add a reference to Skelta.Wizards.Core.dll and Workflow.NET.NET2.dll.
3. Add a new class file called EmailWizard.cs. This class file should implement the interface IWizard and ISkeltaAddInProvider.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Implement the functions as follows.

```
using System;
using System.Collections.Generic;
using System.Text;
using Skelta.Wizards.Core.Interfaces;
using Workflow.NET;
using Skelta.Core;
using System.ComponentModel;
using Skelta.Wizards.Core;
using System.Xml;
namespace CustomWizards
{
public class EmailWizard : IWizard ,ISkeltaAddInProvider
{
///Constructor  defining the outputs of the wizard and setting the title and description
///properties of the wizard
public EmailWizard()
{
_Outputs.Add("Sent");
_Outputs.Add("Not Sent");
_Description ="EmailWizard Description";
_Title = "Email Wizard";
}
#region IWizard Members
///Implementing the Category property of the interface
private string _Category;
public string Category
{
get { return _Category; }
}
//Implementing the Description property of the interface
private string _Description = string.Empty;
public string Description
{
 get { return _Description; }
 set { _Description = value; }
}
///Implementing the Display Name property of the interface
private string _DisplayName = string.Empty;
[DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]
public string DisplayName
{
get { return _DisplayName; }
set { _DisplayName = value; }
}
//Implementing the Favorites property of the interface
private Dictionary<string, string> _Favourites = new Dictionary<string, string>();
[DesignerSerializationVisibility(DesignerSerializationVisibility.Hidden)]
public Dictionary<string, string> Favourites
{
 get { return _Favourites; }
 set { _Favourites = value; }
}
///Implementing the GetDiagram method of the interface for creating Actions and Links
and dumping the same into process designer
public void GetDiagram(Diagram diagram, Workflow.NET.Action previousAction, string
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
output)
{
Workflow.NET.Action startAction = diagram.Actions["Start"];
if (previousAction == null)
{
 previousAction = startAction;
 output = string.Empty;
}
string email1ActivityName = string.Empty;
while (true)
{
 email1ActivityName = ActivityNamingContainer.Instance.GetName("Email");
 if (!diagram.Actions.ContainsKey(email1ActivityName))
 break;
}
 Workflow.NET.Action emailAction = Utilities.CreateAction(diagram,
 previousAction.ProcessDefinition.ActionsDefinition, "Email", email1ActivityName,
"Email",
 previousAction, string.Empty);
 emailAction.Description = "Email";
//setting the properties of the Email Activity i.e. From,To etc
 (emailAction.Properties["From"] as Property).Value = new
 Workflow.NET.Config(diagram.ApplicationName, diagram.WorkflowName).FromEmailAddress;
 (emailAction.Properties["To"] as Property).Value =
Utilities.GetResourceFilter(diagram.ActionsDefinition, this._To);
 (emailAction.Properties["Subject"] as Property).Value = this._Subject;
 (emailAction.Properties["Body"] as Property).Value = this._Body;
Utilities.CreateLink(diagram, previousAction, emailAction, "Reminder");
CreateChildDiagrams(diagram, emailAction);
}

///Implementing the CreateChildDiagrams method of the interface
///Creating the diagram from the outputs of an action
protected void CreateChildDiagrams(Workflow.NET.Diagram diagram, Workflow.NET.Action
previousAction)
{
foreach (string key in _Outputs)
{
WizardActionTypes wizardActionType = _OutputActions[key].ActionType;
switch (wizardActionType)
{
case WizardActionTypes.LinkToWizard:
_OutputActions[key].NextWizard.GetDiagram(diagram, previousAction, key);
 break;

case WizardActionTypes.LinkToTemplate:
_OutputActions[key].NextTemplate.GetDiagram(diagram, previousAction, key);
break;

case WizardActionTypes.LinkToActivity:
Utilities.CreateLink(diagram, previousAction.Name,
_OutputActions[key].NextActivity, key);
break;

case WizardActionTypes.NoLink:
break;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
}
}


///Implementing the GetWebUIServerControl method of the interface
///To get the name of the UI control of the wizard
public string GetWebUIServerControl
{
get { return "EmailWizardUI.ascx"; }
}

///Implementing the ImageName property of the interface
private string _ImageName;
public string ImageName
{
get { return _ImageName; }
}

///Implementing the IsNewUIInstance property of the interface
///Checks whether this is a new instance of the wizard
private bool _IsNewUIInstance = true;
public bool IsNewUIInstance
{
get
{
return _IsNewUIInstance;
}
set
{
_IsNewUIInstance = value;
}
}

///Implementing the IsStartupWizard property of the interface
///Checks whether the wizard is a startupwizard ie one that is ///used only for
provisioning variables
private bool _IsStartupWizard = true;
public bool IsStartupWizard
{
get { return _IsStartupWizard; }

}

///Implementing the Name property of the interface
public string Name
{
get { return "EmailWizard"; }
}

///Implementing the OutputActions property of the interface
///Each wizard has a collection of output actions
/// eg LinkToWizard , LinkToTemplate which is specified in the ///Linking page
 Dictionary<string, WizardAction> _OutputActions = new Dictionary<string,
WizardAction>();
 [DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 public Dictionary<string, WizardAction> OutputActions
 {
 get { return _OutputActions; }
 }

///Implementing the Outputs property of the interface
///Each wizard has a collection of outputs eg in the case
///of Email Wizard the outputs are "Sent" and "Not Sent"
 private List<string> _Outputs = new List<string>();
 [DesignerSerializationVisibility(DesignerSerializationVisibility.Hidden)]
 public List<string> Outputs
 {
 get { return _Outputs; }
 }

///Implementing the ParentId property of the interface
///Each wizard may or may not have a parent wizard which referred ///it . This property
returns the id of that parent wizard
 private Guid _ParentId;
 [DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]
 public Guid ParentId
 {
get { return _ParentId; }
set { _ParentId = value; }
 }

///Implementing the SelectedIndex property of the interface
///Returns the selected index of the selected output of the ///wizard
 private int _SelectedIndex = -1;
 [DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]
 public int SelectedIndex
 {
 get
 {
 return _SelectedIndex;
 }
 set
 {
 _SelectedIndex = value;
 }
 }
///Implementing the SelectedOutput property of the interface
///Returns the selected output of the wizard
public string SelectedOutput
{
get { return _Outputs[_SelectedIndex]; }
}

///Implementing the TemplateUniqueId property of the interface
///Returns the id of the template of which this wizard is linked ///to
private Guid _templateUniqueId;
public Guid TemplateUniqueId
{
get { return _templateUniqueId; }
set { _templateUniqueId = value; }
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
///Implementing the Title property of the interface
///Returns the title of the wizard
 private string _Title = string.Empty;
 [DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]
public string Title
{
get { return _Title; }
set { _Title = value; }
}

///Implementing the UniqueId property of the interface
private Guid _uniqueId;
public Guid UniqueId
{
get { return _uniqueId; }
set { _uniqueId = value; }
}

#endregion

#region ISkeltaAddInProvider Members

///Implementing the Id property of the interface
private Guid _Id;
public Guid Id
{
get { return _Id; }
}

///Implementing the InitializeProvider property of the interface
///Takes the appropriate row from the SKAddInProviders table
///and gets the "SettingsXml" column value
public void InitializeProvider(string settings, Guid id)
{
this._Settings = settings;
this._Id = id;
if (_Settings != string.Empty)
{
//Setting Favourites from xml
XmlDocument xmldoc = new XmlDocument();
xmldoc.LoadXml(_Settings);
/*This is the XML SCHEMA for 'Settings' Field in the table 'SKAddInProviders'

 <Wizard>
 <Category>
 </Category>
 <ImageName>
 </ImageName>
 <IsStartupWizard>
 </IsStartupWizard>
 <Favourties>
<FavouriteElement>
<FavouriteElement>
 </Favourites>
 </wizard> */
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
XmlNodeList _category1 = xmldoc.GetElementsByTagName("Category");
XmlNodeList _imagename1 = xmldoc.GetElementsByTagName("ImageName");
XmlNodeList _isstartupwizard1 = xmldoc.GetElementsByTagName("IsStartupWizard");
XmlNodeList _favourties1 = xmldoc.GetElementsByTagName("FavouriteElement");

if (_category1[0] != null)
_Category = _category1[0].InnerText;

if (_imagename1[0] != null)
_ImageName = _imagename1[0].InnerText;

if (_isstartupwizard1[0] != null)
_IsStartupWizard = Convert.ToBoolean(_isstartupwizard1[0].InnerText);

for (int i = 0; i < _favourties1.Count; i++)
_Favourites.Add("FavouriteElement_" + i, _favourties1[i].InnerText);
}
}

///Implementing the Settings  property of the interface
///Manages the  "SettingsXml" column value got from ///SKAddInProviders table
private string _Settings;
public string Settings
{
get { return _Settings; }
}


#endregion

 ///"To" Property defined for Email Wizard
private string _To = string.Empty;
[DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]
public string To
{
get
{
return _To;
}
set
{
_To = value;
}
}

 ///"From" Property defined for Email Wizard
private string _From = string.Empty;
[DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]
public string From
{
get
{
return _From;
}
set
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
_From = value;
}
}

///"Subject" Property defined for Email Wizard
private string _Subject = string.Empty;
[DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]
public string Subject
{
get
{
return _Subject;
}
set
{
_Subject = value;
}
}

///"Body" Property defined for Email Wizard
 private string _Body = string.Empty;
[DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]
public string Body
{
get
{
return _Body;
}
set
{
_Body = value;
}
}
}
}
```

Note: If you wish to create a property whose value changes dynamically across wizards like the above Subject, Body etc then that property should be prefixed with the below piece of code "[DesignerSerializationVisibility(DesignerSerializationVisibility.Content)]" Otherwise if you prefer to use hard coded values, then prefix that property with the below piece of code to hide that property accordingly "[DesignerSerializationVisibility(DesignerSerializationVisibility.Hidden)]" . The above note is applicable during serialization and deserialization of data inside the Wizard Manager.

Step 2:

Build CustomWizards.dll and add the dll into the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder

Step 3:

1. Create a new ASP Web application **CustomWizardUI** and add user control page called **EmailWizardUI.ascx** which is your UI for the email wizard.
2. Use the <asp:Wizard> control to define your wizard steps.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Hide the wizard navigation buttons.

```
//EmailWizardUI.ascx
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="EmailWizardUI.ascx.cs" Inheri
ts="CustomWizardUI.EmailWizardUI" %>
<link href="styles/template-wizard.css" rel="stylesheet" type="text/css" />
<asp:Wizard ID="AspWizard" runat="server" ActiveStepIndex="0"  Width="100%" DisplaySideB
ar="False" Height="100%">
  <WizardSteps>
  <asp:WizardStep ID="WizardStep1" runat="server" Title="Step 1" >
  <table width="100%" align="center" cellpadding="0" cellspacing="0" border="0" >
  <tr>
 <td align="center" valign="top">
 <table width="100%" border="0" cellspacing="0" cellpadding="0">
 <tr>
 <td height="38" align="left" background="images/title-bar-bg.png">
 <img src="images/visual-screen-
title.png" width="39" height="38" hspace="4" align="absMiddle" />
 <span class="runtext2-white">
  Email Wizard
 </span>
  </td>
  </tr>
  <tr>
  <td bgcolor="#378381" background="images/welcome-email-
bg2.png" valign="top" align="left"
  style="height: 131px">
  <img src="images/icon-email-now.png" hspace="10" vspace="2"  />
  <span style="font-family:Segoe UI, Trebuchet MS; font-size:15pt; color:White;">Welcome
to Email
 Wizard</span>
  </td>
  </tr>
  <tr>
  <td align="center" valign="middle" bgcolor="#f0f0f0">
  <br/><br/>
  <table width="90%"  border="0" cellpadding="4" cellspacing="2" bgcolor="#b9d1ea">
  <tr>
  <td align="left" valign="middle" bgcolor="#ffffff" class="runtext1">
  <blockquote>Email Wizard Description goes here  </blockquote>
  <input type="hidden" id="Hidden1" runat="server" />
  </td>
  </tr>
  </table>
  </td>
  </tr>
  </table>
  </td>
  </tr>
  </table>
  </asp:WizardStep>
  <asp:WizardStep ID="WizardStep2" runat="server" Title="Step 1" >
  <table width="100%" align="center" cellpadding="0" cellspacing="0" border="0" >
  <tr>
  <td align="center" valign="top">
  <table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
  <td>
  <asp:Label ID="tolabel" runat="server" ></asp:Label>
  </td>
  <td>
  <asp:TextBox ID="toTb" runat="server" ></asp:TextBox>
  </td>
  </tr>
  <tr>
  <td>
  <asp:Label ID="fromlabel" runat="server" ></asp:Label>
  </td>
  <td>
  <asp:TextBox ID="fromTb" runat="server" ></asp:TextBox>
  </td>
  </tr>
  <tr>
  <td>
  <asp:Label ID="subjectLabel" runat="server" ></asp:Label>
  </td>
  <td>
  <asp:TextBox ID="subjectTb" runat="server" ></asp:TextBox>
  </td>
  </tr>
  <tr>
  <td>
  <asp:Label ID="bodyLabel" runat="server" ></asp:Label>
  </td>
  <td>
  <asp:TextBox ID="bodyTb" runat="server" ></asp:TextBox>
  </td>
  </tr>
  </table>
  </td>
  </tr>
  </table>
  </asp:WizardStep>
  </WizardSteps>
  <StepNextButtonStyle CssClass="hide" />
  <StartNextButtonStyle CssClass="hide" />
  <FinishCompleteButtonStyle CssClass="hide" />
  <FinishPreviousButtonStyle CssClass="hide" />
  <CancelButtonStyle CssClass="hide" />
  <StepPreviousButtonStyle CssClass="hide" />
</asp:Wizard>
// End of EmailWizardUI.ascx
```

4. Apply the stylesheet template-wizard.css to the page. This exists in \BPMUITemplates\Default\Wizards\ styles folder.
5. Paste the file to the \BPMUITemplates\Default\Wizards\ folder.

Step 4:

1. Create the code behind for your user control. Your user control can use either code behind or library. Either way, it needs to inherit from IWizardUI interface and System.Web.UI.UserControl.
2. If you have used a class file as code behind, copy the same to \BPMUITemplates\Default\Wizards\ folder or

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

use as a library CustomWizardUI.EmailWizardUI.

3. Add the reference of the **CustomWizards.dll**
4. Write the necessary code for the wizard action.

```
// EmailWizardUI.ascx.cs
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using Skelta.Wizards.Core;
using Skelta.Wizards.Core.Interfaces;
using Workflow.NET;
using CustomWizards;


namespace CustomWizardUI
{
public partial class EmailWizardUI : System.Web.UI.UserControl, IWizardUI
{
private bool isValid = true;

protected void Page_Load(object sender, EventArgs e)
{
tolabel.Text = "To";
fromlabel.Text = "From";
subjectLabel.Text = "Subject";
bodyLabel.Text = "Body";
}

///Returns the Email Wizard from pre entered values
private IWizard GetWizard()
{
EmailWizard emailwizard = null;
if (IsPostBack)
{
emailwizard = (_WizardData as EmailWizard);

if (Request.Form[toTb.UniqueID] != null)
emailwizard.To = Request.Form[toTb.UniqueID];
if (Request.Form[fromTb.UniqueID] != null)
emailwizard.From = Request.Form[fromTb.UniqueID];
if (Request.Form[subjectTb.UniqueID] != null)
emailwizard.Subject = Request.Form[subjectTb.UniqueID];
if (Request.Form[bodyTb.UniqueID] != null)
emailwizard.Body = Request.Form[bodyTb.UniqueID];

}
return emailwizard;
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
///Sets  the Email Wizard with user entered values
public void SetWizard(IWizard wizardData)
{
EmailWizard spEventWizard = wizardData as EmailWizard;
toTb.Text = spEventWizard.To;
fromTb.Text = spEventWizard.From;
subjectTb.Text = spEventWizard.Subject;
bodyTb.Text = spEventWizard.Body;
}


#region IWizardUI Members


///Sets or gets  the wizard
private IWizard _WizardData;
public IWizard WizardData
{
get
{
if (_WizardData == null)
return null;
return GetWizard();
}
set
{
_WizardData = value;
SetWizard(_WizardData);
}
}

///Returns the total number of pages in the wizard UI
public int TotalPages
{
get { return AspWizard.WizardSteps.Count; }
}

///Sets or gets the current page of the wizard UI
public int CurrentPage
{
get
{
return AspWizard.ActiveStepIndex;
}
set
{
AspWizard.ActiveStepIndex = value;
}
}

///Sets or gets the Direction whether the user clicked "Next" ///or "Previous"
private NavaigationType _Direction = NavaigationType.Next;
public NavaigationType Direction
{
get
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
return _Direction;
}
set
{
_Direction = value;
}
}

///Sets or gets the ValidationStatus whether the
///wizard pages are valid or not
private bool _validationstatus;
public bool ValidationStatus
{
get
{
return isValid;
}
set
{
isValid = value;
}
}
#endregion
}
}
// end of EmailWizardUI.ascx.cs
```

Note: ValidationStatus property is helpful if you wish to validate the current page before moving on to the next one. If the page is not valid, this property can be set to false and vice versa.

Step 5:

Open your repository database and add a row to the table SKAddInProviders. The sql command is given below.

```
INSERT INTO
[SKAddInProviders]([Type],[Name],[Description],[ClassName],[Assembly],[Settings],[IsGacAsse
mbly],[LastUpdatedDatetime]) VALUES ('Wizard',<your wizard name>,<your wizard
description>,<your project name(which contains EmailWizard.cs)>,<location of your project
dll (which contains EmailWizard.cs)> , <xml string specifying the category of the wizard
and the favorites, image and whether the wizard is a start up wizard or not> ,0,getdate())
```

For example, the SQL Command for the above Email Wizard would be:

```
INSERT INTO
[SKAddInProviders]([Type],[Name],[Description],[ClassName],[Assembly],[Settings],[IsGacAsse
mbly],[LastUpdatedDatetime])  VALUES ('Wizard','EmailWizard','Email Wizard Description',
'CustomWizards.EmailWizard','bin\CustomWizards.dll','<wizard><Category>Human
Activities</Category><ImageName>Images\
Email.png</ImageName><IsStartupWizard>false</IsStartupWizard><Favourites><FavouriteElement>
ApprovalWizard</FavouriteElement></Favourites></wizard>',0,getdate())
```

The example row that is added for the Email Wizard. You can add any wizard of your own. You just need to specify which category it will belong to in the Settings column. In this case, the Wizard comes under the Human Activities category. Favorites of a wizard can also be specified in the Favourites tag. Whether the wizard is just used for provisioning i.e. IsStartupWizard can also be set in the IsStartupWizard tag.

Step 6:

1. Do not forget to paste the EmailWizardUI.ascx and EmailWizardUI.ascx.cs(code behind) files into

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

BPMUITemplates\Default\Wizards folder.

2. Add the necessary resource files into the Locales folder and the required styles into BPMUITemplates\Default\Wizards\styles folder and images into the BPMUITemplates\Default\Wizards\Images folder.

Step 7:

Copy the above created **CustomWizards.dll** into the bin folder of the **Enterprise Console** i.e.*[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\EnterpriseConsole\bin*

Your wizard is now ready.

Return Values from Engine to Client

The below code sample attempts to demonstrate the communication between the AVEVA Work Tasks engine and the client. For example, based on the property Applicationuri entered in the Activity Properties of the custom action named **"Agent app Navigator"** [It is created by using the below code], appropriate messages are displayed in the Logger Console.

PROCEDURE

Dll REFERENCE

```
System
Workflow.NET.NET2
```

NAMESPACE USED

```
System
System.Collections.Generic
System.Text
Workflow.NET
```

Steps to Create Custom Activity

Step 1

1. Define the class for the custom action's handler using the following code; build as a DLL and put this dll in *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Bin* folder.

```
using System;
using System.Collections.Generic;
using System.Text;
using Workflow.NET;

namespace AppNavigatorActivity
{
    public class AppNavigator :
Workflow.NET.Engine.Interfaces.IActionRun,Workflow.NET.Engine.Interfaces.IActionEventMes
sage
    {
        #region IActionRun Members

        Workflow.NET.Interfaces.IActivityWebUIHandler
        Workflow.NET.Engine.Interfaces.IActionRun.GetActivityWebUIHandler()
        {
            throw new Exception("The method or operation is not implemented.");
        }
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 Workflow.NET.Engine.Interfaces.ActionResult Workflow.NET.Engine.Interfaces.IActionRun.R
un(int    ExecutionID, int ExecutionDetailsID,
Workflow.NET.Engine.Context CurrentContext,
Workflow.NET.Action CurrentAction, string InlinkOutput, bool Retrying, out string Output
)
        {
            CurrentContext.log.LogInformation("Inside Action");
//Setting blocking output value for a workflow
            CurrentContext.BlockingOutput =
(string)((Property)CurrentAction.Properties["ApplicationURI"]).Value+","+ExecutionDetail
sID;
            Output = "";
            return Workflow.NET.Engine.Interfaces.ActionResult.Sleep;
        }

        #endregion

 #region IActionEventMessage Members

void Workflow.NET.Engine.Interfaces.IActionEventMessage.HandleEventMessage(int execution
ID, int executionDetailsID, Workflow.NET.Engine.Context currentContext,
Workflow.NET.Action currentAction, object eventMessage, bool retrying, out string output
, out Workflow.NET.Engine.Interfaces.ActionResult result, out bool ignoreOutputAndResult
)
{
      output = eventMessage.ToString().ToLower();
      result = Workflow.NET.Engine.Interfaces.ActionResult.Completed;
      ignoreOutputAndResult = false;
}

#endregion

    }
}
```

2. Define custom action using the following code, this must be added to the Actions.xml.

```
<category name= "BPO Activities" sortorder="8" image="custom.gif">
  <action name="ApplicationNavigator" displayname="Agent App Navigator">
    <description>Dummy action to replace the action types which are not
present.</description>
    <image resourcename="dummyaction.png"></image>
    <handler classname="AppNavigatorActivity.AppNavigator"
     assembly="AppNavigatorActivity.dll"></handler>
    <properties>
      <property name="ApplicationURI" type="string"></property>
    </properties>
          <return>
            <value helpstring="Next">Next</value>
            <value helpstring="Previous">Previous</value>
            <value helpstring="End">End</value>
          </return>
</action>
</category>
```

3. The custom action will be added to the Process Designer under the category BPO Activities (the category defined for the custom action).
4. Set the **ApplicationURI** property for the action.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Step 2

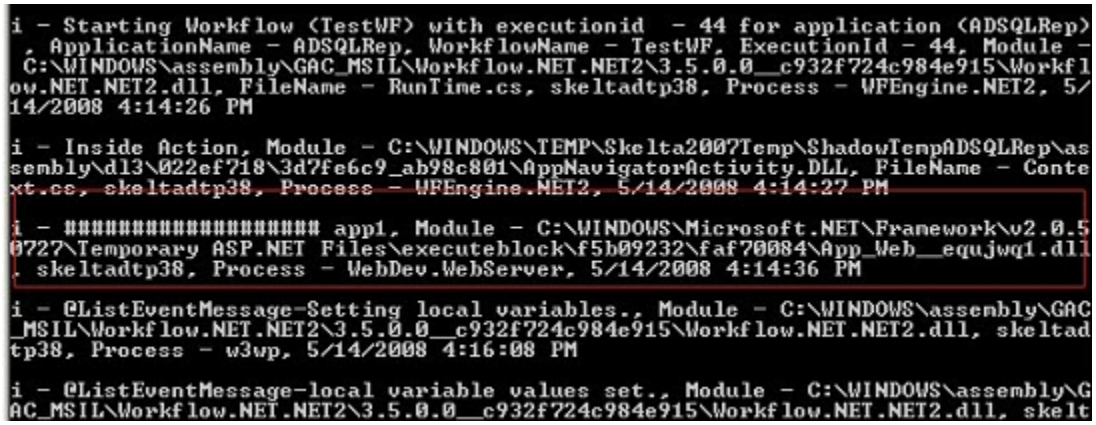
The value from the AVEVA Work Tasks engine can be returned to the client by using the out parameter in the Client.ExecuteBlocking method.

Create a web Application and include this code to check the returned values from the engine.

```
Workflow.NET.Engine.Client cl = new Workflow.NET.Engine.Client("Application", "workflow");
//Encapsulates the information required for redirecting the client.
Workflow.NET.Engine.SynchronousActionData syncData = new
Workflow.NET.Engine.SynchronousActionData(false, this.Context);
string url = "";
//Requests for the execution of the specified version of the workflow and waits till an
action in the workflow returns a URL for redirecting the client.
//The client can handle the redirection or leave the redirection to the application.
cl.ExecuteBlocking("activedirectory::bob", "<content></content>", syncData, true, out url);
cl.Close();
ProcessSyncData(syncData);

void ProcessSyncData(Workflow.NET.Engine.SynchronousActionData syncData)
{
if (syncData.IsExplicitBlockingOutput)
{
//The output returned from the workflow using BlockingOutput in Context is stored to the
variable "message"
string message = syncData.BlockingOutput.ToString().ToLower();
string[] msgs = message.Split(',');
message = msgs[0];
Log logger = new Log();

if (message == "app1")
logger.LogInformation("################### app1");
else if (message == "app2")
logger.LogInformation(" ############### app2");
else if (message == "app3")
logger.LogInformation(" ############### app3");
else
logger.LogInformation(" ############### No matching output");
}
```

Output:

The value typed in the property **ApplicationURI** of the custom activity is **app1** and a message "app1" is displayed in the Logger Console as shown below:



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Implement Interface to Get Action and Workflow Status Change Information from Engine

In scenarios where custom logic has to be implemented on the basis of the changes in the status of Workflows or Actions, we can make use of the IWorkflowInformation interface to get the Workflow events from the AVEVA Work Tasks engine.

We can only get the status and cannot change it using this interface. The class implementing the Workflow.NET.Interfaces.IWorkflowInformation can be used the get alerts from engine on changes in status.

Note: This interface cannot be used to get Workflow Execution Pending status. In the sample code, we will create a dll which logs custom message on to the logger based on the change in the workflow status.

PROCEDURE

Dll REFERENCE

System

Workflow.NET.NET2

NAMESPACE USED

System

Workflow.NET

Workflow.NET.Interfaces

Skelta.Core

A sample implementation code is shown below.

```
using System;
using Workflow.NET;
using Workflow.NET.Interfaces;
public class ExecutionInfo:IWorkflowInformation, ISkeltaAddInProvider
{
Log logger = new Log();
public ExecutionInfo()
{
logger.MachineName="";
logger.ModuleName="WFExecutionInfo";
logger.Source="";
logger.FileName="";
}

public void OnActionExecutedWithErrors(int ExecutionId, int ExecutionDetailsID, string
ActionName, Exception ObjException)
{
// TODO: Add ExecutionInfo.OnActionExecutedWithErrors implementation
logger.LogInformation("Testing: Action executed with errors for ExecutionId:"+ ExecutionId
+", ExecutionDetailsID:"+ ExecutionDetailsID +", ActionName:"+ ActionName +",
Exception:"+ObjException.Message);
}

public void OnWorkflowCompletedSuccessfully(int ExecutionId)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// TODO: Add ExecutionInfo.OnWorkflowCompletedSuccessfully implementation
logger.LogInformation("Testing: Workflow completed successfully for ExecutionId:"+
ExecutionId);
}

public string ApplicationName
{
get
{
// TODO: Add ExecutionInfo.ApplicationName getter implementation
return null;
}
set
{
logger.LogInformation("Testing: Application Name:"+ value);
}
}

public void OnActionStatusUpdate(int ExecutionId, int ExecutionDetailsID, string
ActionName, string Status)
{
// TODO: Add ExecutionInfo.OnActionStatusUpdate implementation
logger.LogInformation("Testing: Action status updated for ExecutionId:"+ ExecutionId +",
ExecutionDetailsID:"+ ExecutionDetailsID +", ActionName:"+ ActionName +", Status:"+Status);
}

public void OnWorkflowCompletedWithErrors(int ExecutionId)
{
// TODO: Add ExecutionInfo.OnWorkflowCompletedWithErrors implementation
logger.LogInformation("Testing: Workflow completed with errors for ExecutionId:"+
ExecutionId);
}

public void OnWorkflowStatusUpdate(int ExecutionId, string Status)
{
// TODO: Add ExecutionInfo.OnWorkflowStatusUpdate implementation
logger.LogInformation("Testing: Workflow status updated for ExecutionId:"+ ExecutionId +",
Status:"+ Status);
}

public void OnWorkflowStarted(int ExecutionId)
{
// TODO: Add ExecutionInfo.OnWorkflowStarted implementation
logger.LogInformation("Testing: Workflow started for Application:"+ ApplicationName +",
ExecutionId:"+ ExecutionId );
}

#region ISkeltaAddInProvider Members
Guid _Id = Guid.Empty;
public Guid Id
{
get { return _Id; }
}
public void InitializeProvider(string settings, Guid id)
{
_Settings = settings;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
_Id = id;
}
string _Settings = "";
public string Settings
{
get { return _Settings; }
}
#endregion
}
```

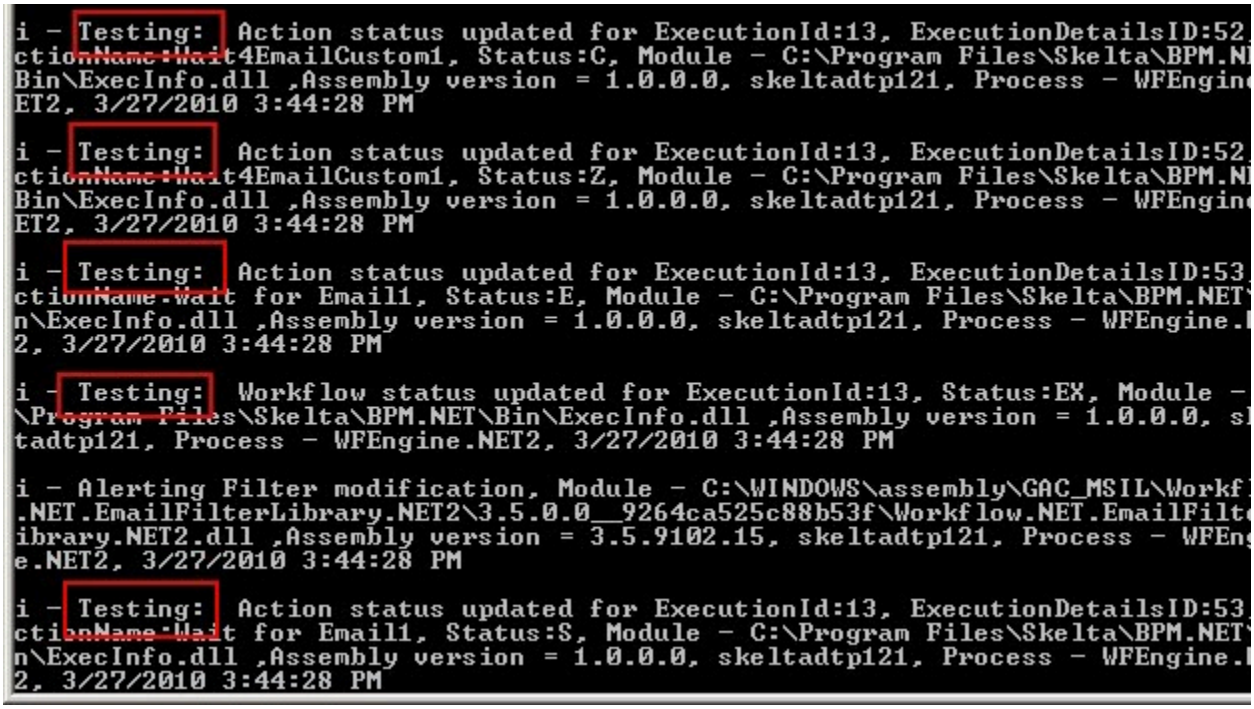
Compile the class and add the dll in to the AVEVA Work Tasks\bin folder. Now make an entry in the SKAddInProviders table in the Repository database.

Run the following SQL Command, after providing the specific values for ClassName and Assembly on the repository database.

```
INSERT INTO SKAddInProviders (Type,[Name],ClassName,[Assembly],Settings,IsGacAssembly)
VALUES('WorkflowInformation','WorkflowInformation','ClassName','bin\<DLLName>.dll','',1)
```

Note: Ensure the Name column of the table is unique. The Type and [Name] must be 'WorkflowInformation'.

To test the working of the implementation, test run a workflow from the specific repository with the logger kept open. You can see that the custom message getting displayed in the Logger Console.



Get User Name with Domain Name

In scenarios where you want to get the user name in the format *<domain name>\<user name>,* you can use the below options based on the requirement.

- For User interface, use `UserContext.GetUserName`.
- To use inside an activity, use `Context.DownLevelLogOnName`. This method can be used in expressions based on the requirement and criteria.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

DownLevelLogOnName is for getting the user name with domain name in run time. The following code sample demonstrates how to use DownLevelLogOnName in action run method. For example, adding a new variable in the variable collection with DownLevelLogOnName and assigning the value at runtime.

```
{
try
{
string customActivityPropertyValue = "";
VariablesCollection variables = CurrentContext.Variables;
if (variables.Contains("DownLevelLogOnName"))
{
Workflow.NET.Variable Var = variables["DownLevelLogOnName"];
Var.Value = CurrentContext.DownLevelLogOnName.ToString();
}
else
{
variables.Add("DownLevelLogOnName", "string",
CurrentContext.DownLevelLogOnName.ToString());
}
CurrentContext.log.LogInformation("Testing DownLevelLogOnName value :" +
CurrentContext.DownLevelLogOnName.ToString());
Output = "Success";
}
catch
{
Output = "UnSuccess";
}
return ActionResult.Completed;
}
```

Best Practices

Workflows:

1. Do not call the same workflow in a loop.
2. Before importing the workflow that has a child workflow, ensure to change or remove the repository name (application) of the child workflow from the process definition XML, if the repository does not exist in the importing environment.

SOA:

Two SOA Assemblies cannot have the same assembly name or class name.

While attaching a backed up Repository DataSource, ensure that the source SOA assembly registration dlls are registered in the target computer. Any customizations in the source SOA assembly must be copied to the target computer before publishing the workflow.

Dashboard:

1. In the dashboard, we recommend to use a Queue web part in the right zone.
2. In the dashboard, if multiple Inbox web parts are used, we recommend to set the **ReadingPaneinPopupWindow** to *true*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. In Mozilla Firefox, dashboard does not support web part drag and drop feature. Instead, use web part editor to move the web parts.
4. Do not to use the same list with different views in the dashboard.
5. We recommend the following configurations when multiple list web parts are displayed in dashboard:
   a. Disable **Show Header** and **Show Ribbon** bar properties.
   b. Use unique list view names across list web parts.
6. To apply the list view, it is mandatory to enable the **ShowListViews** property for the list web part.
7. List Views are recommended only for lists that are created with existing tables and not with generic tables.
8. For better performance of list view, it is recommended not to use the same tables for multiple list tables in the list.
9. To avoid multiple listing of the same items in list web part, it is recommended not to select the child table columns as grid columns for the list view.
10. Avoid using special characters while creating list folders.
11. For the Expression Editor to work in Firefox, pressing the space bar or any other key as a first step.

Web Parts:

1. Web part title should not be more than 200 characters.
2. Security Group Mapping in List Control Web Part should be used at List level and ListItem level.
3. Mandatory configurations for Adding Multiple List Control Web Part in Dashboard are as follows:
   a. Disable the properties Show Header and Show Ribbon Bar.
   b. Use unique list view names across all the List Control Web Part.
   c. Create list views with All items enabled.
   d. If you create similar lists in multiple list control web parts placed in a single homepage, you might not be able to view the list views properly.
4. We recommend to refresh the grid manually after creating items or folders in a grid.
5. You can specify height and width in percentage or in pixels for all the web parts except the Queue Control web part. However, for the Forms List web part, you can specify the height and width in pixels alone.
6. In a dashboard opened in Mozilla Firefox, you can move the web parts across the zones using the Web Editor.
7. We recommend to use the following Web Parts only once in a homepage:
   a. Calendar controls
   b. Queue

Database Tips

- In order to support database with a different culture or case-sensitive collation, make sure the database queries are written in case-sensitive format.

  For example:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

A query build against the above table should be:

Correct:

```
SELECT Id FROM SKAddInProviders
```

Wrong:

```
SELECT id FROM SKAddInProviders [Id column contains small letter i]
SELECT Id FROM SKAddinProviders [Table name contains small letter i]
```

The same applies for readers and rows in DataTable.

```
reader["id"] is wrong
reader ["Id"] is correct
```

Coding Tips

Close the Client Object

1. The below code is used for executing a workflow.

```
string xmlcontent = "<Expense><Amount>2000</Amount></Expense>";
Client objClient = new Client("MyApps","MyWF");
try
{
int ExecutionID = objClient.Execute(Convert.ToInt32(Session["UserID"]), xmlcontent);
 }
 catch
 {
//.....
}
 finally
 {
 if (objClient != null)
{
objClient.Close();
 }
}
```

It is mandatory to call the Close () method after the execute () method.

Dispose the Context object to avoid memory leaks

The Workflow.NET.Engine.Context class encapsulates all the information about the executing workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Usage of this class within other objects during workflow execution is required to access the executing workflow details. The Context object uses the Dispose (implementation of IDisposable interface) method to free up all the unmanaged resources. It is mandatory to call the Dispose method after usage of Context Object.

The sample codes shown below describes different scenarios for usage of Context object and the best practice of ensuring Disposal of context object by calling Dispose method. In some scenarios, calling Dispose on a different object takes care of Disposal of Context object used within that object as well.

The sample code below is used for disposing the context object:

```
Workflow.NET.Engine.Context workflowContext = Workflow.NET.Engine.Context.GetContext(17,
"Spill Repo");
using (workflowContext)
{
Workflow.NET.VariablesCollection workflowVariables = workflowContext.Variables;
}
//Context gets disposed automatically since we use using block
or
Workflow.NET.Engine.Context workflowContext = Workflow.NET.Engine.Context.GetContext(17,
"Spill Repo");
try
{
Workflow.NET.VariablesCollection workflowVariables = workflowContext.Variables;
//Do something
}
catch (Exception exception)
{
//Handle Exception.
}
finally
{
workflowContext.Dispose();
}
//Context is disposed explicitly
```

Mandatory to call Dispose of HWSActivity object

In HWSActivity object the Dispose method has to be called mandatorily as follows:

```
ApplicationObject appObject = new ApplicationObject("Telematics");
WorkflowObject workflowObject = new WorkflowObject("LeaveWF", appObject);
// Creating HWSActivity Object
HWSActivity hwsActivity = new HWSActivity(workflowObject, new
Guid("49e411e3-7487-4ad3-a4c7-40d88f5f75cf"));
try
{
String actionName = hwsActivity.CurrentActivity.ActionDisplayName;
//do something
}
catch (Exception exception)
{
//Handle exception
Console.WriteLine("Error:" + exception.Message);
}
finally
{
hwsActivity.Dispose();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
 // HWSActivity object is disposed explicitly.
```

The Work Item object (Skelta.HWS.WorkItem) also uses the context object which has to be released; now we have the Dispose method exposed in the Work Item object.

Mandatory to call Dispose of Work Item object

AVEVA Work Tasks Work Item List control is used to display all the activities for a user and from which the user can perform the tasks. User can also use the Work Item control in their custom application and load their Work Items. The Work Item object is largely used while building the Work Item detailed view, submitting or processing the Work Item etc.

Ensure that the Dispose method is called in the following format while using Work Item object in your code

```
ApplicationObject appObject = new ApplicationObject("Telematics");
WorkflowObject workflowObject = new WorkflowObject("LeaveWF", appObject);
//Creating Workitem object
WorkItem workItem = new WorkItem(workflowObject, new
Guid("dca38d4d-0c5d-4e3c-af98-77c5f2990709"));
try
{
Console.WriteLine("HWSActivity Name: " + workItem.HWSActivity.ActivityName);
//Do something
}
catch (Exception exception)
{
//Handle Exception
}
finally
{
workitem.Dispose();
}
//WorkItem object is disposed explicitly.
```

References to the WorkItem object through the WebWorkitemAdapter class may be present, where the Dispose method has been exposed.

Mandatory to call Dispose of WebWorkitemAdapter object

While using the WebWorkitemAdapter object (Skelta.HWS.WorkListChannel.Web. WebWorkitemAdapter), please ensure that the following format is followed to dispose the adapter object:

```
public WebWorkItemAdapter adapter;
try
{
adapter = new WebWorkItemAdapter();
adapter.ProcessWebChannelRequest();
_workitemSubject = adapter.CurrentWorkItem.Subject;
//Do something
}
catch(Exception exception)
{
//Handle Exception
}
finally
{
adapter.Dispose();
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: If both HWSActivity and the Work Item object are used, disposing the HWSActivity object will dispose the current context object.

For example:

```
ApplicationObject appObject = new ApplicationObject("Telematics");
WorkflowObject workflowObject = new WorkflowObject("LeaveWF", appObject);
HWSActivity hwsActivity = new HWSActivity(workflowObject, new
Guid("49e411e3-7487-4ad3-a4c7-40d88f5f75cf"));
Skelta.HWS.WorkItem workItem = new Skelta.HWS.WorkItem(hwsActivity);
try
{
workItem.Status = "Approved";
workItem.Save();
workItem.Submit("WebWorkList", "Approved", "","Approved @"+ DateTime.Now);
}
catch(Exception exception)
{
//Handle Exception
}
finally
{
hwsActivity.Dispose();
}
```

2. The below changes are required for culture compatibility

```
"".ToUpper() or "".ToLower() should not be used. Use "".ToUpperInvariant() or
"".ToLowerInvariant() for string comparison.
```

Wrong :

```
string str = "windows";
  if(str.ToUpper() == "WINDOWS"){ }
```

Correct:

```
string str = "windows";
  if(str.ToUpperInvariant() == "WINDOWS"){ }
```

3. Script Activity should not be used for complicated tasks. Performance issues can occur if external DLLs are loaded and used. We recommend using custom activity in place of script activity, as there might be some memory related issues using script activity whenever a reference to an external dll is required.
4. Only use the common methods defined in common functions to get the transaction scope.
   a. Workflow.NET.CommonFunctions.GetNewTransactionScope() [Joins a parent transaction if available, else executes in a new transaction].
   b. Workflow.NET.CommonFunctions.GetRequiredNewTransactionScope() [Starts a new transaction]
   c. Workflow.NET.CommonFunctions.GetSuppressedTransactionScope() [Suppresses the transaction for execution within the scope. Parent transaction is till valid.]
   d. Skelta.Core.DS.SingleTableObject.GetTransactionScope()[Joins a transaction if available, else suppresses the transaction. Combination of GetNewTransactionScope && GetSuppressedTransactionScope]
5. To implement an impersonated database connection, implement the DB connection pooling at the AVEVA Work Tasks Level. Due to this change, you are required to follow a few standards while using data handlers inside transaction scope in a custom activity.

Keep the following points in mind:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Connection has to be disposed of in the transaction after use.
2. Make sure the data reader is closed after use.
3. Always complete the transaction.

Wrong:

Data handler is disposed of outside the transaction. A transaction which is initialized in the transaction should be disposed within the transaction scope.

```
try
{
   using (TransactionScope TScope = Skelta.Core.DS.SingleTableObject.GetTransactionScope())
   {
     DBHandler = Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(new
Workflow.NET.Config(this.ApplicationName));
     IDataParameter idp1 = DBHandler.GetParameter("@PreferredChannelName",
preferredChannelName);
     IDataParameter idp2 = DBHandler.GetParameter("@Id", this.Id);
     IDataParameter idp3 = DBHandler.GetParameter("@LastUpdatedDateTime",DateTime.UtcNow);
     DBHandler.ExecuteUpdate(sql, idp1, idp2, idp3);
     TScope.Complete();
    }
}
catch (Exception e)
{
    throw e;
}
finally
{
if (DBHandler != null)
DBHandler.Dispose();
}
```

Correct:

```
try
 {
  using (TransactionScope TScope = Skelta.Core.DS.SingleTableObject.GetTransactionScope())
  {
     DBHandler = Workflow.NET.Storage.DataHandlerFactory.GetDataHandler(new
Workflow.NET.Config(this.ApplicationName));
     using (DBHandler)
     {
     IDataParameter idp1 = DBHandler.GetParameter("@PreferredChannelName",
preferredChannelName);
     IDataParameter idp2 = DBHandler.GetParameter("@Id", this.Id);
     IDataParameter idp3 = DBHandler.GetParameter("@LastUpdatedDateTime", DateTime.UtcNow);
     DBHandler.ExecuteUpdate(sql, idp1, idp2, idp3);
     }
     TScope.Complete();
     }
   }
   catch (Exception e)
   {
      throw e;
   }
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Performance Tips

Recommended Steps for Optimum Performance

1. AVEVA Work Tasks dlls configured in the SKAddInProviders table and AVEVA Work Tasks table schemas can be cached for better performance.
   - The code given below needs to be put in the application on start of the web application.

     ```
     Skelta.Core.EnableCaching.Initialize(new
      Skelta.Core.ApplicationObject("ApplicationName"), null);
     ```
   - If new dlls are added/existing dlls are changed and if they need to be re-cached the below code can be used.

     ```
     Skelta.Core.EnableCaching.Reset(new Skelta.Core.ApplicationObject("ApplicationName"),
     null);
     ```
   - For resetting the cache in engine, use the code given below. By default when the engine starts all the dlls and table schemas are cached.

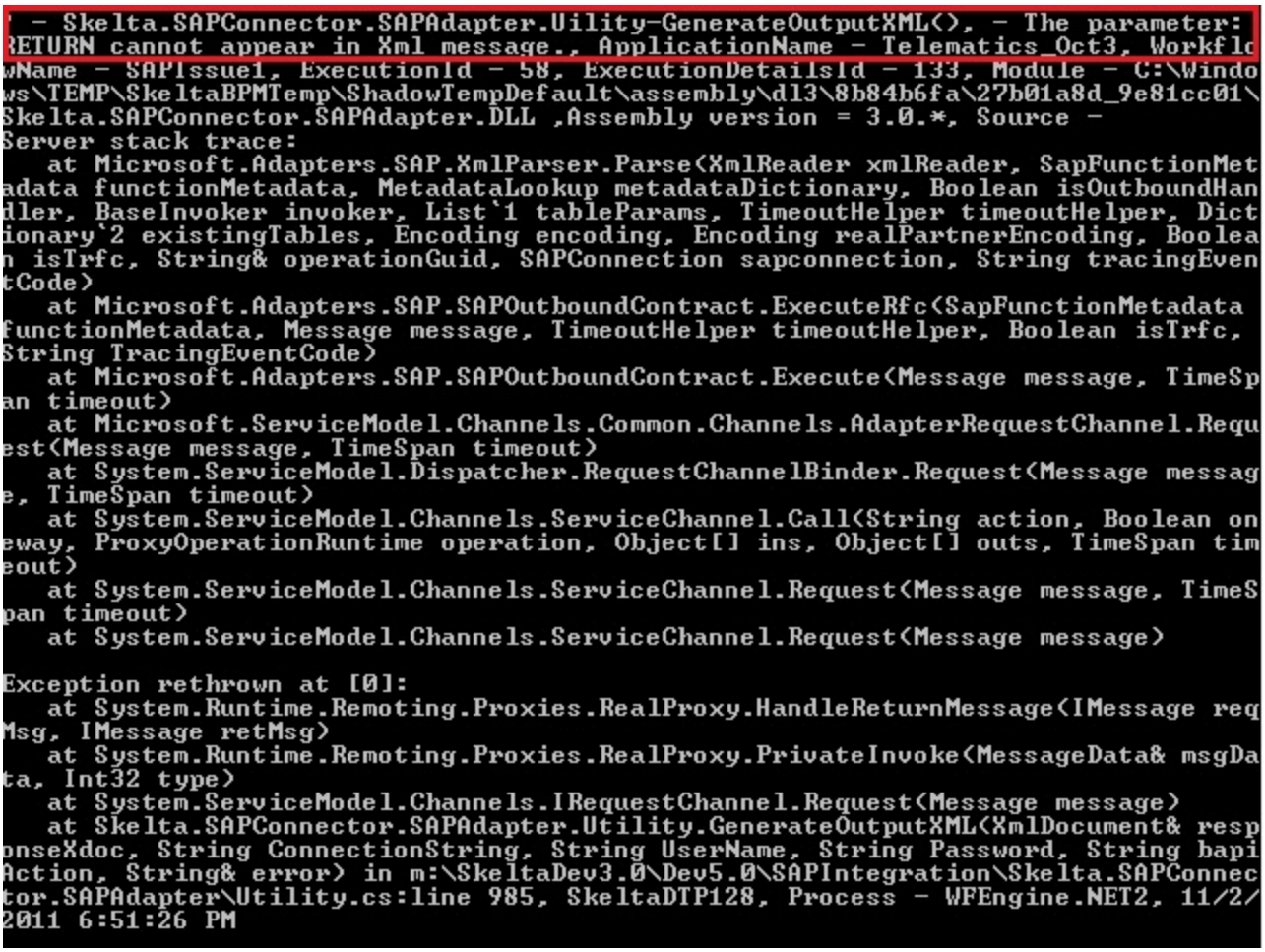
     ```
     Workflow.NET.Engine.Client client = new
     Workflow.NET.Engine.Client("ApplicationName","WorkflowName");
     client.ResetCache();client.Close();
     ```
2. You can set the threads that can be used by AVEVA Work Tasks in the farm configuration UI.
   - AVEVA Work Tasks Engine properties have the option to set Thread Pool Size.
3. You can ensure reliable performance even under heavy loads.
   - To ensure reliable performance even under heavy loads make sure **Connection Pooling** is enabled. By default, pooling is enabled and Max Pool Size is **300**.
4. Go for fail over clustering if the loads expected are high. Use different servers for running the AVEVA Work Tasks Service and Databases.

Troubleshoot Issues

1. While executing the BAPI_RFC Execute activity, the logger might log the following error message:

The parameter: RETURN cannot appear in XML message..... as shown in the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



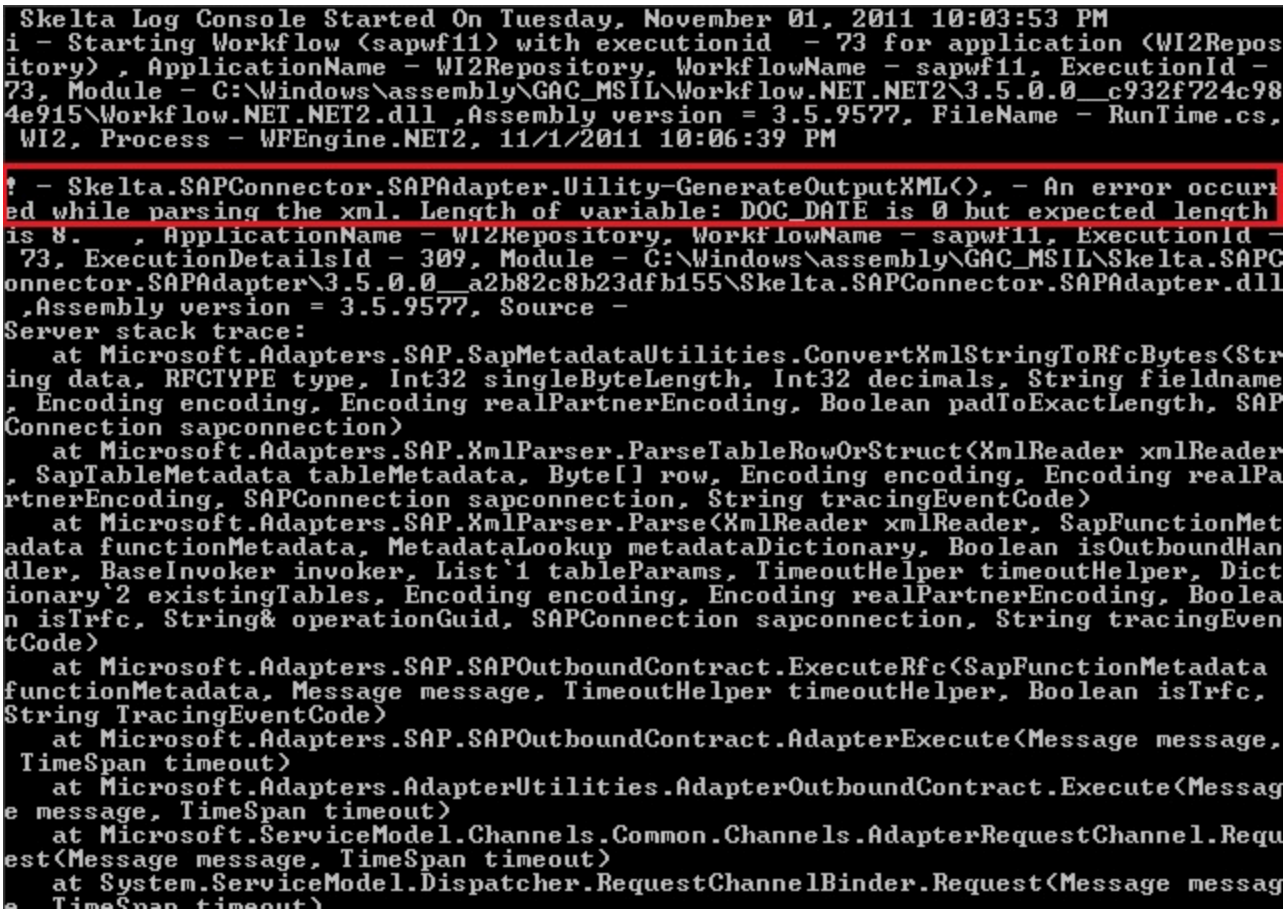
Solution

- Go to **BAPIRFC Execute** activity and select the **No** option in the **Include Return Parameter** property.

3. The following error message appears:

An error occurred while parsing the xml. Length of variable: DOC_DATE is 0 but expected length is 8.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The Date Time type parameter must always be provided with the required values.

Solution:

You can either provide the required value to all the Date Time type parameters or delete the nodes from the Input XML Variable present in the Start activity. To delete the node, right-click the node, delete it, and save the variable.

4. Unable to send or receive emails and SMS using the AVEVA Work Tasks Communication Service.

Possible Cause

The Communication Service is hosted on Port 25. If any other application or process uses the same port, the port will be unavailable to the Communication service. Hence, you will not be able to send and receive emails and SMS.

Solution

After you restart the **Communication Service**, we recommend you to use Telnet to verify the availability of port 25. This helps you to ensure that the communication service is started and running. However, ensure to maintain a time delay of a minute between the communication service restart and Telnet.

Use the following command to verify the availability of port 25:

```
Telnet <Computer Name> 25
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. When upgrading a large database, the following error appears: The transaction associated with the current connection has completed, but has not been disposed. The transaction must be disposed before the connection can be used to run SQL statements.

Possible Cause

Upgrading a large database takes more than 10 minutes, due to the size of the database, but the default transaction timeout settings is 10 minutes.

Solution

Increase the transaction timeout settings in the **machine.config** file as follows:

```
<system.transactions>
<machineSettings maxTimeout="01:00:00"/>
</system.transactions>
```

6. Start Activity slows down and takes time to start when the workflow is executed.

Possible Cause

The invalid web service was registered in the Repository.

Solution

Make the necessary changes in the web service URL and ensure that the web service URL is valid.

Limitations

- Messages or notifications in **AVEVA Work Tasks**, which are returned by third party controls such as, ISWebGrid, Telerik, or Kendo does not comply with the Message bar usage and guidelines.
- When Email Notification is configured for an **Invoke Form Activity**, the form opens through the Email. However, when the form is submitted successfully, the confirmation message does not appear.
- When you receive an email notification configured for an **Invoke Form Activity**, and click the link in the email, the form opens in a window. If Google Chrome is the default browser, then the window closes as expected.
- In **Invoke Form Activity**, *Multi-select* and *Close* on *Finish* property are not supported in Forms.
- The user interface messages and error messages for Publish, UnPublish, and Publish and Test run in workflows do not conform to the Message bar standards.
- While packaging, workflows containing the following combination of Property and Activity does not work as expected:
  - Exception Template Property (Affected activity : Start Activity)
- You can associate multiple events to a single workflow. However, the event provider type must be the same.
- Although localization is supported, some of the special characters such as, ' " # & / & [\ ] | ~ + < > are not supported for Workflow names and List names. We recommend that these special characters are not used in naming the activities.
- When a workflow with Message Queue activities is installed in the target environment and if its associated artifact (Message Queue Connection) does not get installed, then the workflow is installed in Draft status. If you manually publish this workflow, it will return unexpected results.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Forms and Lists User Guide

This section is intended for the users to create user interfaces and entities respectively for business process management applications. Forms and lists form the foundation of a business application and integrate with workflows for managing the business processes.

Features available in Forms and Lists

Forms

Designing the user interface is one of the most important aspects of developing a business application. Use web-based forms to easily create complex user interfaces with responsive design and cross-browser compatibility. Forms integrate with lists and workflows to enable rapid deployment of business applications.

- Create rich browser-based forms and use folders to organize forms.
- Import and export forms as XML documents and save specific versions of forms to keep track of changes.
- Design intricate forms using the simple drag-and-drop forms designer with advanced set of controls, and use layouts to organize controls on forms.
- Preview and publish forms across platforms and devices with a native look and feel for mobile devices.
- Secure forms by defining rights assigned to forms or set of forms grouped under folders.
- Generate dynamic list forms and associate forms to workflows to trigger workflows and pass data.
- Populate options through scripts for Check Box, Drop-Down, Radio Button, and List controls.
- Add validation checks and interactivity to forms using the dedicated script editor with client-side JavaScript.
- Achieve fast and fluid application performance with optimized client-side code execution.

For more information about forms, see Forms.

Lists

Entities are the building blocks of business applications. Use lists to easily create complex business entities. Lists are mapped to single or multiple SQL tables. Lists also integrate with forms and workflows to enable rapid deployment of business applications.

- Create rich browser-based lists and use folders to organize lists.
- Import and export lists as XML documents.
- Design functional lists using the simple drag-and-drop lists designer with basic set of controls.
- Secure lists by defining rights assigned to lists or set of lists grouped under folders.
- Generate dynamic list forms and associate lists to workflows to trigger workflows and pass data.

For more information about lists, see Lists.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

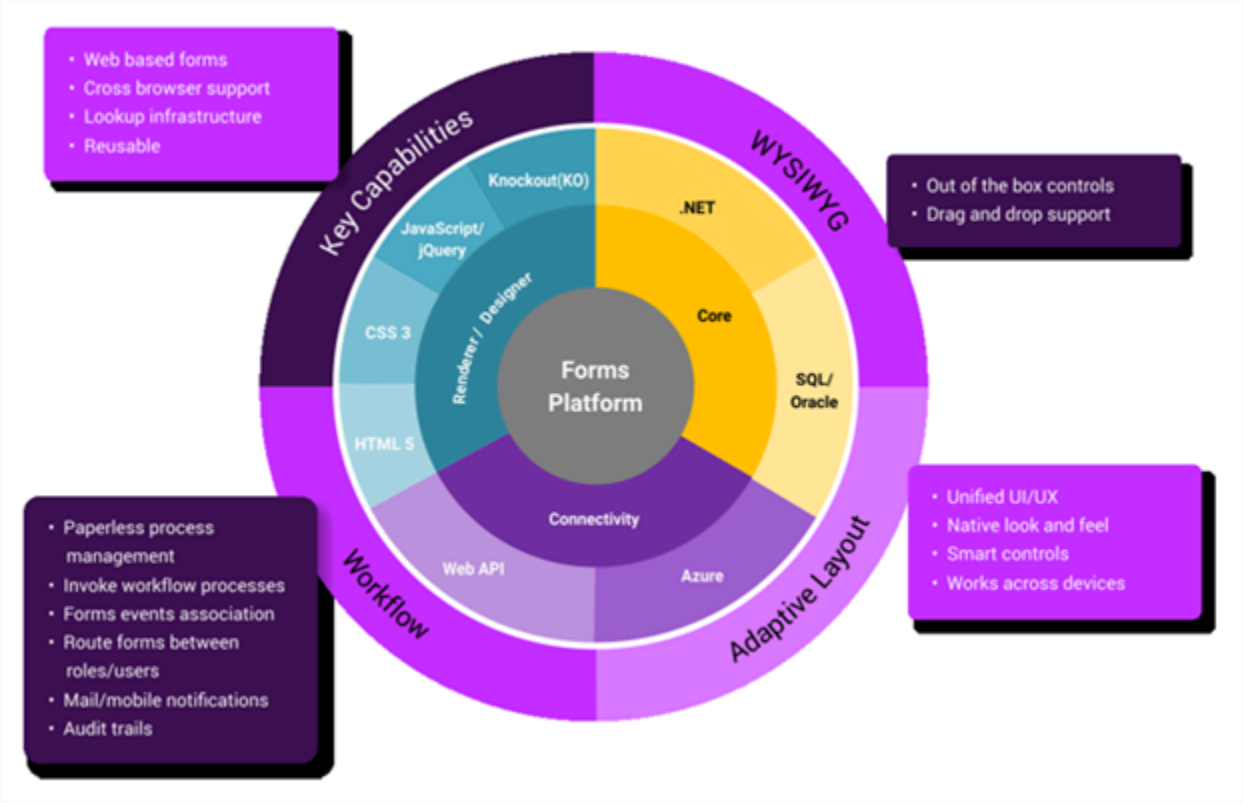
Forms

This section provides information on getting started with Forms and how to access the Forms.

Introduction

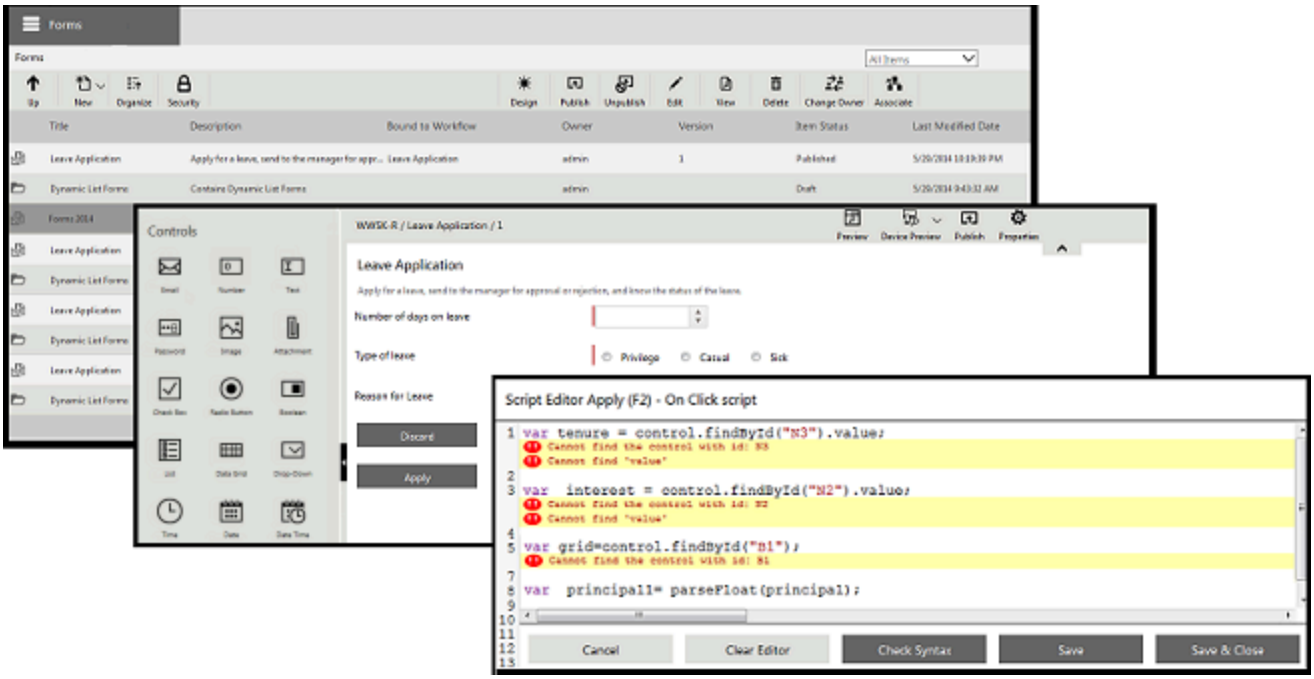
AVEVA Work Tasks Forms is a powerful platform to design and use sophisticated forms in your enterprise. The forms platform uses the latest .NET framework, and is developed using HTML5, CSS3, and Knockout JavaScript for a rich and responsive user experience. You can connect to forms with various Web APIs and through the Azure cloud platform.

Forms Platform



The forms platform offers a "what you see is what you get (WYSIWYG)" experience with an adaptive layout and embeddable support for workflow enhancements.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Forms, Forms Designer, and Script Editor

Key Capabilities	Workflow Enhancements
Web based forms • Cross-browser support • Lookup infrastructure • Reusable	• Paperless process management • Invoke workflow processes • Forms events association • Route forms between roles and between users • Mail and mobile notifications • Audit trails

Form preview on a mobile device

WYSIWYG Experience	Adaptive Layout	Embeddable Support
• Out of the box controls • Drag and drop support	• Unified user interface and user experience • Native look and feel • Smart controls • Works across devices	• Custom Control Infrastructure • Single page – Model View ViewModel (MVVM)

Get Started with Forms

Create, design, preview, and publish a simple form to use with workflows.

Follow the steps to create an employee registration form to enter and submit employee information such as

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

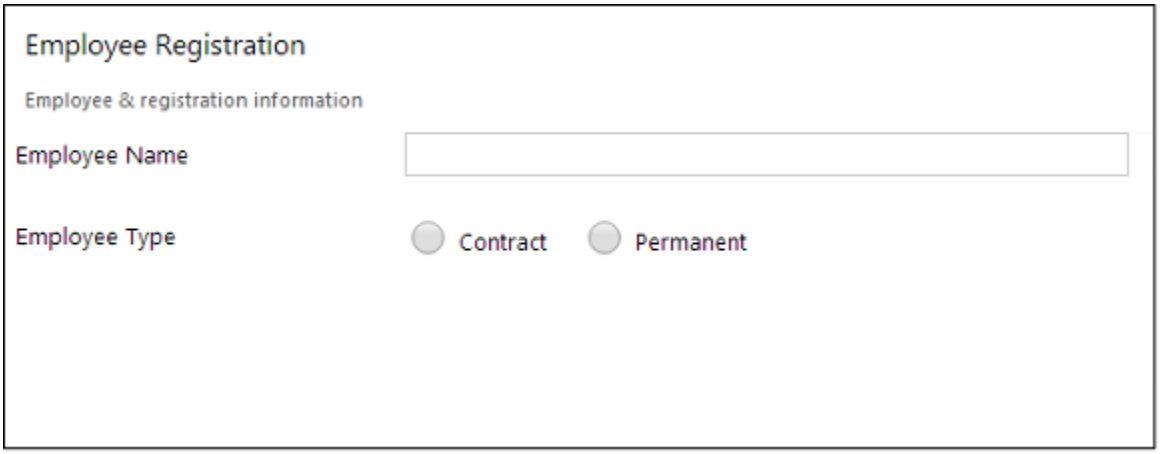
name and type.

To get started with your first form

Creating Form

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. Click **New**, and then click **Form**. The **New Form** wizard appears.
3. In the **Name** box, type **Employee Registration**, and in the **Description** box, type **Employee & registration information**.
4. Click **Save & Continue**. The **Employee Registration** form is created, and the **Form Options** wizard appears.

Designing Form



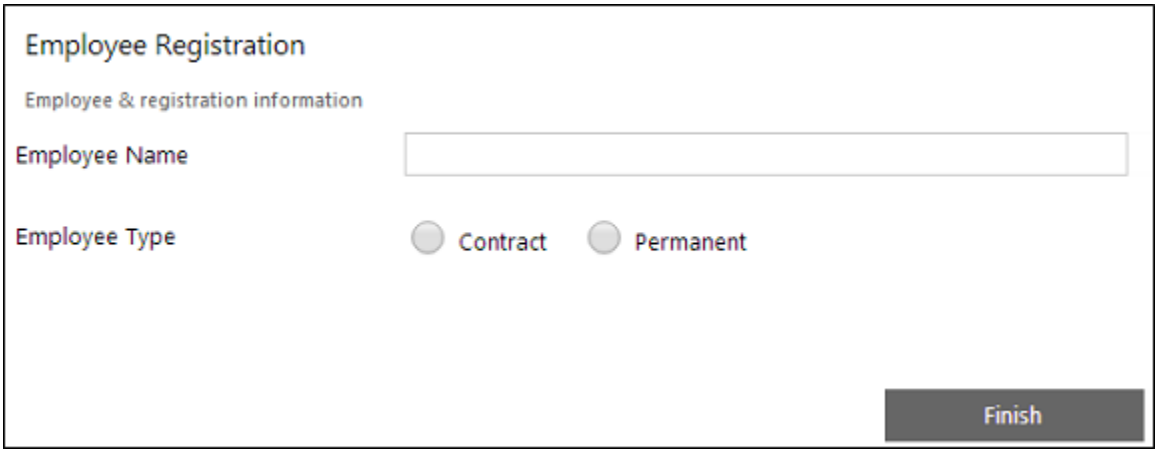
Employee Registration Form on Forms Designer Canvas

1. Click **Design this form**.
2. Click **Finish**. The **Forms Designer** page appears.
3. Drag **Text** control from the toolbox to the canvas. The **Text** control appears on the canvas.
4. Point to the **Text** control on the canvas, click **Textinput1**, and then type **Employee Name**.
5. Drag **Radio Button** control from the toolbox to the canvas. The **Radio Button** control appears on the canvas.
6. Point to the **Radio Button** control on the canvas, click **Radiobuttongroup1**, and then type **Employee Type**.
7. Point to the **Radio Button** control on the canvas. The **Shortcut** menu appears.
8. Click the **Properties** button. The **Properties** dialog box appears.
9. Under **Options**, Click **Add Record**. A new record appears.
10. In the **Text** box, type **Contract**, and in the **Value** box, type **1**.
11. Again, under **Options**, Click **Add Record**. A new record appears.
12. In the **Text** box, type **Permanent**, and in the **Value** box, type **2**.
13. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Previewing Form

Employee Registration Form preview on browser:



Publishing Form

1. On the action bar, click **Publish**. The form is published and an information message appears.
2. Click **OK**. The information message is closed.

Closing Forms Designer

Click the **Close** button. The **Forms Designer** page is closed.

Note:
- After you create the **Employee Registration** form, the form will be available in the **Forms** page to perform further operations.

Access Forms

Use the **Forms** page to create, organize, import, save, design, publish, manage, secure, and associate forms to workflows.

To access Forms

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.

   The **Forms** page shows the title, description, bound to workflow, owner, version, item status, and last modified date for the forms.

Forms Interface

Forms interface has breadcrumbs, an action bar, and a list of forms.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Breadcrumbs

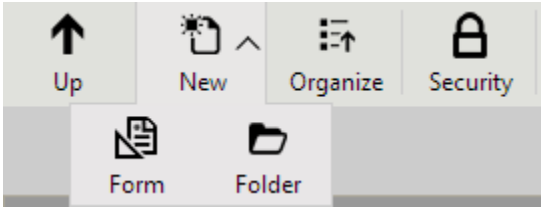
Breadcrumbs display the path to the child form or folder, and has commands to customize and view the list of forms.

Command	Description
All Items	Default view to display the list of forms.
Create View	Create a new view to display the list of forms.

Action Tool Bar

Action tool bar has commands to manage forms and folders. Standard commands are available all the time whereas contextual commands are available only when a form is selected.

Standard Commands

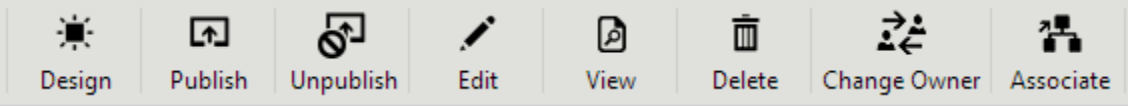


Menu/Command	Command	Description
Up		Go back to the parent form or folder.
New	Form	Create a new form. For more information, see Create Forms.
	Folder	Create a new folder to group and organize forms. For more information, see Create Folders for Forms.
Organize		Organize the form by moving the form to a folder or by making the form a child item of another form. For more information, see Organize Forms.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Security		View or customize the security level for the forms. For more information, see Set Security of Forms.

Contextual Commands

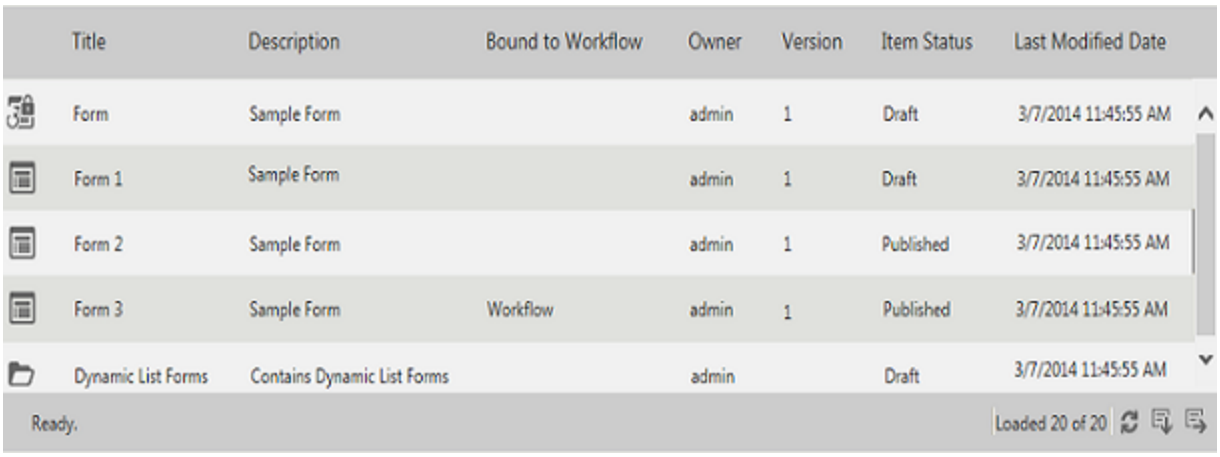


Command	Description
Design	Design the form with the forms designer. For more information, see Design Forms.
Publish	Publish the form or the folder. For more information, see Publish Forms or Publish Folders.
Unpublish	Unpublish the form or the folder. For more information, see Unpublish Forms or Unpublish Folders.
Edit	Edit description of the form and then change design of the form with the forms designer. For more information, see Edit Forms.
View	Preview the form. For more information, see Preview Forms.
Delete	Delete the form. For more information, see Delete Forms.
Change Owner	Change owner of the form. For more information, see Change Ownership of Forms.
Associate	Associate the form to a workflow. For more information, see Associate Forms to Workflows.

List

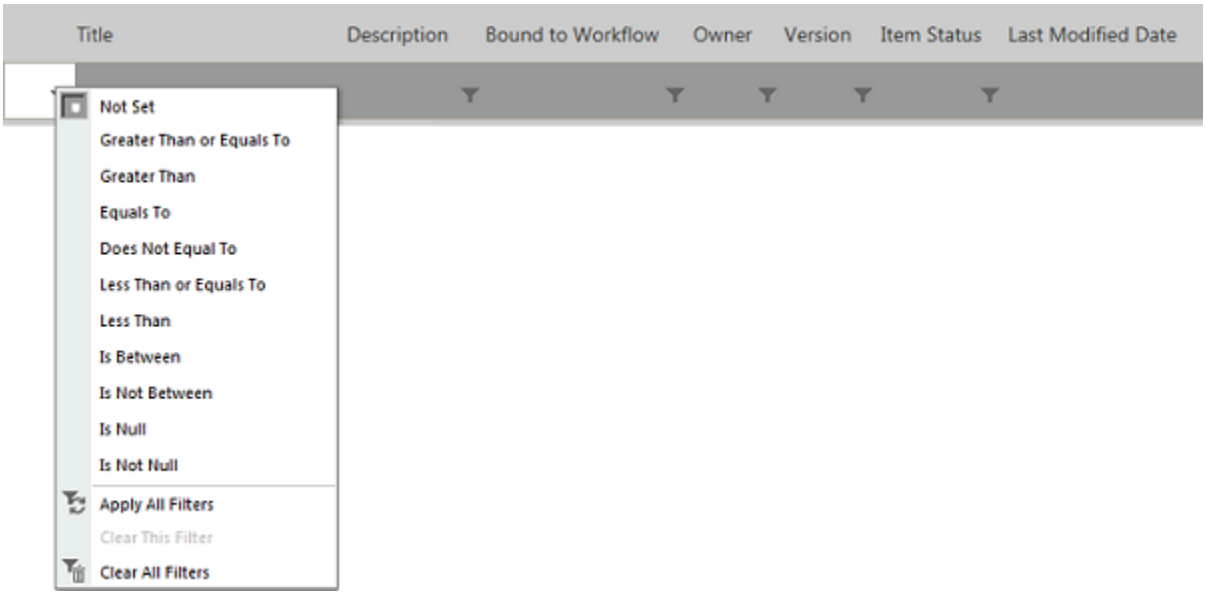
List of forms shows the title, description, bound to workflow, owner, version, item status, and last modified date for the forms. Click on a column to sort the forms.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Filter Bar

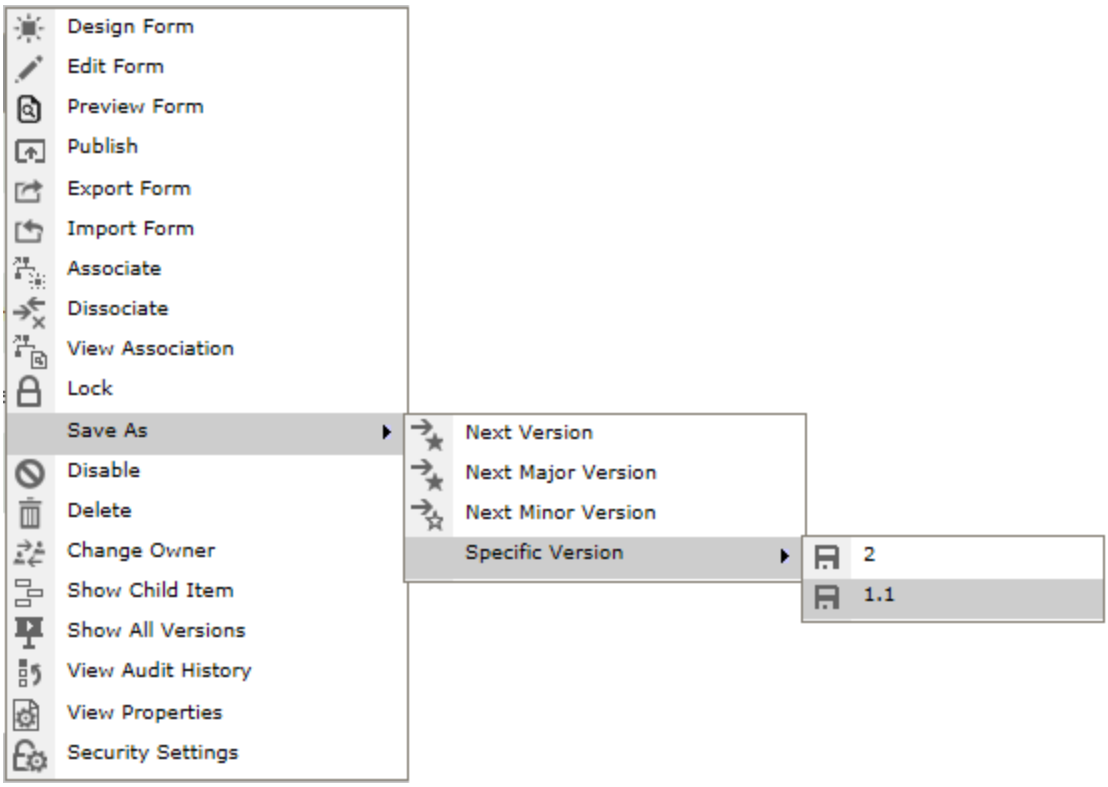
Filter bar can help to quickly find the desired forms from the list of forms. Filter bar has a funnel button and a parameter area for every column of the forms list. The filter button shows the available filter options and the parameter area is for entering the filter criteria. The filter bar is hidden by default. Right-click on the column header, and then click **Filter Bar**, to access the filter bar. For more information about filter bar, see Filter Forms.



List Shortcut Menu

List shortcut menu has commands to manage the form. The list of commands available is dynamic and changes when the form is locked or disabled. Right-click on a form to access the shortcut menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Command	Description
Design Form	Design the form with the forms designer. For more information, see Design Forms.
Edit Form	Edit description of the form and then change design of the form with the forms designer. For more information, see Edit Forms.
Preview Form	Preview the form. For more information, see Preview Forms.
Publish	Publish the form. For more information, see Publish Forms or Publish Folders.
Unpublish	Unpublish the form. For more information, see Unpublish Forms or Unpublish Folders.
Export Form	Export the form as an XML document. For more information, see Export Forms.
Import Form	Import the form from an XML document. For more information, see Import Forms.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Command	Description
Associate	Associate the form to a workflow. For more information, see Associate Forms to Workflows.
Dissociate	Remove association of the form from a workflow. For more information, see Dissociate Forms from Workflows.
View Association	View association of the form with a workflow. For more information, see View Association of Forms with Workflows.
Lock	Lock the form. For more information, see Lock Forms.
Unlock	Unlock the form. For more information, see Unlock Forms.
Save As	Save a different version of the form. For more information, see Saving Versions of Forms.
Disable	Disable the form. For more information, see Disable Forms.
Enable	Enable the form. For more information, see Enable Forms.
Delete	Delete the form. For more information, see Delete Forms.
Change Owner	Change owner of the form. For more information, see Change Ownership of Forms.
Show Child Item	Show child items of the form. For more information, see View Child Items of Forms.
Show All Versions	Show all versions of the form. For more information, see View Versions of Forms.
View Audit History	View audit history of the form. For more information, see View Audit History of Forms.
View Properties	View properties of the form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Command	Description
	For more information, see View Properties of Forms.
Security Settings	View or customize the security level for the forms. For more information, see Set Security of Forms.

Create and Organize Forms

For information on creating and organizing the forms, see the topics:

- Create Forms
- Create Folders for Forms
- View Child Items of Forms
- Organize Forms

Create Forms

Create a form to use with workflows.

To create a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. On the action bar, click **New**, and then click **Form**. The **New Form** wizard appears.
3. In the **Name** box, type a name for the form.

   Note: Name cannot contain special characters such as # ~ & + \ " ' < > | @ ! $ % * ( ) = / { } ?
4. In the **Description** box, type a description for the form.

   Note: Description cannot contain special characters such as # ~ & + \ " ' < > | @ ! $ % * ( ) = / { } ?
5. Click **Save & Continue**. The form is created, and the **Form Options** wizard appears.
6. Do one of the following:

Task	Action
• Design the form.	1. Click **Design this form**. 2. Click **Finish**. The **Forms Designer** page appears. 3. Design the form, and then click the **Close** button. The form is saved and the **Forms Designer** page is closed.
• Cancel further operations.	1. Click **No task to be performed now**. 2. Click **Finish**. The form is saved and an information

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
	message appears. 3. Click **OK** (or click the **Close** button). The information message and the **Form Options** wizard are closed.

Note: You can set the description only when you create or edit a new form.

Create Folders for Forms

Create a folder for a form to group and organize the form.

To create a folder for a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. On the action bar, click **New**, and then click **Folder**. The **New Folder** wizard appears.
3. In the **Title** box, type a name for the folder, and in the **Description** box, type a description for the folder.
4. Click **Save & Close**. The folder is created, and an information message appears.
5. Click **OK** (or click the **Close** button). The information message and the **New Folder** wizard are closed.

View Child Items of Forms

View child items of a form to see forms organized under the form.

To view child items of a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Show Child Item**. The child items page for the form appears.
   The child items page for the form shows the available child forms.

   Note: To go back to the **Forms** page, on the action bar, click **Up**.

Organize Forms

Organize a form by moving the form to a folder or by making the form a child item of another form.

To organize a form by moving the form to a folder

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. On the action bar, click **Organize**. The **Organize Forms** dialog box appears.
3. Drag a form to a folder. The form is moved to the folder.
4. Click **Close**. The **Organize Forms** dialog box is closed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To organize a form by making the form a child item of another form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. On the action bar, click **Organize**. The **Organize Forms** dialog box appears.
3. Drag a form to another form. The form is now a child item of another form.
4. Click **Close**. The **Organize Forms** dialog box is closed.

Import and Save Forms

You can import a form to create the form from an XML document as well as export a form to save the form as an XML document.

- Import Forms
- Export Forms
- View Versions of Forms

Import Forms

Import a form to create the form from an XML document.

The XML schema defines the controls and properties of the controls for the form.

To import a form

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Import Form**. The **Import** dialog box appears.
3. Click **Choose File** to select the Form XML file to be imported, and then click **Import**. The form is imported and an information message appears.
4. Click **Close**. The **Import** dialog box is closed.

To import a form in Forms Designer

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. On the action bar, click **Import**. The **Import** window appears.
4. Click **Browse**. The **Choose File to Upload** dialog box appears.
5. Select an XML document, and then click **Open**. The XML document is selected and the **Choose File to Upload** dialog box is closed.
6. Click **Import**. The form is imported and an information message appears.
7. Click **Close**. The **Import** window is closed.

Note:
- When a form is imported, the existing controls and properties of the controls for the form are overwritten with

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the controls and properties of the controls defined in the XML schema.
- You cannot import an XML schema defined for Forms.

Export Forms

Export a form to save the form as an XML document.

The XML schema defines the controls and properties of the controls for the form. After successfully exporting a form, you can import the XML document to create similar forms.

To export a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Export** Form. A message bar appears.
3. Click **OK**. The **Save As** dialog box appears.
4. Click **Save**. The list is exported as an XML document, and the **Save As** dialog box is closed.

To export a form in Forms Designer

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. On the action bar, click **Export**. The list is exported as an XML document.

View Versions of Forms

View versions of a form to see available versions of the form and other information for each version of the form.

To view versions of a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Show All Versions**. The versions page for the form appears.

   The versions page for the form shows the version, comments, status, archived status, creation date and time, and workflow for the associated form.

   Note: You can also design, edit properties, preview, delete, publish or unpublish, and change owner for a version of the form.

Design and Publish Forms

You can do the following while working with Forms:

- Design a form to add controls and containers to the form.
- Edit a form to change description and design of the form.
- Delete a form when the form is no longer required.
- Preview a form to check look and feel of the form at run-time and the XML generated for the form.
- Publish a form to use the form in a workflow.The form associated to a workflow is a Fill Form, and appears in

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the Fill Form page only when both the form and the workflow are published.

- Unpublish a form to prevent using the form in a workflow.
- Publish a folder to use the forms associated to workflows in the folder in the Fill Form page.
- Unpublish a folder to prevent using the forms associated to workflows in the folder in the Fill Form page.

Design Forms

Design a form to add controls and containers to the form.

To design a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. Design the form, and then click the **Close** button. The form is saved and the **Forms Designer** page is closed.

Edit Forms

Edit a form to change description and design of the form.

To edit a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Edit Form** (or click on a form, and then on the action bar, click **Edit**). The **Edit Form** dialog box appears.
3. In the **Description** box, edit description for the form.
4. Click **Save & Continue**.

   The description for the form is saved, and the **Edit Form** dialog box is closed.

   The **Forms Designer** page appears.
5. Design the form, and then click the **Close** button. The form is saved and the **Forms Designer** page is closed.

Note: You cannot edit the name of the form. To edit properties of a form, see Edit Properties of Forms.

Delete Forms

Delete a form when the form is no longer required.

To delete a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Delete** (or click on a form, and then on the action bar, click **Delete**). A message bar appears.
3. Click **OK**.

   The message bar is closed,and the form is deleted.

   A message bar appears.
4. Click **Close** (or click the **Close** button). The message bar is closed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: If you delete a form, then the form cannot be restored.

Preview Forms

Preview a form to check look and feel of the form at run-time and the XML generated for the form.

To preview only the form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Preview Form** (or click on a form, and then on the action bar, click **View**). The **Preview** dialog box appears.
3. Check the look and feel of the form.
4. Click the **Close** button. The **Preview** dialog box is closed.

To preview the form and the XML generated for the form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Preview Form** (or click on a form, and then click **View**). The **Preview** dialog box appears.
3. Check the look and feel of the form, and then click **Finish**. The **Instance XML** dialog box appears.
4. Check the generated XML, and then click the **Close** button. The **Instance XML** dialog box is closed.
5. Click the **Close** button. The **Preview** dialog box is closed.

Publish Forms

Publish a form to use the form in a workflow.The form associated to a workflow is a **Fill Form**, and appears in the **Fill Form** page only when both the form and the workflow are published.

To publish a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**.

   The **Forms** page appears.
2. Right-click on a form, and then click **Publish** (or click on a form, and then on the **Action** bar, click **Publish**).

   The form is published and a message bar appears.
3. Click **Close**.

   The message bar is closed.

Note: When you publish a form, the item status of the form changes from draft to published.

To publish multiple forms

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**.

   The **Forms** page appears.
2. Select multiple forms, and then on the **Action** bar, click **Publish**.

   The forms are published and a message bar appears.
3. Click **Close**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The message bar is closed.

Note: When you publish forms, the item status of the forms changes from draft to published.

Unpublish Forms

Unpublish a form to prevent using the form in a workflow.

To unpublish a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**.

   The **Forms** page appears.
2. Right-click on a form, and then click **Unpublish** (or click on a form, and then on the **Action** bar, click **Unpublish**).

   The form is unpublished and a message bar appears.
3. Click **Close**.

   The message bar is closed.

   Note: When you unpublish a form, the item status of the form changes from published to draft.

To unpublish multiple forms

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**.

   The **Forms** page appears.
2. Select multiple forms, and then on the **Action** bar, click **Unpublish**.

   The forms are unpublished and a message bar appears.
3. Click **Close**.

   The message bar is closed.

Note: When you unpublish forms, the item status of the forms changes from published to draft.

Publish Folders

Publish a folder to use the forms associated to workflows in the folder in the **Fill Form** page.

To publish a folder

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a folder, and then click **Publish** (or click on a folder, and then on the action bar, click **Publish**). The folder is published and an information message appears.
3. Click **Close** (or click the **Close** button). The information message is closed.

   Note:
   - When you publish a folder, the item status of the folder changes from draft to published.
   - Associated forms appear in the **Fill Form** page only when both the forms and the workflows are published.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Unpublish Folders

Unpublish a folder to prevent using the forms associated to workflows in the folder in the **Fill Form** page.

To unpublish a folder

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a folder, and then click **Unpublish** (or click on a folder, and then on the action bar, click **Unpublish**). The folder is unpublished and an information message appears.
3. Click **Close** (or click the **Close** button). The information message is closed.

Note: When you unpublish a folder, the item status of the folder changes from published to draft.

Manage and Secure Forms

To manage and secure forms, you can do the following:

- Lock Forms
- Unlock Forms
- Disable Forms
- Enable Forms
- Filter Forms
- Change Ownership of Forms
- Set Security of Forms

You can also view properties of a form to see versions and child items for the form.

View audit history of a form to see date, time and user information when the form was locked and unlocked, and when the form owner or version was changed.

Create a form link to add a form to the menu, and to set a form as the home page.

Lock Forms

Lock a form to prevent other users from editing the form.

To lock a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Lock**. The form is locked and a message bar appears.
3. Click the **Close** button. The message bar is closed.

Note: When a form is locked, a lock overlay icon appears on the icon of the form.

Unlock Forms

Unlock a form to allow other users to edit the form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To unlock a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a locked form, and then click **Unlock**. The form is unlocked and a message bar appears.
3. Click the **Close** button. The message bar is closed.

Note: When a form is unlocked, the lock overlay icon disappears from the icon of the form.

Disable Forms

Disable a form to prevent editing or using the form.

To disable a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The Forms page appears.
2. Right-click on a form, and then click **Disable**. The form is disabled and a message bar appears.
3. Click the **Close** button. The message bar is closed.

Shortcut Menu Commands

When a form is disabled, some commands on the shortcut menu are no longer available to prevent editing or using the form.

Available Commands	Unavailable Commands
• Preview Form • View Association • Lock (or Unlock if the form is locked) • Enable • Delete • Show Child Item • View Audit History • View Properties	• Design Form • Edit Form • Publish • Unpublish • Export Form • Import Form • Associate • Dissociate • Save As • Change Owner • Show All Versions • Security Settings • Add to Package Template

Enable Forms

Enable a form to allow editing or using the form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To enable a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a disabled form, and then click **Enable**. The form is enabled and a message bar appears.
3. Click the **Close** button. The message bar is closed.

Filter Forms

Filter forms to quickly find the forms you are looking for from the list of forms. Filtering forms is useful especially when there are a large number of forms.

To apply a filter for the forms

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click the column header, and then click **Filter Bar**. The filter bar appears.
3. Enter a value in the filter bar for a column, and then press Enter.
4. Click the **Funnel** button on the column, and then click a filter option. The filter option is selected.
5. Click **Apply All Filters**. The filtered result appears.

To clear a filter for the forms

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click the filter bar (or click the **Funnel** button on the column), and then click **Clear All Filters**. The filtered result is cleared.

Filter Options

Filter forms using one of the filter option for each column:

Option	Description
Not Set	Show forms without applying any filters. This is the default filter option.
Greater Than or Equal To	Show forms where the value of the column is greater than or equal to the specified value.
Greater Than	Show forms where the value of the column is greater than the specified value.
Equal To	Show forms where the value of the column is equal to the specified value.
Does Not Equal To	Show forms where the value of the column is not equal to the specified value.
Like	Show forms where the value of the column is like the specified value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Option	Description
Does Not Like	Show forms where the value of the column is not like the specified value.
Contains	Show forms where the value of the column contains the specified value.
Does Not Contain	Show forms where the value of the column does not contain the specified value.
Less Than or Equal To	Show forms where the value of the column is less than or equal to the specified value.
Less Than	Show forms where the value of the column is less than the specified value.
Is Between	Show forms where the value of the column is between the specified inclusive range of values.
Is Not Between	Show forms where the value of the column is not between the specified inclusive range of values.
Is Null	Show forms where the value of the column is null.
Is Not Null	Show forms where the value of the column is not null.
Is Empty	Show forms where the value of the column is an empty string.
Apply All Filters	Apply all filters set for the columns.
Clear This Filter	Clear all filters set for a column.
Clear All Filters	Clear all set filters.

Change Ownership of Forms

Change owner of a form to allow other user to own and control the form.

To change owner of a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Change Owner** (or click on a form, and then on the action bar, click **Change Owner**). The **Change Owner** dialog box appears.
3. Click the **User Lookup** button. The **User Lookup** dialog box appears.
4. Select a user, and then click **Update**.
5. Click **Save**. The owner of the form is changed and an information message appears.
6. Click **OK** (or click the **Close** button). The information message and the **Change Owner** dialog boxes are

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

closed.

Set Security of Forms

Secure a form by defining rights assigned to a form or a set of forms grouped under a folder.

A forms user or role inherits the rights assigned at a repository level. However, the administrator can customize the rights at a form level and also assign rights to groups and map users to groups.

Standard Settings

Use standard settings to view security set at the list level, if available, or at the repository level for:

- a role or user.
- the security group customization and resource mapping.

To view security set for a role or user

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. On the action bar, click **Security**. The **Security Settings** page appears.
3. In the **Standard Settings** tab, click **View Effective Security For User/Role**.
4. Do one of the following:

Task	Action
• View security set at the list level, if available, or at the repository level for a role.	a. In the **Role Based** tab, click the **Role Lookup** button. The **Role Lookup** dialog box appears. b. Select a role, and then click **Update**. c. Click **View Rights**. The effective rights for the role appear.
• View security set at the list level, if available, or at the repository level for a user.	a. Click the **User Based** tab. b. Click the **User Lookup** button. The **User Lookup** dialog box appears. c. Select a user, and then click **Update**. d. Click **View Rights**. The effective rights for the user appear.

To view security group customization and resource mapping

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Click **Security**. The **Security Settings** page appears.
3. In the **Standard Settings** tab, click **View Security Mapping** at **List Level**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click on a **Security Group**. The **Viewing Security Group Rights** dialog box appears.
5. Check the security group customization and resource mapping.
6. Click **Close** (or click the **Close** button). The **Viewing Security Group Rights** dialog box is closed.

Advanced Settings

Use advanced settings to edit the existing mapping, add a new resource mapping or delete the existing mapping by enabling mapping at List level.

To set or edit the security group customization and resource mapping

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Click **Security**, in the **Security group**. The **Security Settings** page appears.
3. Click the **Advanced** settings tab, and then do one or more of the following:

Task	Action
• Set the security group customization and resource mapping.	• Click **Enable List Specific Security**. **Note:** This option is available only when the security group customization and resource mapping has not been set or removed.
• Remove the security group customization and resource mapping.	• Click **Remove Security from this List**.
• Edit the security group customization.	• Click **Customize Security Group for this List**.
• Edit the resource mapping.	• Click **Security Group - Resource Mapping for this List**.

Follow the steps for the required task from Workflow Management Security Framework in the *Workflow Management User Guide*.

View Properties of Forms

View properties of a form to see versions and child items for the form.

To view properties of a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **View Properties**. The **Form Properties** dialog box appears.
3. Do one of the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
• See version information for the form.	• Click the **Version** tab. The **Version** tab shows the version, status, and the archived status for the form.
• See child item information for the form.	• Click the **Childitems** tab. The **Childitems** tab shows the title, description, bound to work version, and status for the form.

4. Click the **Close** button. The **Form Properties** dialog box is closed.

View Audit History of Forms

View audit history of a form to see date, time and user information when the form was locked and unlocked, and when the form owner or version was changed.

To view audit history of a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **View Audit History**. The **Audit History** dialog box appears.
   The **Audit History** dialog box shows the action, user, version, message, and date for the form.
3. Click the **Close** button. The **Audit History** dialog box is closed.

Create Forms Links

Create a form link to add a form to the menu, and to set a form as the home page.

To create a form link and add the form to the menu

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. Click the <User> icon, and then click **Repository Settings**. The **Repository Settings** page appears.
3. Click the **Advanced settings** tab, and then click **Manage Navigation List**. The **Navigation List** page appears.
4. Click **New**, and then click **Form Link**. The **New Form Link** dialog box appears.
5. In the **Title** box, type a name for the form link, and in the **Description** box, type a description for the form link.
6. In the **Select a Form** list, select a form.
7. If there are parameters for the form, then the parameters appear. In such cases, type values for the parameters, if required.
8. In the **Image Name** box, type the name of an image for the form link.
9. Under **Window Mode**, do one of the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
• Make the form appear over the window.	• Click **Pop-up**.
• Make the form appear in the window.	• Click **Inline**.

10. Under **Show in Navigation List**, select as follows:

Task	Action
• Show form link in the navigation menu of the **Enterprise Console**.	• Select **Web**.
• Show form link in the navigation menu of the **AVEVA Work Tasks Pro** app.	• Select **Mobile**.

Note:
- The **Show in Navigation List** property is applicable only when the **Window Mode** property is set to **Inline**.
- By default, the **Show in Navigation List** is set to **Web**.
- To improve usability of the navigation menu in the **AVEVA Work Tasks Pro** app, we recommend limiting the number of links.

11. In the **Menu Order** box, type a number to position the link in the **Enterprise Console** menu.

Note:
- By default, the **Menu Order** number will be one more than the highest **Menu Order** number entered for the links in the **Navigation List**.
- You can enter either a positive or a negative integer. Negative **Menu Order** numbers take precedence over positive **Menu Order** numbers.
- If you clear the **Menu Order** number, then the system will set the **Menu Order** to **0**.
- The link with the lowest **Menu Order** number appears first in the menu followed by other links in the ascending order of their **Menu Order** number.
- In case of links with the same **Menu Order** number, the last saved link takes precedence and appears before the other links with the same **Menu Order** number.
- When you upgrade the product, the **Menu Order** number for the user created links will be set to null.
- You must log out and log on to the **Enterprise Console** to see changes in the order of the menu items.

12. Click **Save**. The form link is saved and an information message appears.
13. Click **OK** (or click the **Close** button). The information message and the **New Form Link** dialog box are closed.

To set a form link as the home page

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. Click the <User> icon, and then click **Repository Settings**. The **Repository Settings** page appears.
3. Click the **Advanced settings** tab, and then click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click on a security group, and then click **Edit** (or click on a security group, and then on the action bar, click **Edit**). The **Modify a Security Group** dialog box appears.
5. In the **Default Home Page** list, select a form link.
6. Click **Save**. The security group settings are saved, the **Modify a Security Group** dialog box is closed, and the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Synchronization dialog box appears.

7. Under **Do you want to Synchronize it?,** click **Yes**, and then click **Submit**. The synchronization settings are saved and an information message appears.
8. Click **OK** (or click the **Close** button). The information message and the **Synchronization** dialog box are closed.

Note:

- To set a form as home page:
  - You must create a form link.
  - The **Window Mode** for the form link must be set to **Inline**.
  - The Default Home page item will not be available in the Navigation menu. Login and logout of the Enterprise Console to view the newly created item in the Dashboard.
  - To set the Default Home page for Limited Access Security Group in the Work Tasks Pro App, disable the permission for **Show Only Owned Item**.
- It is recommended not to use Form link with **Window Mode** as **Inline**, which uses a form associated to a synchronous workflow with an **Invoke Form** activity having **Redirect URL** and **Custom Document View URL** properties.
- Enter the values for forms parameters as follows:

Forms Parameters	**Value**
Array	Comma separated **Example:** 1,2,3,4 OR a,b,c,d
Date	mm/dd/yyyy 00:00:00 **Example:** 12/31/2099 00:00:00
Date-Time	mm/dd/yyyy HH:mm:ss in UTC format **Example:** 12/31/2099 23:59:59
Object	JSON string **Example:** {"firstname":"Allen","lastname":"Brewer","telephone":["123-456-7890","098-765-4321"]}

- The values for forms parameters are not validated when saving forms links. However, at run time, when the parameter values of type number, array, and object are converted, a messaged is logged if the conversion fails.
- When you edit a forms link, you can refresh the parameters of the form by clicking on **Reload Parameters**.
- Forms links created show the name of the forms link instead of the name of the form in the **Enterprise Console** menu.
- When you upgrade the product, you can change the name shown on the menu from forms name to forms link name in one of the following ways:
  1. Opening and saving the forms link.
  2. Appending the forms links title in **SKENavigationDetails** in the database. For example, if the forms link name is **Billing Address** then append **FormLink=Billing Address** as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
InstanceForm.aspx?ListItemId=67f364af-96e4-4d9e-9295-2a70ba3d30d5&ItemType=2&WM=1&FormLi
nk=Billing Address
```

Dynamic List Forms

You can automatically generate the form when you create a list to create a dynamic list form. You can also validate the automatically generated form to check mapping of the controls between the list and the form associated with the list.

Create Dynamic List Forms

Automatically generate the form when you create a list to create a dynamic list form.

To create a list and then automatically generate the form associated with the list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. On the action bar, click **New**, and then click **List**. The **New List** wizard appears.
3. In the **Title** box, type a name for the list, and in the **Description** box, type a description for the list.
4. Do one of the following:

Task	Action
• Enable folder support for the list to group and organize the list	• Click **Yes** for **Enable folder support?** When folder support is enabled, you can create folders in the list, and you would be able to move lite items under the folders and organize them.
• Disable folder support for the list.	• Click **No** for **Enable folder support?**

5. Click **Save & Continue**. The list is created, and the **Select Table** wizard appears.
6. Do one of the following:

Task	Action
• Store the list data in generic table.	• Click **Generic Table**. **Note**: The list fields are automatically mapped to the fields in the generic table.
• Store the list data in an existing table.	• Click **Existing Table** and select a table from the list of existing tables. **Note**: - Click the **Refresh** button to refresh the list of existing tables.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
	- If the List Form is created with an existing table and if the table contains a column name with a space, then list items are not created for such list forms.

7. Click **Save & Continue**. The list is created, the **New List** wizard is closed, and the **Table** wizard appears.
8. Design the list, and then click **Submit**. The list is saved and the **Form Settings** dialog box appears
9. If you want to edit the auto generated form, then follow the steps for *To auto generate the form* in Auto Generating Forms.
10. Click **OK**. The form is generated, the **Form Settings** dialog box is closed, and an information message appears.
11. Click **OK**. The information message and the **List Designer** page are closed.

To edit the auto generated form

1. In the **Form Settings** dialog box, click **Edit Auto Generated Form**.
2. Click **OK**. The form is generated, the **Form Settings** dialog box is closed, and an information message appears.
3. Click **OK**. The information message and the **List Designer** page are closed, and the **Forms Designer** page appears.
4. Design the form, and then click the **Close** button. The form is saved and the **Forms Designer** page is closed.

Note:
- The auto generated form is saved, with the same name as that of the list, in the **Dynamic List Forms** folder in the **Forms** page.
- The form associated with the list must be published before using the list. Otherwise, you will not be able to add or edit list items.
- List Forms should not be associated to workflows.
- If an existing custom table from the repository is deleted, the table name still appears in CustomList table List Box because the **Refresh** button is not working.
- For information about using lists, see Lists.

Validate Dynamic List Forms

You can validate the automatically generated form to check mapping of the controls between the list and the form associated with the list.

To check mapping of the controls between a list and the form associated with the list

1. In the **Forms Designer** page, on the action bar, click the **Validate List** button. The mapping of the controls between the list and the form associated with the list is checked.
2. Do one of the following:

Outcome	Action
• Validation fails and an error page appears.	a. Read the error and the description.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Outcome	Action
	b. Click **Close** (or click the **Close** button). The error page is closed. c. Edit the controls on the **Forms Designer** page to correct the errors, if possible.
• Validation is successful and a message bar appears.	• Click the **Close** button.

3. Click the **Close** button. The **Forms Designer** page is closed.

Note: In the **Forms Designer** page, on the action bar, the **Validate List** button is available only when:
- The **Edit Auto Generated Form** option is selected in the **Form Settings** dialog box while creating and submitting the list.
- The **Yes** option is selected in the **Form Settings** dialog box while editing and submitting the list.

Associate Forms to Workflows

Forms are seamlessly integrated with workflows. You can associate a form to a new or an existing workflow. The XML variables associated with a form allow you to build a complex workflow.

Associating a form to different types of workflow activities such as Invoke Form, Information, and so on help to create an end-to-end business application.

Associate Forms to Workflows

Associate a form to a workflow to initiate the workflow when the form is submitted.

Associated forms appear in the **Fill Form** page only when the forms, the workflows, and the folders containing them are published.

You can associate multiple forms to a workflow.

You can overwrite SFFormData of a workflow associated with another version of the form, considering that the existing version of the form is a primary form.

To associate a form to a workflow

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Associate** (or click on a form, and then on the action bar, click **Associate**).
3. Do one of the following:

Task	Action
• Associate a form to the workflow.	• Follow the steps for Binding a form to the workflow.
• Edit association of the the form already associated to a workflow.	• Follow the steps for Editing Association. **Note**: When you edit the association, the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
	existing association details of the form gets deleted.
• Edit mappings of the form already associated to a workflow.	• Follow the steps for Editing Mappings.

Note:
- Ensure the form is published before associating the form to a workflow. Otherwise, the **SFFormData** will not show the schema in the workflow.
- Ensure the associated form is republished, the **SFFormData** is reloaded and saved, and the workflow is republished when there is an update in the form. Otherwise, the **SFFormData** will not show the updated schema in the workflow.
- When the workflow is associated with the unpublished Form (primary Form), you need to either publish the primary Form or dissociate the workflow to associate it with the other Form.
- You cannot unpublish the Form defined as a part of mapping details in the other Form.

Bind a Form to the Workflow

Use **Bind to an existing workflow** option to bind the form to the workflow. You can also associate a form to the workflow to which another form is already associated.

In the **Form Binding** window, you can do one of the following:

Task	Action
• Select a workflow to which the form will be associated.	1. Select **Bind to an existing workflow**. 2. Select a workflow, and then select a version of the workflow. 3. Select the **Execute workflow synchronously** check box if you want to execute the associated form synchronously. 4. Click **Save**, and then click **Close**. The form is associated to the workflow. **Note**: When you select a workflow to which another form is already associated, you get the message *Workflow is already associated to a form. Would you like to specify the mappings for the Workflow Xml?* Click **OK** in the message belly bar and follow the steps for Form XML Mapping to associate a form.
• Create a workflow to which the form will be associated.	• Follow the steps for To design a workflow to which the form will be associated.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
• Cancel further operations.	1. Click **No task to be performed now**. 2. Click **Finish**. The **Bind** wizard is closed.

Form XML Mapping

In the **Form XML Mapping** window, you can map the elements and the attributes of the workflow XML variable with the form's XML structure.

To map the elements and attributes

1. Select a form name from the **Form Name** drop-down list and then select a version from the **Form Version** drop-down list which would be used for triggering the workflow. The **Form Name** drop-down lists all the published forms associated to the workflow.

   Note: Select the Form Name and Form Version having the same schema as that of Workflow XML which would be used to trigger the workflow.
2. Select the attributes and elements from the **Workflow XML** column to be mapped with the form being associated.
3. Select the attributes or elements from the **Form XML** column to be mapped with the selected elements and attributes of the **Workflow XML**.
4. Click **Add Record** to specify more mappings.
5. Repeat steps 2 and 3 to add and map more attributes and elements to the form being associated.
6. Click **Save**. The successful mapping message appears.
7. Click **Close**.

Note:
- The form being associated and mapped will not be listed in the **Form Name** drop-down list in the **Form XML Mapping** page.
- You cannot dissociate a form that is defined as a part of mapping details in another form.
- The schema of the Workflow XML should be same as that of the mapped Form XML. When the parent node having child elements in Workflow XML is mapped with the parent node in Form XML, then only the elements and attributes with the same XML nodes will be considered.
- The mappings cannot contain same Workflow XML values.
- The runtime execution of the mappings would be as per the order defined.
- The Form Xml Mapping is not considered while packaging the Form artifact.
- Mapping of the skFilePath attribute of an Attachment-based XML node is not supported.

Edit Association of the Already Associated Form

When the workflow is already associated to the form and when you associate the form again to the workflow, you get the **Edit Association** option to redefine or edit the existing association details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To edit association of the already associated form

1. In the Enterprise Console, click the Menu button, and then click Forms. The Forms page appears.
2. Right-click on a form, and then click Associate (or click on a form, and then on the action bar, click Associate).

   The message *Workflow is already associated to the form with the mapping details. Would you like to edit the association or edit the mapping?* appears.
3. Click **Edit Association**.

   Note: When you edit the association, the existing association and mapping details of the form is removed.
4. Bind the form to an existing workflow. Follow the steps for Binding a form to an existing workflow.

   The message *Workflow is already associated to a form. Would you like to specify the mappings for the Workflow Xml?* appears.
5. Click **OK**.

   The **Form XML Mapping** window appears.
6. Follow the steps for Form XML Mapping.

   The form is associated again to an existing workflow with the updated association and mapping details.

Edit Mappings of the Already Associated Form

When the workflow is already associated to the form with mapping details and when you associate the form again to the workflow, you get the **Edit Mappings** option to edit the existing mapping details.

To edit mappings of the already associated form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Associate** (or click on a form, and then on the action bar, click **Associate**).

   The message *Workflow is already associated to the form with the mapping details. Would you like to edit the association or edit the mapping?* appears.
3. Click **Edit Mappings**.

   The **Form XML Mapping** window appears.
4. Follow the steps for Form XML Mapping.

   The form is associated again to an existing workflow with the updated mapping details.

View Association of Forms with Workflows

View association of a form with the workflow to see which workflow is initiated when the form is submitted.

To view the workflow name for the associated form

- In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The Forms page appears.

  The **Bound to Workflow** column shows the workflow name for the associated form.

To view the workflow name, version, and status for the associated form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Right-click on a form, and then click **View Association**. The **Associated Workflow** dialog box appears.

   The **Associated Workflow** dialog box shows the workflow name, version, and status for the associated form.
3. Click the **Close** button. The **Associated Workflow** dialog box is closed.

Dissociate Forms from Workflows

Dissociate a form to remove its association from the workflow.

To dissociate a form from the workflow

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on an associated form, and then click **Dissociate**. The form is dissociated from the workflow and a confirmation message appears.
3. Click **Yes**. The confirmation message is closed.

Note: When a form is dissociated from the workflow, the form is not available in Fill Forms.

Design Workflows for Associated Forms

Design a workflow for the associated form to initiate the workflow when the form is submitted.

Follow the steps for Associate Forms to Workflows till you a get the **Bind** wizard.

To design a workflow to which the form will be associated

1. In the **Bind** wizard, click **Design a new workflow**.
2. Select the **Execute workflow synchronously** check box if you want to execute the associated form synchronously.
3. Click **Next**. The **New Workflow** wizard appears.
4. In the **Title** box, type a name for the workflow, and in the **Description** box, type a description for the workflow.
5. Click **Save & Continue**. A workflow is created, and the **Workflow Options** wizard appears.
6. Do one of the following:

Task	Action
• Design the workflow to which the form will be associated.	a. Click **Design the Workflow now**. b. Click **Finish**. The **Process Designer for Workflow** page appears. c. Create a new workflow.
• Cancel further operations.	a. Click **No task to be performed now**. b. Click **Finish**. The **Workflow Options** wizard is closed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Forms Designer

Forms Designer is a "what you see is what you get (WYSIWYG)" web-based design tool, with a drag-and-drop capable interface, to create rich web-based forms. Forms designer supports direct drop, smooth scroll, and device previews.

- Drag a control from the toolbox to the canvas to add the control to the form.
- Drag a control on the canvas to move or clone a control on the form.
- Edit tags inline and manage actions for controls with an on demand control shortcut menu.
- Create complex responsive layouts without affecting the schema with smart controls.
- Select columns and adjust column widths with ease.

Use **Forms Designer** to create rich forms with various controls.

To access Forms Designer

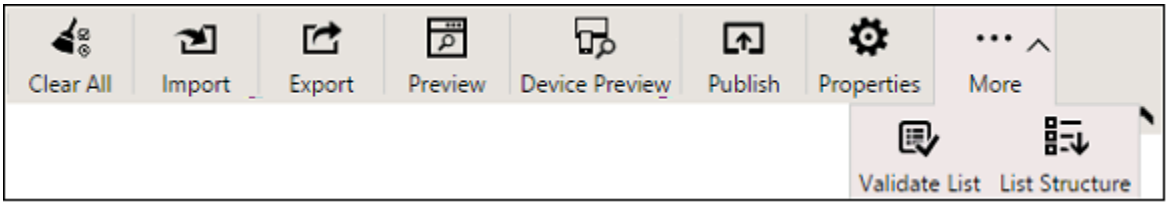
1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.

Forms Designer Interface

Forms Designer interface has an action bar, toolbox, and canvas.

Action Bar

Action bar has commands to manage the form and controls on the form.



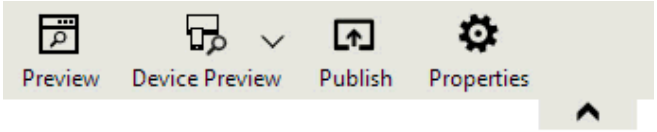
Menu/Command	Description
Clear All	Clear the controls. For more information, see Clear Controls.
Import	Import the form. For more information see, Import Forms.
Export	Exporting the form. For more information see, Export Forms.
Preview	Preview the form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Menu/Command	Description
	For more information, see Preview Forms.
Device Preview	Preview the form for mobile device. For more information, see Preview Forms.
Publish	Publish the form. For more information, see Publishing Forms.
Properties	Edit properties of the form. For more information, see Edit Properties of Forms.
Validate List	Validate form associated with the list. For more information, see Validate Dynamic List Forms.
List Structure	View the structure of the list. For more information, see List Structure.

To manage the visibility of the action bar

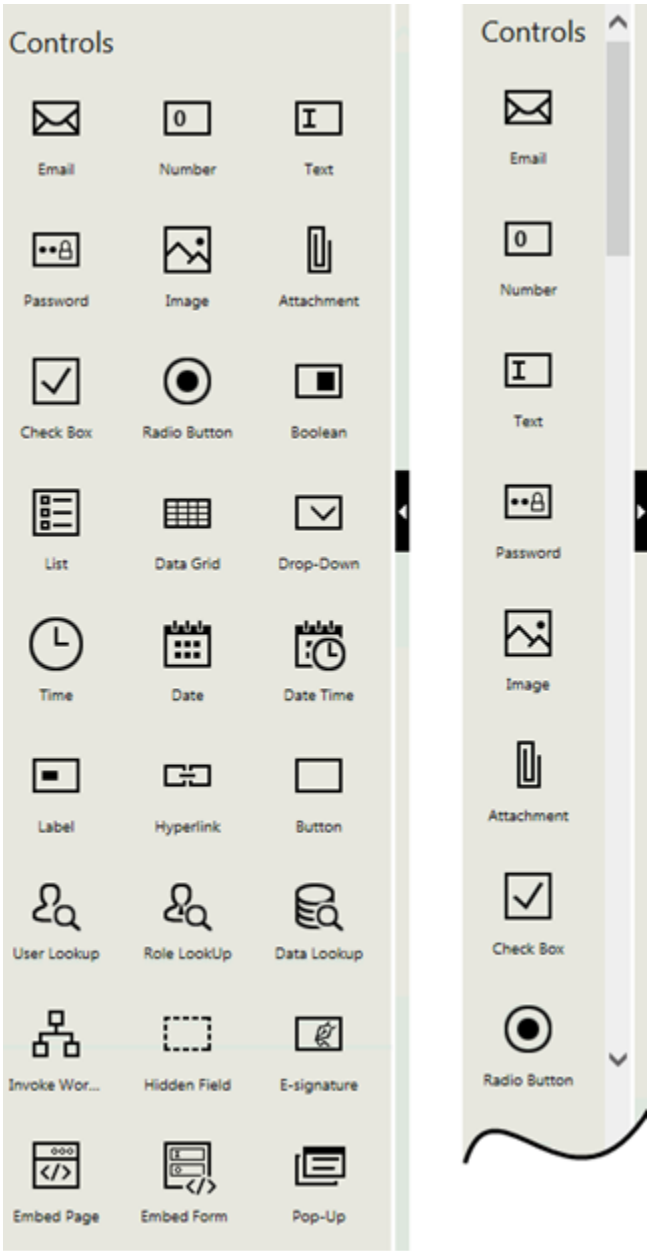
1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. Do one of the following:

Task	Action
Hide the action bar.	On the action bar, click the **Arrow** button. The action bar is hidden in the **Forms Designer** except for the **Arrow** button. 
Show the action bar.	Click the **Arrow** button at the top of the **Forms Designer**. The action bar is shown in the **Forms Designer**.

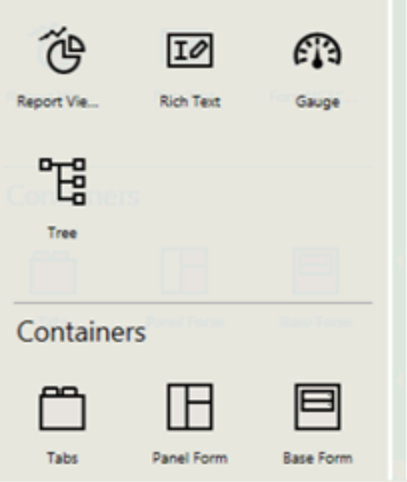
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Toolbox

Toolbox has categories to group controls for designing the form. The size of the toolbox can be changed from three columns to one column to increase the size of the canvas.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



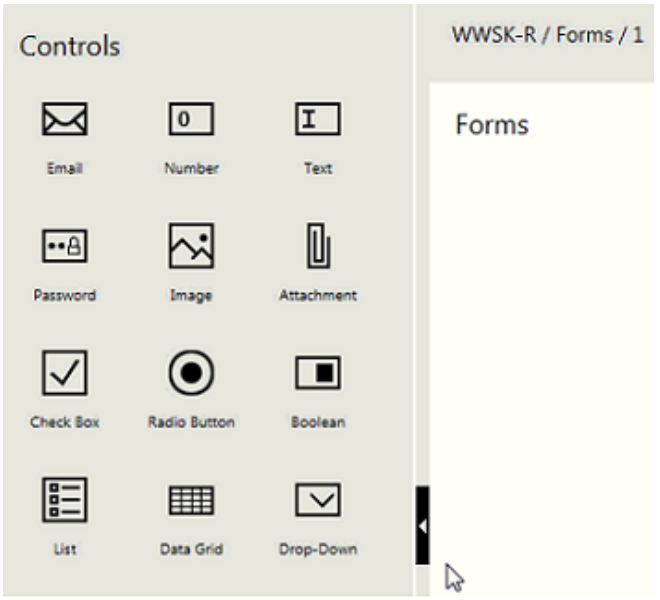
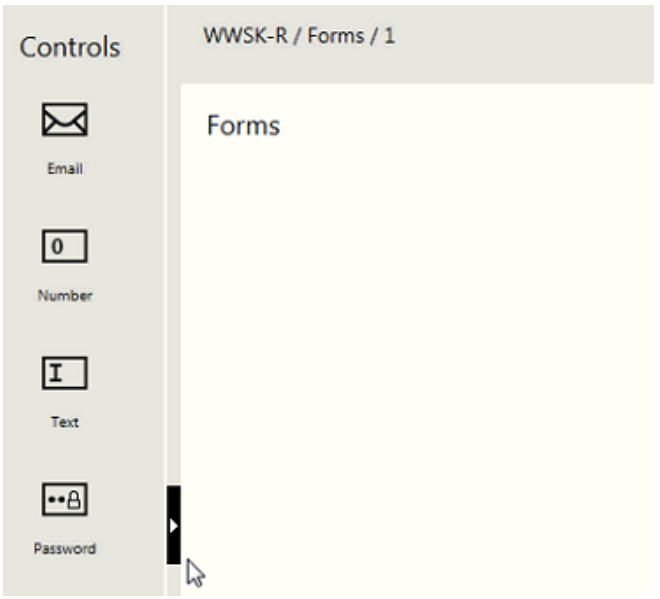
Categories	Description
Controls	Common controls to design the form. For more information, see Controls.
Containers	Container controls to design the form. For more information, see Containers.

To manage the size of the toolbox

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. Do one of the following:

Task	Action
Change size of the toolbox from three columns to one column.	On the toolbox, click the **Arrow** button. The size of the toolbox is changed from three columns to one column.

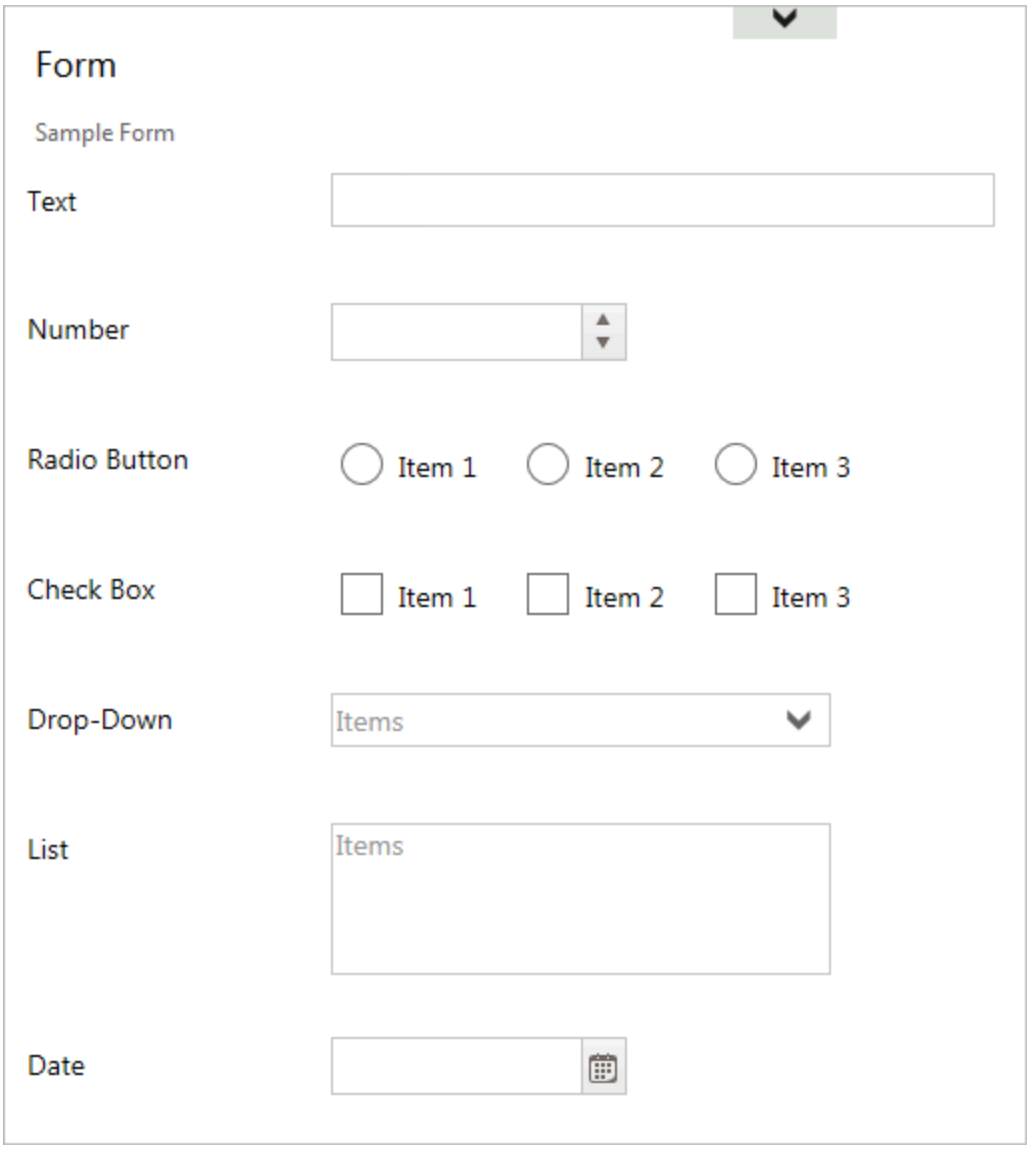
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
	
Change size of the toolbox from one column to three columns.	On the toolbox, click the **Arrow** button. The size of the toolbox is changed from one column to three columns. 

Canvas

Canvas is the area in the **Forms Designer** where the form is designed by dragging controls from the toolbox. Dragging is also supported for repositioning and copying controls.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Control Shortcut Menu

Control shortcut menu has commands to manage the control. Point to a control on the canvas to access the shortcut menu.

Command	Description
Move	Move the control to the desired location on the canvas. For more information, see Moving Controls.
Clone	Create a copy of the control and then move it to the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Command	Description
	desired location on the canvas. For more information, see Cloning Controls.
Delete	Delete the control. For more information, see Deleting Controls.
Replace	Replace the control. For more information, see Replace Controls.
Properties	Edit properties of the control. For more information, see Editing Properties of Controls.

To access control shortcut menu

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. Point to the control on the canvas. The **Shortcut** menu appears.

Note: To design a form or edit an existing form in **Forms Designer**, the Workflow Management Forms Server license has to be acquired.

Manage Forms

You can edit properties of the form, preview and publish forms. You can view the structure of the list associated with the form.

- Edit Properties of Forms
- Preview Forms
- Publish Forms
- List Structure

Edit Properties of Forms

Edit properties of a form to change appearance and functionality of the form.

To edit properties of a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. On the action bar, click **Properties**. The **Properties** dialog box appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Edit properties of the form, and then click **Apply**. The properties are saved and the **Properties** dialog box is closed.

Preview Forms

Preview a form to check look and feel of the form at run-time and the XML generated for the form.

To preview a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. Do one of the following:

Task	Action
• Check look and feel of the form for browser.	• On the Action bar, click **Preview** to preview the form.
• Check look and feel of the form for mobile device.	• On the Action bar, click **Device Preview** to preview the form.

Publish Forms

Publish a form to use the form in a workflow.

To publish a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. On the action bar, click **Publish**. The form is published and an information message appears.
4. Click **OK**. The information message is closed.

List Structure

View the structure of the list associated with the form. The **List Structure** option is applicable only for the forms that are generated from a AVEVA Work Tasks List. You can also export the list structure as an XML file.

To view the list structure

1. In the **Enterprise Console**, click menu icon, and then click **Forms**.

   The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**).

   The **Forms Designer** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. On the action bar,click **List Structure**.

   The **List Structure** dialog box appears.

   You can view the structure of the list.
4. To export the list structure as an XML file, click **Export**.

   The list structure is exported in an XML file.
5. Click **Cancel** to close the dialog box.

Manage Controls

You can add, move, clone, delete, clear and replace controls. You can also edit the name and properties of control.

- Add Controls
- Move Controls
- Clone Controls
- Delete Controls
- Clear Controls
- Replace Controls
- Edit Name of Controls
- Edit Properties of Controls

Add Controls

Add a control to the canvas to build the form.

To add a control

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click **Design**). The **Forms Designer** page appears.
3. Drag a control from the toolbox to the desired location on the canvas. The selected control appears on the canvas.

Move Controls

Move a control to reposition the control to the desired location on the canvas.

To move a control

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click **Design**). The **Forms Designer** page appears.
3. Point to the control on the canvas. The **Shortcut** menu appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click the **Move** button, and then drag the control to the desired location on the canvas.

Clone Controls

Clone a control to create a similar control on the canvas.

To clone a control

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click **Design**). The **Forms Designer** page appears.
3. Point to the control on the canvas. The **Shortcut** menu appears.
4. Click the **Clone** button, and then drag the control to the desired location on the canvas.

Note:
- You can clone all controls including container controls.
- When you clone a container control, the container including the controls in the container is copied.
- When you clone a control, the copy of the control is the same as the original control except for the ID, that is generated automatically.

Delete Controls

Delete a control when the control is no longer required.

To delete a control

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click **Design**). The **Forms Designer** page appears.
3. Point to the control on the canvas. The **Shortcut** menu appears.
4. Click the **Delete** button. A confirmation message appears.
5. Click **OK**. The control is deleted from the canvas and the confirmation message is closed.

Note: If you delete a control, then the control cannot be restored.

Clear Controls

You can clear all the controls in a form.

Note: You cannot recover the controls after clearing them using this option. Export the form as an XML file to maintain a backup of the form before executing this option.

To clear all controls

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. On the action bar, click **Clear All**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click **OK**. All the controls in the form are deleted.

Replace Controls

You can replace a control to change a control to a different control at the same position on the canvas.

To replace a control

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click **Design**). The **Forms Designer** page appears.
3. Point to the control on the canvas. The **Shortcut** menu appears.
4. Click the **Replace** button. The **Replacement Control Configuration** page appears.
5. In the **Replacement Control** list, click a control.

   The **Replacement Control** drop-down lists,

   a. the controls if the control you want to replace is the basic control

   b. the containers if the control you want to replace is the container (Base Form and Panel Form)
6. Do one of the following:

Task	Action
• Copy values of similar properties from the existing control to the replaced control.	• Under **Copy Similar Properties**, click **Yes**.
• Delete the existing control and add a new control.	• Under **Copy Similar Properties**, click **No**.

7. Click **Apply**. The control is replaced and the **Replacement Control Configuration** page is closed.

Note:
- If you replace a control, then the control cannot be restored.
- Replacing a control with the **Pop-Up**, **Embed Form**, or **Tab** control or vice-versa is not supported.
- You can replace the basic control with another basic control and not with the container.
- You can replace the container with another container and not with the basic control.

Edit Name of Controls

You can edit name of a control inline to quickly change the name that identifies the control.

To edit name of a control inline

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click **Design**). The **Forms Designer** page appears.
3. Point to the control, click the existing name, and then type a new name.

   The new name is saved.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Edit Properties of Controls

You can edit properties of a control to change appearance and functionality of the control.

To edit properties of a control

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then on the action bar, click **Design**). The **Forms Designer** page appears.
3. Point to the control on the canvas. The **Shortcut** menu appears.
4. Click the **Properties** button. The **Properties** dialog box appears.
5. Edit properties of the control, and then click **Apply**. The properties are saved and the **Properties** dialog box is closed.

Control Properties

The properties for a control are grouped under the following tabs depending on the control:

- Basic
- Appearance
- Validation
- Advanced
- Scripts

For information about the properties of a control, see Controls.

Design Guidelines

This section provides information about the custom styles, boundaries and limits, and the best practices to follow while designing the forms.

- Custom Styles
- Boundaries and Limits
- Best Practices

Custom Styles

Custom style sheets help you to change the style of the forms user interface and customize the style according to your corporate style guidelines.

Use custom style sheets to change style for HTML elements of controls using CSS classes.

To create a custom style

1. Create a style sheet in the **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/css/** folder.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. If Theme is applied, the css file must be tagged with **FormCustomThemePath**, for example, var cssFiles=["FormCustomThemePath:MyCustomStyleSheet.css"]; and must be placed in **../<BPMUITemplates>/Default/NextGenForms/custom/Themes/<ThemeName>/css**.
3. If Theme is not applied, the css file can be placed in **../<BPMUITemplates>/Default/NextGenForms/custom/css/MyCustomStyleSheet.css**.
   You can create folder structure in **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/css/** to place the css files, for example, **../<BPMUITemplates>/Default/NextGenForms/custom/css/Mycssfolder/MyCustomStyleSheet.css**
4. Enter the name of the style sheet for the **Custom Style Sheet** property of the form. For example, enter **MyCustomStyleSheet** for **MyCustomStyleSheet.css**.
5. Enter the name of the identifier for the **Custom Style Identifier** property of the control.
6. Edit the style sheet to define styles for HTML elements of controls using CSS classes.

Note:
- Prefix the **Custom Style Identifier** of a control with the **CSS Class** name to apply the style to the specific control.
- Use the **CSS Class** name by itself to apply the style to all the controls.
- Suffix your CSS declarations with `!important` to ensure it takes precedence over normal declarations.

Arc

You can modify the style for the HTML elements of Arc using SVG elements and CSS classes.

HTML Element	CSS Class
Unit of Measurement (UOM)	.skarc-uom#skarc-uomid
Value	.skarc-value#skarc-valueid
Title	.skarc-title#skarc-titleid
Scale Ticks	.skarc-scaleticks
Scale Text	.skarc-scaletext#skarc-scaletextid

Prefix the Custom Style Identifier of a control with the CSS Class name to apply the style to the specific control. For example:

```
/* arc1id is the Custom Style Identifier of the Arc control. */
.arc1id_skctr
```

Note: If you plan to use your own CSS file, the Custom Style Identifier that you define in the file must match the value that you specify in the **Custom Style Sheet Identifier** property in the **Appearance** tab of the control's properties page. For example: `arc1id`

Example

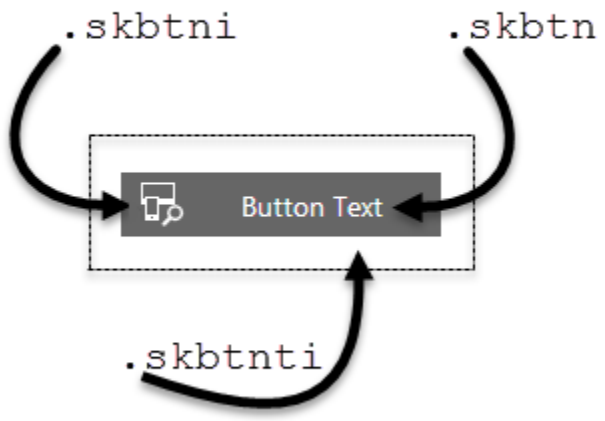
```
/* Change scale text fill color and font to black and Roboto. */
.arc1id_skctr .skarc-scaletext#skarc-scaletextid
{
fill: black !important;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
font-family: 'Roboto' !important;
font-weight: 400 !important;
font-size: 15px !important;
}
```

Button

You can change style for HTML elements of **Button** using CSS classes.



HTML Element	CSS Class
Button (for Button with Text and Image)	.skbtnti
Button Text	.skbtn
Button Image	.skbtni

For other HTML elements and the corresponding CSS classes related to **Button**, see Other Controls.

Prefix the **Custom Style Identifier** of a control with the **CSS Class** name to apply the style to the specific control. For example:

```
/* Button1 is the Custom Style Identifier of the Button control. */
.Button1_skbtn
```

Example

```
/* Change the Button text color to red. */
/* Button1 is the Custom Style Identifier. */
.Button1_skbtn
{
 color: red !important;
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: Suffix your CSS declarations with `!important` to ensure it takes precedence over normal declarations.

Card

You can use custom style sheets to change style for HTML elements of controls using CSS classes.

For detailed steps on creating a custom style, see Custom Styles.

Modify the style for the HTML elements of Card using SVG elements and CSS classes.

HTML Element	CSS Class
CardContainer	.skcardwrapper
CardControl	.skcard.skcardflt
CardHeader	.skcardheader
CardContent	.skcardcontent
CardValue1Container	#skcardval1block
CardValue1	#skcardvl1
CardValue2	#skcardvl2
Cardv345Wrapper	#skcardvl345Wrap
CardValue3	#skcardvl3
CardValue4	#skcardvl4
CardValue5	#skcardvl5
CardImage	.skcardImg
CardFooter	.skcardfooter
CardButtonDefault	.skcardbtndef
CardButtonBlack	.skcardbtnblk
CardActionButton1	#skcardactbtn1
CardActionButton2	#skcardactbtn2
CardActionButton3	#skcardactbtn3

Prefix the Custom Style Identifier of a control with the CSS Class name to apply the style to the specific control. For example:

```
/* mdmcardcn is the Custom Style Identifier of the Card control. */
.mdmcardcn_skctr
```

Note: If you plan to use your own CSS file, the Custom Style Identifier that you define in the file must match the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

value that you specify in the **Custom Style Sheet Identifier** property in the **Appearance** tab of the control's properties page. For example: `mdmcardcn`

Examples

```
/* Change card header font size. */
.mdmcardcn_skctr .skcardheader h1
{
font-size: 14px!important;
}
/* Change card value font properties and color. */
.mdmcardcn_skctr #skcardvl3 p #skcardvl4 p #skcardvl5 p
{
font-size: 8px!important;
color: red!important;
font-family: 'robotobold' !important;
}
```

Gantt Chart

You can modify the style for the HTML elements of Gantt Chart using SVG elements and CSS classes.

HTML Element	CSS Class
Title	.skgantt-title#skgantt-titleid
Start Time and End Time	.skgantt-datetimetext#skgantt-datetimetextid
Segment Highlight	.skgantt-segment#skgantt-segmentid
GanttHeader	.skgantt-header#skgantt-headerid
GanttValue1	.skgantt-vl1#skgantt-vl1id
GanttValue2	.skgantt-vl2#skgantt-vl2id
GanttValue3	.skgantt-vl3#skgantt-vl3id

Prefix the Custom Style Identifier of a control with the CSS Class name to apply the style to the specific control. For example:

```
/* gantt1id is the Custom Style Identifier of the Gantt Chart control. */
.gantt1id_skctr
```

Note: If you plan to use your own CSS file, the Custom Style Identifier that you define in the file must match the value that you specify in the **Custom Style Sheet Identifier** property in the **Appearance** tab of the control's properties page. For example: `gantt1id`

Example

```
/* Change title color and font to black and Roboto. */
.gantt1id_skctr .skgantt-title#skgantt-titleid
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
fill: black !important;
font-size: 23px !important;
font-family: 'Roboto' !important;
font-weight: 400 !important;
}
```

Gauge

You can change style for HTML elements of **Gauge** using SVG elements and CSS classes.

HTML Element	CSS Class
Control Area	.skga
Gauge Area	.skga-guage
Gauge Value Labelr	.skga-label

Prefix the **Custom Style Identifier** of a control with the **CSS Class** name to apply the style to the specific control. For example:

```
/* E1 is the Custom Style Identifier of the Gauge control. */
.E1_skctr
```

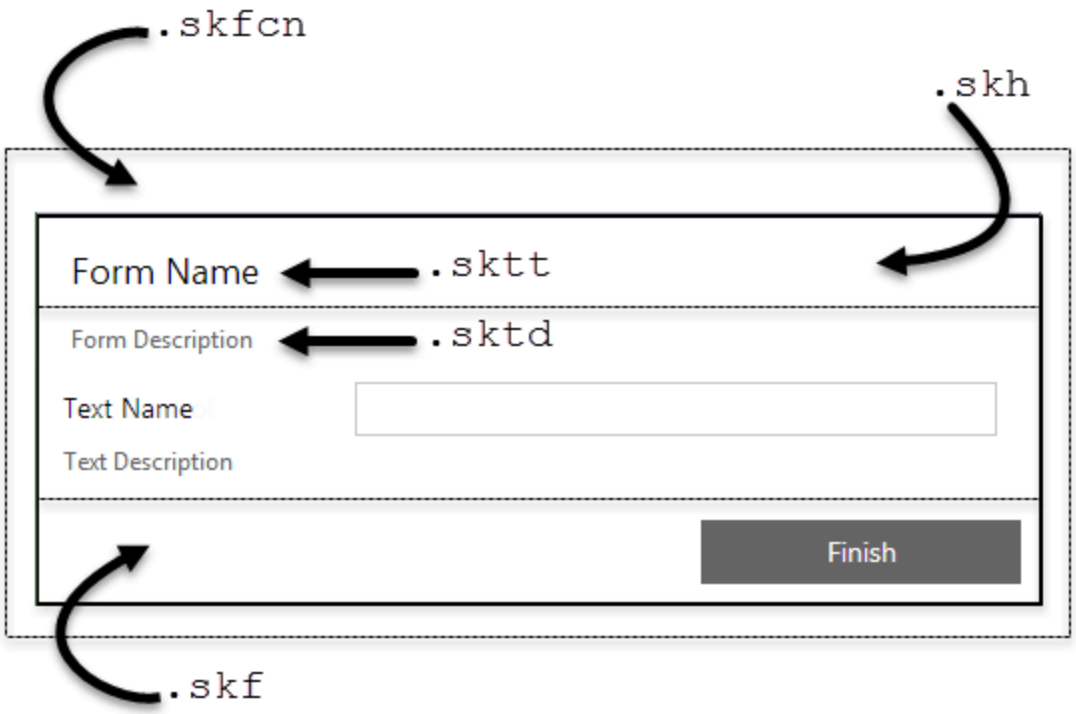
Example

```
/* Change value label color to green and bold. */
.E1_skctr skga-label
{
font-weight:bold;
color: ‘green’;
}
```

Form

You can change style for HTML elements of **Form** using CSS classes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



HTML Element	CSS Class
Form	.skfcn
Form Header	.skh
Form Footer	.skf
Form Name	.sktt
Form Description	.sktd

Example

```
/* Change Form background color to grey. */
.skfcn
{
 background-color: grey !important;
}
```

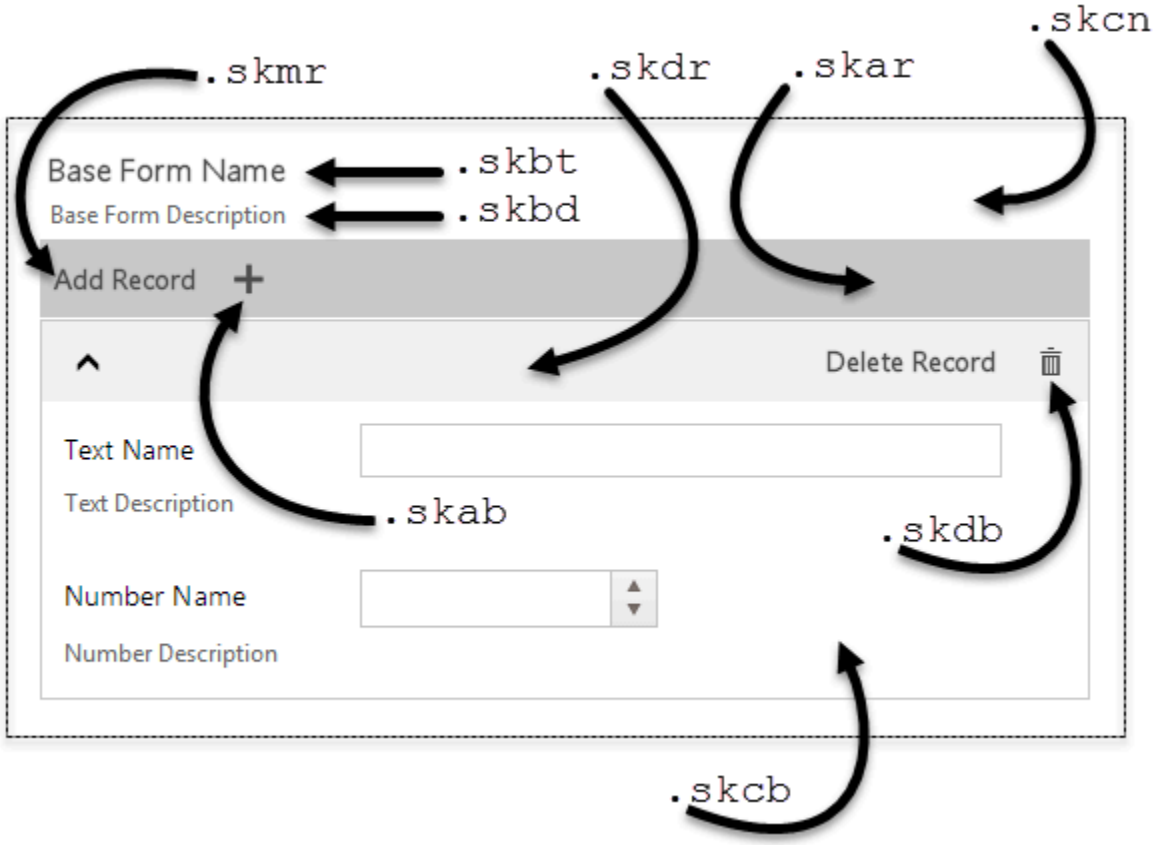
```
/* Change Form background color to black except in Form Designer. */
html:not(.skd) .skfcn
{
background-color: black;
}
```

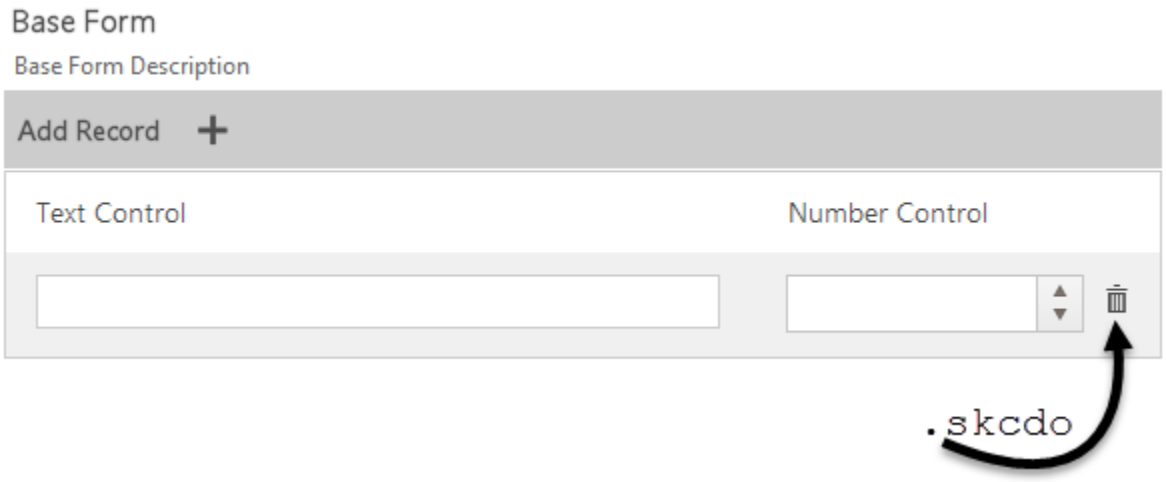
Note: Suffix your CSS declarations with `!important` to ensure it takes precedence over normal declarations.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Base Form

You can change style for HTML elements of **Base Form** using CSS classes.





HTML Element	CSS Class
Base Form	.skcn
Base Form Name	.skbt

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

HTML Element	CSS Class
Base Form Description	.skbd
Record Caption	.skmr
Add Record Row	.skar
Add Record Icon	.skab
Delete Record Row	.skdr
Delete Record Icon (Default)	.skdb
Delete Record Icon (Grid)	.skcdo
Base Form Body	.skcb

Prefix the **Custom Style Identifier** of a control with the **CSS Class** name to apply the style to the specific control. For example:

```
/* BaseForm1 is the Custom Style Identifier of the Base Form container. */
.BaseForm1_skbt
```

Example

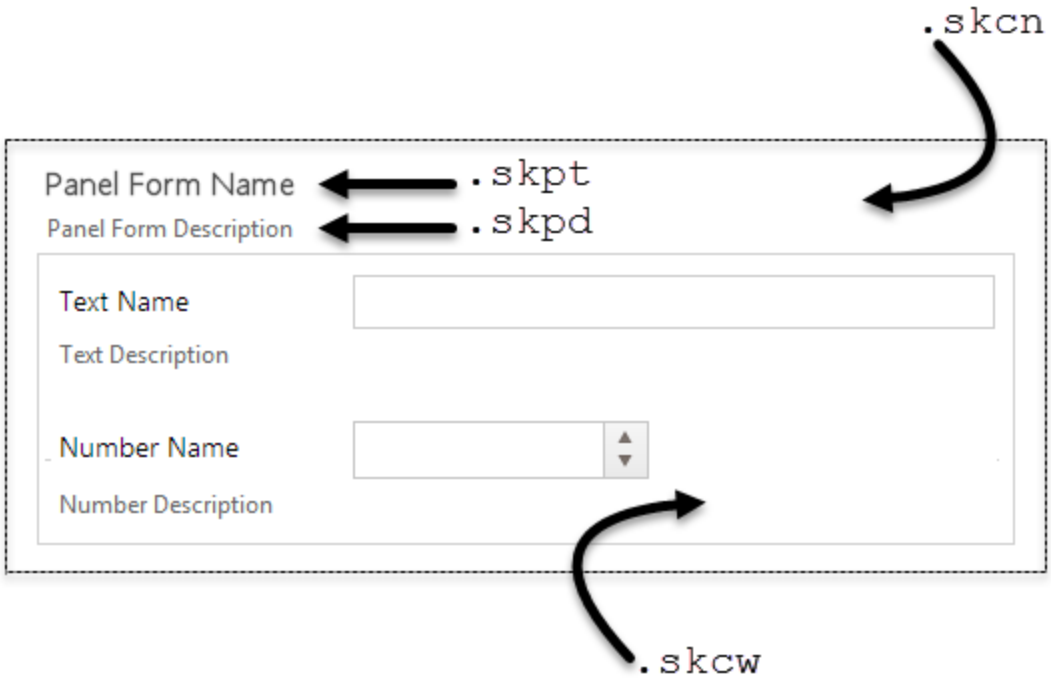
```
/* Change font size of Base Form name to 20px. */
/* BaseForm1 is the Custom Style Identifier. */
.BaseForm1_skbt
{
 font-size: 20px !important;
}
```

Note: Suffix your CSS declarations with `!important` to ensure it takes precedence over normal declarations.

Panel Form

You can change style for HTML elements of **Panel Form** using CSS classes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



HTML Element	CSS Class
Panel Form	.skcn
Panel Form Name	.skpt
Panel Form Description	.skpd
Panel Form Body	.skcw

Prefix the **Custom Style Identifier** of a control with the **CSS Class** name to apply the style to the specific control. For example:

```
/* PanelForm1 is the Custom Style Identifier of the Panel Form container. */
.PanelForm1_skpt
```

Example

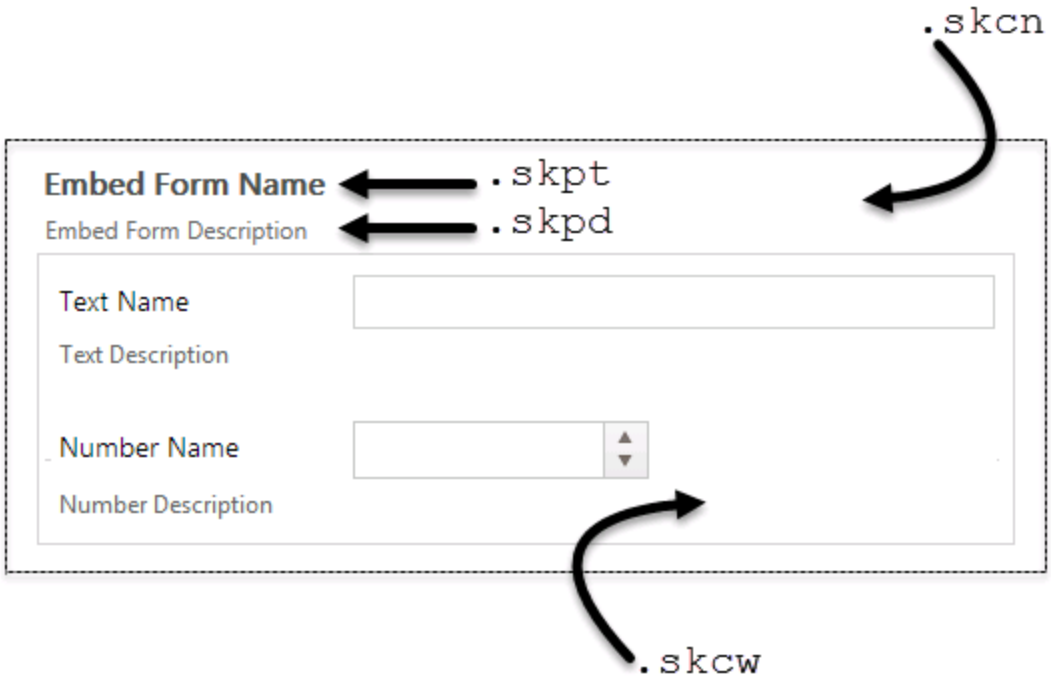
```
/* Change font size of Panel Form name to 20px. */
/* PanelForm1 is the Custom Style Identifier. */
.PanelForm1_skpt
{
 font-size: 20px !important;
}
```

Note: Suffix your CSS declarations with `!important` to ensure it takes precedence over normal declarations.

Embed Form

You can change style for HTML elements of **Embed Form** using CSS classes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



HTML Element	CSS Class
Embed Form	.skcn
Embed Form Name	.skpt
Embed Form Description	.skpd
Embed Form Body	.skcw

Prefix the **Custom Style Identifier** of a control with the **CSS Class** name to apply the style to the specific control. For example:

```
/* EmbedForm1 is the Custom Style Identifier of the Embed Form container. */
.EmbedForm1_skpt
```

Example

```
/* Change font size of Embed Form name to 20px. */
/* EmbedForm1 is the Custom Style Identifier. */
.EmbedForm1_skpt
{
 font-size: 20px !important;
}
```

Note: Suffix your CSS declarations with `!important` to ensure it takes precedence over normal declarations.

Progress Bar

You can modify the style for the HTML elements of Progress Bar using SVG elements and CSS classes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

HTML Element	CSS Class
Unit of Measurement (UOM)	.skprogress-uom#skprogress-uomid
Value	.skprogress-value#skprogress-valueid
Title	.skprogress-title#skprogress-titleid
Scale Ticks	.skprogress-scaleticks
Scale Text	.skprogress-scaletext#skprogress-scaletextid

Prefix the Custom Style Identifier of a control with the CSS Class name to apply the style to the specific control. For example:

```
/* progress1id is the Custom Style Identifier of the Progress Bar control. */
.progress1id_skctr
```

Note: If you plan to use your own CSS file, the Custom Style Identifier that you define in the file must match the value that you specify in the **Custom Style Sheet Identifier** property in the **Appearance** tab of the control's properties page. For example: `progress1id`

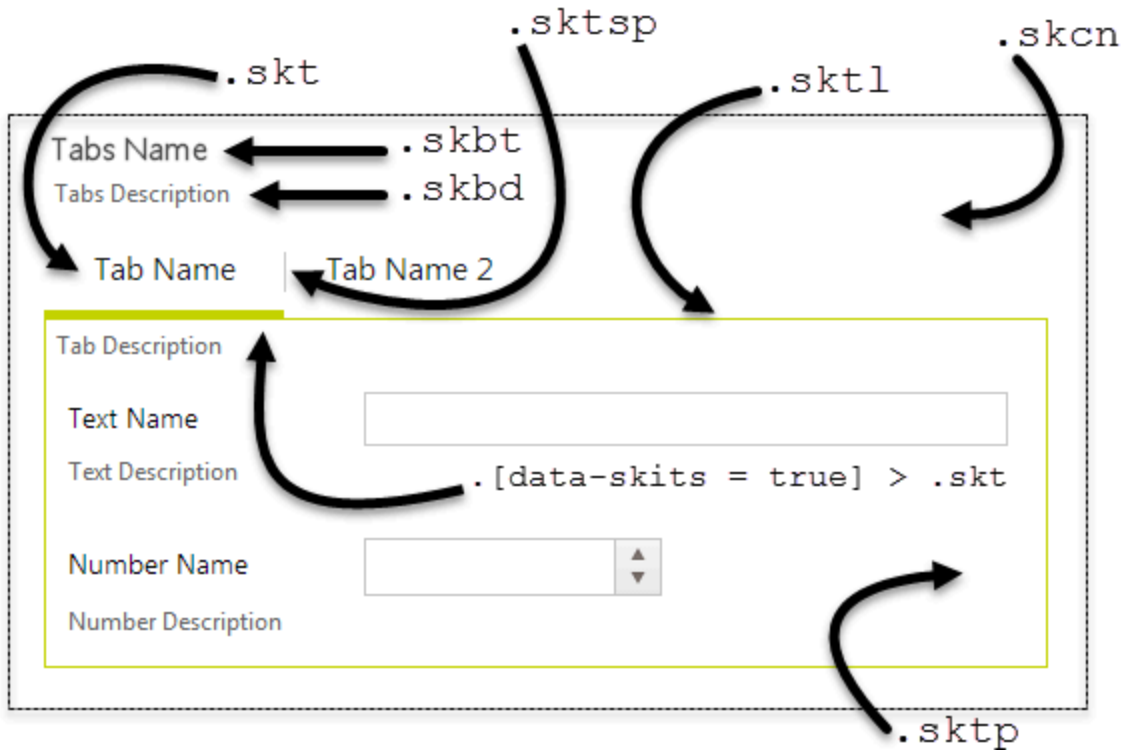
Example

```
/* Change scale text fill color and font to black and Roboto. */
.progress1id_skctr .skprogress-scaletext#skprogress-scaletextid
{
fill: black !important;
font-family: 'Roboto' !important;
font-weight: 400 !important;
font-size: 15px !important;
}
```

Tabs

You can change style for HTML elements of **Tabs** using CSS classes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



HTML Element	CSS Class
Tabs	.skcn
Tab	.skt
Tabs/Tab Name	.skbt
Tabs/Tab Description	.skbd
Selected Tab	[data-skits=true] > .skt
Tab Line	.sktl
Tab Body	.sktp
Tab Separator	.sktsp

Prefix the **Custom Style Identifier** of a control with the **CSS Class** name to apply the style to the specific control. For example:

```
/* Tabs1 is the Custom Style Identifier of the Tabs container. */
.Tabs1_skcn
```

Example

```
/* Change the Tabs border color to blue. */
/* Tabs1 is the Custom Style Identifier. */
```

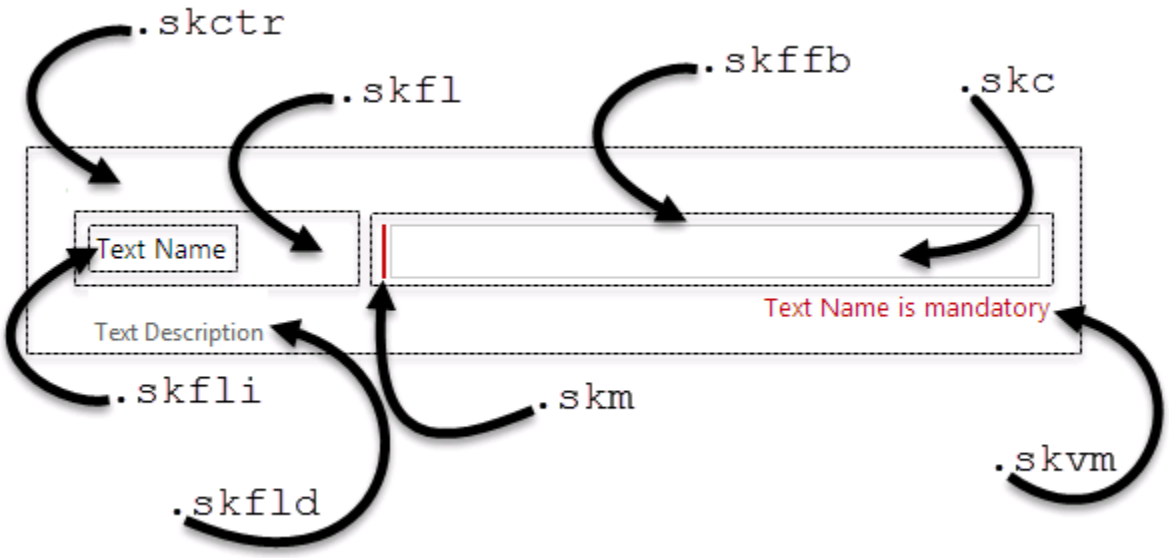
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
.Tabs1_skcn
{
 border: 1px solid blue !important;
}
```

Note: Suffix your CSS declarations with `!important` to ensure it takes precedence over normal declarations.

Other Controls

You can change style for HTML elements of controls other than **Form**, **Base Form**, **Panel Form**, and **Tabs** using CSS classes.



HTML Element	CSS Class
Control Container	.skctr
Control	.skc
Control Description	.skfld
Label Area	.skfl
Control Area	.skffb
Control Label	.skfli
Control Mandatory Indicator	.skm
Control Validation Message	.skvm

Prefix the **Custom Style Identifier** of a control with the **CSS Class** name to apply the style to the specific control. For example:

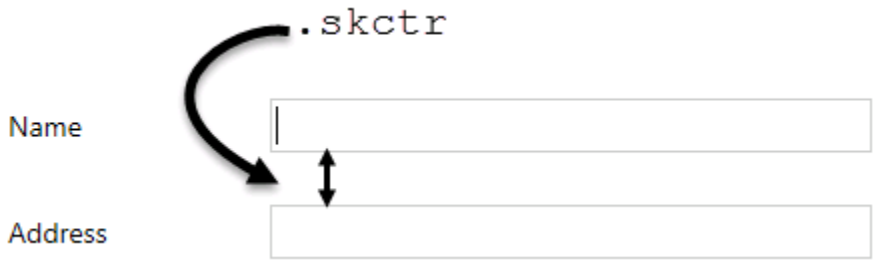
```
/* Text1 is the Custom Style Identifier of the Text control. */
.Text1_skvm
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example

```
/* Set the control label area to 100 pixels and use the remaining pixels for the control.
*/
/* Label Styles */
.skfl
{
 width: 100px;
}
/* Control Styles */
.skffb
{
 width: calc(100%-100px);
}
```

The default width and padding between the controls is 50px and 5px respectively. You can reduce the width and padding using CSS classes.




```
/* Decrease space from 50px to 20px and padding from 5px to 1px.*/
.skctr
{
   min-height: 20px;
   padding: 1px;
}
```

Note: Suffix your CSS declarations with `!important` to ensure it takes precedence over normal declarations.

Boundaries and Limits

Boundaries are absolute values that cannot be exceeded by design, whereas limits are maximum values that are recommended for optimal performance.

You need to know the boundaries and limits, and configure limits for your environment to achieve acceptable performance.

Test Environment

The boundaries and limits specified are determined by testing performance in mobile devices in the following intranet environment.

System	Operating System	Processor	Memory
AVEVA Work Tasks Engine	Microsoft Windows Server 2012 R2 Standard	Intel(R) Xeon(R) Silver 4110 CPU @ 2.10 GHz,	16 GB

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

System	Operating System	Processor	Memory
		2095 MHz	
AVEVA Work Tasks Client and Webserver	Microsoft Windows Server 2012 R2 Standard	Intel(R) Xeon(R) Silver 4110 CPU @ 2.10 GHz, 2095 MHz	16 GB
Database Server	Microsoft Windows Server 2012 R2 Standard	Intel(R) Xeon(R) Silver 4110 CPU @ 2.10 GHz, 2095 MHz	16 GB

Device	Operating System
Apple iPhone 8	iOS 13
Apple iPad	iOS 11
Samsung Galaxy S8	Android 8
UWP	Windows 10 (version number 1903)

Note:
- Many factors affect performance in a given environment (including configuration of systems running the server software, network configurations, maximum data transfer rates associated with different devices, other enterprise applications participating in the processes, and back-end database systems), and each factor may affect performance in a different area.
- If the limits specified do not work or are not applicable in your environment, then you will have to thoroughly test your environment to determine the optimal limits.
- Do not interpret the performance characteristics presented in this document as a benchmark measurement for supporting all computers. Only empirical testing on the target computer can provide accurate data.

Boundaries and Limits by Property

The boundaries and limits by property is as follows.

Property	Boundary	Limit
Name	30 characters	10 characters
XML Node	15 characters	5 characters
Enable Multi Select	75 work items	50 work items
Scripts	200 lines	100 lines

Note: The boundaries and limits recommended are based on the performance in mobile devices.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Control Classification

Controls are classified as simple, moderate, or complex based on data that can be bound to the control.

Simple	Moderate	Complex
Boolean	Attachment	Chart
Button	Check Box	Data Grid
Date	Drop-Down	Data Lookup
Date Time	Image	Hierarchical List
Email	List	
Hidden Field	Radio Button	
Hyperlink	Rich Text	
Invoke Workflow	Tab	
Label		
Number		
Password		
Slider		
Text		
Time		

Note:
- The number of records retrieved determines the complexity of the **Role Lookup** and **User Lookup** controls.
- The number and type of controls determines the complexity of the **Base Form** and **Panel Form** controls.
- The **Pop-Up** control acts as a normal form.
- The number and type of controls and also the data that persists back to the parent form determines the complexity of the **Pop-Up** Control.
- The **Pop-Up** control also does not support forms with **Pop-Up** controls.
- The data loaded determines the complexity of the **Embed Page** and **Report Viewer** controls.
- The data transfer between the server and the client is limited to 4 MB.

Boundaries and Limits by Control

The boundaries and limits by control is as follows.

Simple Controls

Control	Boundary	Limit
Date	1/1/1901 to 12/31/2099	1/1/1901 to 12/31/2099
Time	00:00:00	00:00:00

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Control	Boundary	Limit
	to 23:59:59	to 23:59:59
Number	-999999999999999.00 to 999999999999999.00	-999999999999999.00 to 999999999999999.00
Text	4096	255
Slider	15	5

Moderate Controls

Control	Boundary	Limit	Notes
Attachment	30 MB	5 MB	
Check Box	100 options	10 options	
Drop-Down	200 options	50 options	
Image	300 KB	50 KB	Image of type Portal Network Graphics (PNG) is recommended.
List	100 options	10 options	
Radio Button	100 options	10 options	
Rich Text	10 controls with 5000 characters per control	5 controls with 2500 characters per control	Images and tables are not considered.
Tab	15 controls per tab	10 controls per tab	
Tree	1000 child nodes	500 child nodes	

Complex Controls

Control	Boundary	Limit	Notes
Base Form	12 records with 5 controls per record	10 records with 5 controls per record	
Chart	10000 records	5000 records	
Data Grid	1000 records with 10	500 records with 5	The data transfer

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Control	Boundary	Limit	Notes
	records per page	records per page	between the server and the client is limited to 4 MB.
Data Lookup	10000 records with 50 records per lookup with autocomplete length of 5	5000 records with 50 records per lookup with autocomplete length of 3	The data transfer between the server and the client is limited to 4 MB.
Hierarchical List	20000 records with 50 records per page	10000 records with 50 records per page	
Panel Form	20 controls per panel with 5 columns	10 controls per panel with 3 columns	

Note:
- The boundaries and limits recommended are based on the performance in mobile devices.
- The **Embed Page** and **Report Viewer** controls are not considered for determining the boundaries and limits.
- The Form loading slows when Hierarchical List and Chart control is loaded using Web API Lookup.

Boundaries and Limits by Form

The boundaries and limits by form is as follows.

Form	Boundary	Limit	Notes
Form	Form with: • 40 simple controls • 30 moderate controls • 20 complex controls • 20 lines of script for 20 controls • 15 concurrent access	Form with: • 25 simple controls • 20 moderate controls • 15 complex controls • 20 lines of script for 15 controls • 10 concurrent access	Form loads within 5 seconds.

Note:
- The boundaries and limits recommended are based on the performance in mobile devices.
- The **Embed Page** and **Report Viewer** controls are not considered for determining the boundaries and limits.
- The combination of the controls determines the complexity of the form.
- The number of container controls (such as Base Form, Panel Form, Tabs, and so on) affects the performance of the form.
- A maximum of 5 levels of container controls (such as Base Form, Panel Form, Tabs, and so on) per form is recommended.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Best Practices

We recommend the following best practices when designing your forms.

- Ensure you use **Containers** appropriately for aesthetic and performance benefits.
  - Use containers to group controls logically.
  - Use **Base Form** to create a set of repeatable controls at run-time.
  - Use **Panel Form** to create layouts without generating XML nodes.
  - Use **Tabs** to group associated controls on the same page with set of related controls separated into sections (**Tab**).
  - Use limited number of **Tabs** and **Tab** to load tabs and forms faster.
- Ensure you use **Controls** appropriately for performance benefits.
  - Use minimum viable controls per form and avoid overloading controls to load controls and forms faster.
  - Use small file size attachments with **Attachment** to upload attachments faster.
  - Use small file size images with **Image** to load images faster.
  - Use predefined options either through the property or through the script instead of populating the options from the server when the number of options is limited with **Check Box, Drop-Down, Radio Button**, and **List** to load options faster.
  - Use **Drop-Down** only the number of options is limited with **Data Lookup** to load lookups faster.
  - Use **Pop-Up** to include sub-forms to load forms faster.
  - Use **Pop-Up** to open forms in a new browser window to improve forms performance.
  - Use lookup definitions configured for limited result set with **Data Lookup** to load lookups faster.
  - Use 10 records and autocomplete length of 3 or more with **Data Lookup** to improve data retrieval speed and to load lookups faster.
  - Use **Extended Information** only when required and with limited fields with **Data Lookup** and **Data Grid** to improve data retrieval speed and to load lookups and grids faster.
  - Use **Persist Display Value** with **Data Lookup** to cache the display value in the edit mode of forms to load lookups faster.
  - Use parameterized lookups and grids for limited result set with **Data Lookup** and **Data Grid** to improve data retrieval speed and to load lookups and grids faster.
  - Use grid definitions configured for limited result set with **Data Grid** to load grids faster.
  - Use only the required fields to display and persist with **Data Grid** to improve data retrieval speed and to reduce the size of the instance XML.
  - Use selection (highlight) instead of confirmation (check) with **Data Grid** to reduce the size of the instance XML.
  - Use selection of record instead of with **Data Grid** to reduce the size of the instance XML.
- Ensure you use **Forms** appropriately for aesthetic and performance benefits.
  - Use forms with minimal controls and add or remove controls progressively as the workflow progresses to load forms faster and to improve the usability of forms.
  - Use short **Name** and **XML Node** for properties of controls to reduce the size of the instance XML and to improve the data transfer speed between servers and clients.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

  - Use short **Description** and only if required for properties of controls to reduce the size of the definition XML.
  - Use optimized JavaScript code without unnecessary closures to improve script execution.
  - Use asynchronous calls instead of synchronous calls with **On Data Change** event.
  - Use well-formed JSON instead of XML when exchanging data between servers and clients, and limit the return information to only the required information for performance benefits.
  - Use GET method as appropriate with AJAX calls in scripts and cache the relevant APIs to avoid reloading the APIs on every request to improve script execution.
  - Use queue to handle multiple AJAX calls in scripts and request at once to improve script execution.
- Ensure you use Content Security Policy appropriately. Loading of the External URL for Web API, Embed Page, Images, and so on depends upon the Content Security Policy for the Enterprise Console Web Application.
  - Modify and define the Content Security Policy as per the user requirement. For more details, refer to Content Security Policy documentation on the W3C website.
  - Specify the URLs based on the same domain in the Form Controls. For images, you can use data-URLs.

Form Controls

Controls are the basic components of a form to view, enter, or submit data. Controls are grouped as basic controls and container controls. Container controls are used to place basic controls, and container controls, on a container control to create a complex form section or layout.

Note: The **Print** button in the Form does not work if there are angular controls like Hierarchical list, Chart, and Slider present in the Form.

Containers

Container controls are used to place basic controls, and container controls, on a container control to create a complex form section or layout.

Form

Use **Form** as a basic container to place other containers and controls.

Properties

The properties for the form are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the form. This value is also used for the title of the form.
XML Node	XML node mapped to the form. Use this value to map

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	a list to a form and a form to a workflow. The default value is the value for the **Name** property without the spaces. **Note:** If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the form.
Show Save Button	Show or hide the save button for the form. • Select **Yes** to show the save button for the form. • Select **No** to hide the save button for the form. The default value is **Yes**.
Save Button Text	Text for the save button of the form. The default value is **Finish**.

Appearance

Property	Description
Label Position	Position of the label for the child control. • Select **Default** to allow the system to align the label for the child control. The system aligns the label to the left of the child control. • Select **Left** to align the label to the left of the child control. • Select **Top** to align the label to the top of the child control. • Select **Right** to align the label to the right of the child control. • Select **Bottom** to align the label to the bottom of the child control. • Select **Suppress** to hide the label for the child control. The default value is **Default**. **Note:** - This property is applicable only if the **Label Position** property for the child control is set to **Default**. - For example, if the **Label Position** property for the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	child control is set to **Default** and the **Label Position** property for the **Base Form** is set to **Top**, then the label for the child control is aligned to the top of the child control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Auto-size Window	Allow or deny the system to automatically size the window to the available screen size on form load. • Select **Yes** to allow the system to automatically size the window to the available screen size on form load. • Select **No** to deny the system to automatically size the window to the available screen size on form load. The default value is **No**. **Note**: - When the **Auto-size Window** is set as Yes, the following forms appear as screen size on form load: - Preview form - Fill form - Invoke form - Reports - Submitted form

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	- Pop-up form, where the form is set as Pop-up - For a form set as Inline form, the **Auto-size Window** property not applicable. - When the Auto-size Window is set to *Yes*, and **Maximize the window on load** of **Invoke Form activity** is set to *No*, the **Auto-size** Window value overrides the **Invoke Form activity** value. - While creating a list item, the **Auto-size Window** property is not applicable even if the **Auto-size Window** property for the list associated form is set to *Yes*.
Desktop View Only	Turn off the UI adjustment based on the screen width. • If the value is set to **Yes**, the view generated for the Form is not responsive, even if the value of the **Adapt UI** property at the container level is set to **Yes**. • If the value is set to **No**, the view generated for the Form is responsive. You can change the responsiveness at the container level using the **Adapt UI** property. The default value is **No**.
Delay the Loading of Containers	Display contents when the container controls are accessed. • Select **Yes** to display the content of the container controls only when the controls are accessed. • Select **No** to always display the content of the container controls. Consider a scenario where **Delay the Loading of Containers** option is enabled and a **Data Grid** with **Behavior** property is set as **Data Source**. When a script is used for the **Options** property to add options for **Drop-down** control, data will not be loaded in the drop-down control until the data grid is loaded. For more information on **Options** property, see

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Options. **Note:** - This property is applicable only if the **Desktop View Only** property is set to **Yes**. - This property is not supported in AVEVA Work Tasks Pro App.
Custom Style Sheet	Name of the custom style sheet. For example, **FormCustomPath:MyCustomStyleSheet** for style sheet located at **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/css/MyCustomStyleSheet.css** **Note**: - If the style sheet name is tagged with **FormCustomPath:**, the location is **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/css/MyCustomStyleSheet.css** - If the style sheet name is tagged with **FormCustomThemePath:**, the location is **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/Themes/<ThemeName>/css/MyCustomStyleSheet.css** - If the stylesheet name is not tagged with **FormCustomPath:** or **FormCustomThemePath:**, the location of style sheet of earlier version will be considered i.e. **<Installpath>/Web/BPMUITemplates/Default/Themes/<ThemeName>/NextGenForms/css/custom/ MyCustomStyleSheet.css**. - If **Theme** is not applied and if the style sheet name is tagged with **FormCustomPath:** or **FormCustomThemePath:**, the location is **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/css/MyCustomStyleSheet.css** - If **Theme** is applied and if the style sheet name is tagged with **FormCustomPath:** or **FormCustomThemePath:**, the location is **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/Themes/<ThemeName>/css/**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	MyCustomStyleSheet.css - For earlier versions of the product, the location for style sheet is **<Installpath>/Web/BPMUITemplates/Default/Themes/<ThemeName>/NextGenForms/css/custom/ MyCustomStyleSheet.css** - For AVEVA Work Tasks Pro, mapping of themes is not considered. For information about using custom style sheets, see Custom Styles.

Advanced

Property	Description
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Submit Form with Enter Key	Allow or deny a user to submit a form by pressing the Enter key. • Select **Yes** to allow a user to submit the form by pressing the Enter key. • Select **No** to deny a user to submit the form by pressing the Enter key. The default value is **Yes**. **Note:** - This property is applicable only if the **Button Type** property of **Button** control is set to **Submit**. -If there are multiple Submit buttons in the form, then the submit action with Enter key is performed on the first Submit button. - If the Submit button is the first control in the form, then point the cursor to the input controls and then press the **Enter** key to submit the form.
Forms Parameters	Form-level variables for use in workflows. For information about using **Forms Parameters**, see Forms Parameters.
MQTT Topics	Access external data through MQTT for use in forms. • Click the **Add** button to add MQTT topic. Then

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	click the **Settings** button to configure the topic. For more information, see MQTT Topic Configuration.
Enable Action Bar	Enable or disable Action bar during run-time. • Select the **No** check box if you do not want the Action bar to be enabled. • Select the **Yes** check box if you want the action bar to be enabled. When you select **Yes**, the **Action Bar** section appears. For information about enabling action bar, see Action Bar.

Scripts

Property	Description
On Form Load	Run a script when a form is loaded. For more information, see scripting guidelines for On Form Load event

Action Bar

An Action bar contains a group of icons, which are used to perform certain actions. The Action bar has the action items defined by the user during the design time and it is available to the user while using the form during run-time.

Note:
- Action bar is supported only on the web browsers. However, it is not supported in AVEVA Work Tasks Pro App.
- When the window is in the Responsive mode (< 800 pixels in width) in the Web browsers, the Action bar is not displayed.
- Action bar supports only one level of child Action Items.
- You can add multiple child Action Items to an Action Item.

To enable Action bar

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. Right-click on the form for which you want to enable Action bar, and then click **Design Form**. The **Forms Designer** page appears.
3. On the Action tool bar in the **Forms Designer**, click **Properties**. The **Properties** dialog box appears.
4. Click the **Advanced** tab.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Click **Yes** for the **Enable Action Bar** property.

   The **Action Bar** section appears.

6. Click the **+** icon.

   A new action item is added.

7. Click the **Properties** icon.

   The **Action Bar Item - Properties** window appears.

8. In the **Basic** tab, specify the following details:

Property	Description
Name	Name that identifies the Action Item. This value is also used for the label of the Action Item.
Image Path	Path to the image for the Action Item. The image path is specified as: • A URL starting with http:// or https://. • An image path tag. For information about using image path tags, see Image Path Tags in Image. • A path to the BPMUITemplates folder as BPMUITemplates\Default\Images\. If you want to use an image and a hover image for an Action Item, then add the respective images in the *<NextGenForms/custom/Themes/<ThemeName>/images* path. **Note**: - If the image name is tagged with **FormCustomPath:**, the location is **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/images** - If the image name is tagged with **FormCustomThemePath:**, the location is **<Installpath>/Web/BPMUITemplates/Default/<NextGenForms/custom/Themes/<ThemeName>/images** - If the image name is not tagged with **FormCustomPath:** or **FormCustomThemePath:**, the location of image of earlier version will be considered i.e. **<Installpath>/Web/BPMUITemplates/Default/Themes/<ThemeName>/NextGenForms/images** folder. - For AVEVA Work Tasks Pro, mapping of themes is not considered.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	For example: If the image **ClearAll.png** is available in *<NextGenForms/custom/Themes/<ThemeName>/images*, then ensure that you add the hover image **ClearAll-Hover.png** (with **-Hover** suffixed to the image name) in the same path. **Note:** - Ensure the size of the image (icon) is 25x25 pixels. - You can use relative paths as follows: -- Use . for the current folder where the html page resides. -- Use .. for the folder above the current folder where the html page resides.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description

Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet.
	If you want to use an image and a hover image for an Action Item using a CSS, then add the respective images in the *<NextGenForms/custom/Themes/<ThemeName>/images* path and add the CSS file in the *<NextGenForms/custom/Themes/<ThemeName>/css* path.
	The CSS can be defined as:
	`.customItem .icon`
	`{`
	`background-image: url("../../images/ClearAll.png");`
	`}`
	`.customItem:hover .icon`
	`{`
	`background-image: url("../../images/ClearAll-Hover.png");`
	`}`
	`.customItem[disabled='disabled']:hover .icon`
	`{`
	`background-image: url("../../images/ClearAll.png");`
	`}`
	For the Action Item, enter the CSS class name as **customItem**.
	For information about using custom style sheets, see Custom Styles.
	Note:
	- The value must start with an alphabet and can contain only alphanumeric characters.
	- For AVEVA Work Tasks Pro, mapping of themes is not considered.

Tooltip	Message that appears when the pointer hovers over the Action Item.
Enable	Enable or disable the Action Item. • Select **Yes** to enable the Action Item. • Select **No** to disable the Action Item. The default value is **Yes**. For more information about enabling or disabling the Action Item via script, see control.topLevelForm.actionbar.getActionItemByName().

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Visible	Show or hide the Action Item. • Select **Yes** to show the Action Item. • Select **No** to hide the Action Item. The default value is **Yes**. For more information about changing the Visible property of the Action Item via script, see control.topLevelForm.actionbar.getActionItemByName().
Action Type	Action for the Action Item. • Select **Link to Existing Control** to perform the respective action, when the Action Item is clicked. • Select **Specify Script** to enter a script that must be executed, when the Action Item is clicked. To link to an existing control. a. Select **Link to Existing Control**. The **Existing Control** drop-down list appears with the controls (of **Button**, **Popup**, **Invoke Workflow** types**)** present in the **Forms Designer** including the ones present in the **Embed Form Controls**. b. In the **Existing Control** drop-down list, select the required control to be executed, when the Action Item is clicked. The Action Item is linked to the selected control.

9. Click **Apply**.

The properties are applied to the Action Item.

10. Repeat steps 6 - 9 to add more Action Items.

Scripts

For information about using scripts, see scripting guidelines for Action Bar control.

Applying styles for Action Bar

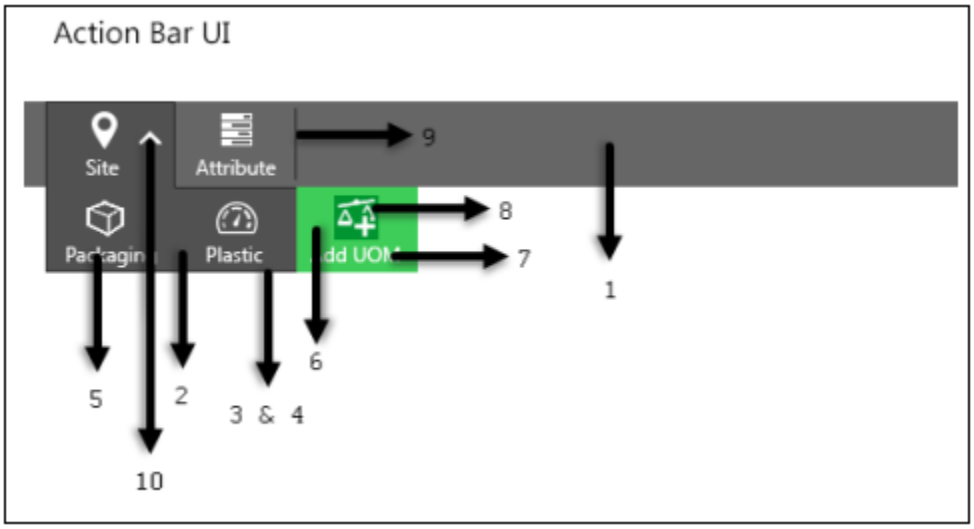
You can apply the following styles for the Action bar:

1. Background color for the action bar.
2. Background color for the parent and child Action items in expanded mode.
3. Borders for the expanded child Action item.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Borders for the expanded parent Action item.
5. Text color for the Action item.
6. Background color for the Action item on hover.
7. Text color for the Action item on hover.
8. Background color for the icon in the Action item on hover.
9. Action item separator.
10. Action item down arrow, Action item up arrow.

The following image represents the Action bar with custom styles applied. The numbers in the image represent the styles for the Action bar mentioned above.



Add the following CSS file in the *<NextGenForms/custom/Themes/<ThemeName>/css* path. The CSS can be defined as follows:

```
/*Background color for the action bar*/
.actionBarWrapper .actionBar
{
   background-color: #666666;
}
/*Background color for the parent and child Action items in expanded mode*/
.actionBarWrapper .actionBar .subMenu, .actionBarWrapper .actionBar .menuItem.selected,
.actionBarWrapper .actionBar .menuItem.subSelected, .actionBarWrapper .actionBar
.subSubMenu
{
   background-color: #515151;
}
/*Borders for the expanded parent Action item*/
.actionBarWrapper .actionBar .menuItem.selected, .actionBarWrapper .actionBar
.menuItem.subSelected
{
   box-shadow: inset -1px 0 0 0 #7F7F7F, inset 1px 1px 0 0 #7F7F7F;
   /*box-shadow: none !important;*/
}
/*Borders for the expanded child Action item*/
.actionBarWrapper .actionBar .subMenu, .actionBarWrapper .actionBar .subSubMenu
{
   box-shadow: inset 0 0 0 1px #7F7F7F;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
/*Text color for the Action item*/
.actionBarWrapper .actionBar .menuItem .content .name
{
   color: #ffffff;
}
/*Text color for the Action item on hover*/
.actionBarWrapper .actionBar .menuItem:hover .content .name
{
   color: #ffffff;
}
/*Background color for the Action item on hover*/
.actionBarWrapper .actionBar .menuItem:not(.menuItemDisabled):hover
{
   background-color: #3dcd58;
}
/*Background color for the icon in the Action item on hover*/
.actionBarWrapper .actionBar .menuItem:hover .icon
{
   background-color: #009530;
}
/*Action item separator*/
.actionBarWrapper .actionBar .separator
{
   background: #7F7F7F;
}
/*Action item down arrow*/
.actionBarWrapper .actionBar .menuItem .downArrow
{
   border-color: rgba(0, 0, 0, 0) #FFFFFF #FFFFFF rgba(0, 0, 0, 0);
}
/*Action item up arrow*/
.actionBarWrapper .actionBar .menuItem.selected .downArrow
{
   border-color: #FFFFFF rgba(0, 0, 0, 0) rgba(0, 0, 0, 0) #FFFFFF;
}
```

MQTT Topic Configuration

To add an MQTT topic and link it to an existing control or specify a script

1. In the MQTT Topic Configuration page, specify the following details:

Property	Description
Topic	Enter the path of the topic.
MQTT Connection	Select the MQTT Connection. **Note**: When MQTT Connection is changed, the latest published message of the topic is displayed in the **Subscribed Value** text box.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Subscribed Value	The latest published message is displayed in the **Subscribed Value** text box. The **Subscribed Value** text box is empty when the published message is not available. Click **Refresh** button to retrieve the latest published message. **Note**: When the **Topic** and the **MQTT Connection** is changed, **Refresh** button is disabled for 5 seconds. It is enabled after the subscribed value is received.
Data Type	Select the data type of the topic value. For more information on available data types, see Data Types in MQTT Topic Configuration.
JSON Path	Specify the JSON Path. This is required only for topic message containing valid JSON string and when publisher sends message as JSON object. You must specify the path to access the node in the JSON string. For examples, see Specify JSON Path.
Array	Select the **Array** option as **Yes** if the published message is an array. **Note**: The format for the supported array values is <element1>, <element2>, ..., <elementN>
Separator	Specify the character comma (,) to separate the array. **Note**: Only one character can be entered in the Separator box.
Element	Enter the position of the element in the array to be extracted. The minimum value is 1 and the maximum value is 999999999999999. **Note**: When the **Array** option is **Yes** and an array element position is not specified, the value displayed in **Extracted Value** box is based on the value selected in **Action Type** box. - If you select **Link to Existing Control**, then the first element value of the array is displayed. - If you select **Specify Script**, the whole array value is displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Action Type	Two options are available for Action Type: • **Link to Existing Control** In the **Existing Control** list, select a control (available on the form). • **Specify Script** Click the **Script Editor** button to specify the script in the Script Editor page.
Existing Control	Select the existing controls available in the form. The controls which have the value is listed like TextInput, Number, etc.
Action Script	Click the **Script Editor** button. Specify the script that is to be executed. For example, `control.findByXmlNode("Textinput1").value= currentValue;` Click **Save & Close.**The script is executed when message is arrived.
Extracted Value	Displays the value extracted from the Subscribed Value. The Extracted value is based on **Topic**, **MQTT Connection**, **Data Type**, **JSON path**, and **Array.** • If Array is **Yes**, then the Separator and Element is considered. • If the **Action Type** is **Link to Existing Control**, then the Extracted value is shown based on the Existing Control type. • If the **Action Type** is **Specify Script**, then the currentValue type is based on the selected Data Type .

2. Click **Save & Continue**.

The **Forms** properties page appears.

Note: After you save the settings, the setting details are displayed in the **Topic** box. For example, the displayed information is:
- <*Control name ~ topic*>, when the Action type is **Link To Existing Control**, and <*Control not linked*> when control is not linked.
- <*Script ~ topic*>, when the Action type is **Script**.
- MQTT topics are not supported in AVEVA Work Tasks Pro.

For information about MQTT Connections, see **MQTT Connections** in the Developer Guide.

Data Types in MQTT Topic Configuration

The available data types you can configure in MQTT topic are:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **String**

  If the string value is assigned to the TextBox control, then the value is converted to string.

  If there is any mismatch of format, the error is logged in Browser Console as:

  Error occurred during String type conversion for the topic '<@topicname@>'.

- **Boolean**

  If the true value is assigned to the Boolean control, True is enabled.

  If the false value is assigned to the Boolean control, False is enabled.

  The supported Boolean values are:

  1. True values:

     - true (is not case sensitive and is a Boolean flag)

     - 1 (can be numeric or in string format)

     - yes (is not case sensitive and should be passed only in string format)

     - y (is not case sensitive)

  2. False values:

     - false (is not case sensitive and is a Boolean flag)

     - 0 (can be numeric or in string format)

     - no (is not case sensitive and should be passed only in string format)

     - n (is not case sensitive)

  If the Boolean values are assigned to the TextBox control, the value is converted to string.

  If the Boolean values are assigned to the Number control other than 1 and 0; and if there is any mismatch of format, the error is logged in the Browser Console as:

  Error occurred during Boolean type conversion for the topic '<@topicname@>'.

- **Integer**

  If the integer value is assigned to the Number control, the value takes 15 digits as specified by the property's boundary limit.

  It is recommended to have the precision set to the Number control as per the source.

  If the integer value is assigned to the TextBox control, the value is converted to strings.

  If there is any mismatch of format, the error is logged in Browser Console as:

  Error occured during Number type conversion for the topic '<@topicname@>'.

- **Float**

  If the float value is assigned to the Number control, the value displayed is as specified by the property's boundary limit.

  It is recommended to have the precision set to the Number control as per the source.

  If the float value is assigned to the TextBox control, the value is converted to strings.

  Only decimal separator is allowed to represent the decimal number.

  If there is any mismatch of format, the error is logged in Browser Console as:

  Error occurred during Number type conversion for the topic '<@topicname@>'.

- **Double**

  If the Double value is assigned to the Number control, the value displayed is as specified by the property's

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

boundary limit.

It is recommended to have the precision set to the Number control as per the source.

If the Double value is assigned to the TextBox control, the value is converted to strings.

Only decimal separator is allowed to represent the decimal number.

If there is any mismatch of format, the error is logged in Browser Console as:

E*rror occurred during Number type conversion for the topic '<@topicname@>'*.

Note: When the selected data type is Float/Double and the Linked control is a number (without decimal), then the extracted value is displayed with precision.

- **DateTime**

  If the DateTime value is assigned to the ‘DateTime’ control, the value displayed is as specified by the property's boundary limit and local browser specific culture.

  If the DateTime value is assigned to the 'TextBox' control, the value displayed is string values. For example, *Mon Mar 11 2019 19:23:35 GMT+0530 (India Standard Time)*.

  The supported DateTime value is only in string format and should be in en-US format.

  The supported DateTime formats are:

  “MM/dd/yyyy hh:mm:ss.fff tt",

  "MM/dd/yyyy hh:mm:ss tt”

  If there is any mismatch of format, the error is logged in Browser Console as:

  Error occurred during DateTime format conversion for the topic '<@topicname@>'.

- **ElapsedTime**

  If the ElapsedTime value is assigned to the TextBox control, the time is displayed as HH:mm:ss.fff (where HH represents 24 hour format, mm represents minutes, ss represents seconds, and fff represents milliseconds).

  The ElapsedTime datatype should be the decimal number. For example, the decimal part is milli seconds(.fff) and the integer part is converted to HH:mm:ss.

  If there is any mismatch of format, the error is logged in Browser Console as:

  Error occurred during ElapsedTime format conversion for the topic '<@topicname@>'.

Specify JSON Path

To access **b** in the JSON string:

```
{"a": {"b":"1"}}
```

specify as follows:

```
a.b
OR
a["b"]
OR
["a"].b
OR
["a"]["b"]
```

To access the second item of the array **b** that contains value **2** in the JSON string:

```
{
"a": [
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"b": "1"
},
{
"b": "2"
},
{
"b": "3"
}
]
}
```

specify as follows:

```
a.b[1]
OR
a["b"][1]
OR
["a"].b[1]
OR
["a"]["b"][1]
```

Base Form

Use **Base Form** control to manage controls placed on the record (child controls) in your form.

You can place any number of controls on a record, and add and delete records for the control. At run-time, you can click **Add Record** to repeat a record, and click **Delete Record** to remove a record. You can view the values of the child controls in the **Summary** of the **Base Form** only when the **Repeating** property for the child controls is set to **True**, and the form is viewed on a small form factor mobile device. Maximum three control values are considered in the **Summary** view.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. The default value is the value for the **Name** property without the spaces. **Note:** If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.

Appearance

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**. **Note:** If the value is **No**, then the child controls will be hidden even when the **Visible** property for the child controls is set to **Yes**.
Label Position	Position of the label for the child control. • Select **Default** to allow the system to align the label for the child control. The system aligns the label to the left of the child control. • Select **Left** to align the label to the left of the child control. • Select **Top** to align the label to the top of the child control. • Select **Right** to align the label to the right of the child control. • Select **Bottom** to align the label to the bottom of the child control. • Select **Suppress** to hide the label for the child control. The default value is **Default**. **Note:** - This property is applicable only if the **Label Position** property for the child control is set to **Default**. - For example, if the **Label Position** property for the child control is set to **Default** and the **Label Position** property for the **Base Form** is set to **Top**, then the label for the child control is aligned to the top of the child control. -The labels for all the controls are suppressed except - Data Grid, E-signature, Label, Embed Form, Embed Page, Report Viewer, Tab , Panel, and Base Form.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Display Mode	Show or hide the tabular view for the control. • Select Form to hide the tabular view for the control. • Select Grid to show the tabular view for the control. The default value is **Grid**.
Show Column Name	Show or hide the column name, when the **Display Mode** is set to **Grid**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Yes** to show the column name. • Select **No** to hide the column name. The default value is **Yes**.
Records Caption	Suffix for the add and delete links. The default suffix is **Record**. **Note:** The suffix is limited to 50 characters.
Record Position	Position of the records added at run-time for the control. • Select Top to position the records, added at run-time, at the top of the existing records for the control. • Select **Bottom** to position the records, added at run-time, at the bottom of the existing records for the control. The default value is **Bottom**.
Image Path	Path to the image for the control. The image path is specified as: • A URL starting with **http://** or **https://**. • An image path tag. For information about using image path tags, see *Image Path Tags* in Image. • A path to the **BPMUITemplates** folder as **BPMUITemplates\Default\Images**. **Note:** You can use relative paths as follows: - Use **.** for the current folder where the html page resides. - Use **..** for the folder above the current folder where the html page resides. - Base form image will be displayed when the screen is in Mobile View (< 800px)
Adapt UI	Change the user interface of the control based on the display width. • Select **Yes** to allow the system to change the user interface of the control based on the display

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	width. The **Web View** is rendered for the control when the screen width is greater than or equal to 800 pixels. Otherwise, the **Mobile View** is rendered for the control. • Select **No** to deny the system to change the user interface of the control based on the display width. In this case, the **Web View** is rendered for the control. The default value is **Yes**. **Note:** The responsiveness of the container depends on the value set for the **Desktop View Only** property at the Form level.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Show Confirmation Message	Show or hide the confirmation message for the control. • Select **Yes** to show the confirmation message for the control. • Select **No** to hide the confirmation message for the control. The default value is **No**.
Confirmation Message Text	Confirmation message for the control. **Note:** This property is applicable only when the **Show Confirmation Message** property is set to **Yes**.
Repeating	Allow or deny a user to repeat records for the control. • Select **True** to allow a user to repeat records for the control. • Select **False** to deny a user to repeat records for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Not Set** to allow the system to set the value for the control. The system allows a user to repeat records for the control. The default value is **Not Set**.
Minimum Records	Minimum records for the control. The default value is **0**. **Note:** This property is applicable only when the **Repeating** property is set to **True**.
Maximum Records	Maximum records for the control. Enter 0 for infinite records. The default value is **0**. **Note:** This property is applicable only when the **Repeating** property is set to **True**.

Advanced

Property	Description
On Record Add	Run a script when a record is added. For more information, see scripting guidelines for On Record Add property.
On Record Delete	Run a script when a record is deleted. For more information, see scripting guidelines for On Record Delete property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Behavior	Manage records and edit values at run-time. • Select **Editable** to allow a user to add or delete records for the control, and to edit values of the child controls. • Select **Read-Only** to deny a user to add or delete records for the control, and deny to edit values of the child controls. • Select **Allow Only Add** to allow a user to add records for the control, but deny to edit values of the existing child controls. That is when the form

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	is loaded, the existing records cannot be edited or deleted, but new records can be added, edited, and deleted till the form is submitted. The default value is **Editable**.
Default Number of Record	Number of records added at run-time for the control. The default value is **1**.
Fast Append	Add records in the **Grid** view at run-time by pressing the **Tab** key from the last focused control of the record. • Select **Yes** to allow adding records in the **Grid** view at run-time by pressing the Tab key from the last focused control of the record. • Select **No** to deny adding records in the **Grid** view at run-time by pressing the Tab key from the last focused control of the record. The default value is **No**. **Note:** This property is not applicable if: - The **Display Mode** is set to Form or the Repeating property is set to **False**. - The last focused control is a control on a child **Base Form** or a custom control. - The form is viewed on a small form factor mobile device.

Scripts

For information about using scripts, see scripting guidelines for Base Form control.

Note:
- Description of controls placed on **Base Form** with **Display Mode** set as **Grid** is not displayed.
- Name set through scripts for controls placed on **Base Form** with **Display Mode** set as **Grid** is not displayed.
- Label of controls placed on **Base Form** is truncated if there are too many controls or if the label is too long.
- When a **Base Form** with the **Default Number of Records** set in a form, associated to a workflow with an **Invoke Form** activity to the same form, is used in **Fill Form**, and if all records for the **Base Form** are deleted while submitting the **Fill Form**, then, in the **Invoke Form** activity, the **Base Form** will continue to display the **Default Number of Records**.
- In a Form with Delay Loading enabled, **Chart**, **Slider**, **Card**, and **Hierarchical List** controls inside a container are not supported.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Panel Form

Use **Panel Form** control to group controls placed on the control (child controls) in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. **Note:** If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**. **Note:** If the value is **No**, then the child controls will be hidden even when the **Visible** property for the child controls is set to **Yes**.
Label Position	Position of the label for the child control. • Select **Default** to allow the system to align the label for the child control. The system aligns the label to the left of the child control. • Select **Left** to align the label to the left of the child control. • Select **Top** to align the label to the top of the child control. • Select **Right** to align the label to the right of the child control. • Select **Bottom** to align the label to the bottom of

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	the child control. • Select **Suppress** to hide the label for the child control. The default value is **Default**. **Note:** - This property is applicable only if the **Label Position** property for the child control is set to **Default**. - For example, if the **Label Position** property for the child control is set to **Default** and the **Label Position** property for the **Panel Form** is set to **Top**, then the label for the child control is aligned to the top of the child control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Consider Available Height	Resize the height of the control or container based on the available height. • Select **Yes** to resize the height based on available height. • Select **No** to retain the space. The default value is **No**. **Note:** - The Consider Available Height property is not applicable for the following: -Designer view -Print form -Panel form with horizontal tab order - If this property is set to **Yes** and a value is specified in the **Height** property, minimum height is considered. - This property is applicable only in web view. - When more than one control or container is set with **Consider Available Height** at the same level of container or form, then the available space is distributed evenly between those controls or containers. For example, consider a Form which contains two Panel forms with **Consider Available Height** set to **Yes**, where Panel form 1 has two TextInput controls and Panel form 2 has one TextInput control. The Panel form 1's height is more than Panel form 2's height as Panel form 1 contains two TextInput controls and the available empty space in the form is evenly distributed between the two Panel forms.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Show Panel Details	Show or hide the title, description, and margin for the control. • Select **Yes** to show the title, description, and margin for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **No** to hide the title, description, and margin for the control. The default value is **Yes**.
Columns	Number of columns for the control. Enter a value from 1 to 10. **Note:** - Decreasing the number of columns will delete that many last columns with all its controls. For example, decreasing the number of columns from 5 to 3 would delete the last 2 columns with all its controls. - The columns with its controls deleted cannot be restored.
Tab Order	Tab order for the child controls. • Select **Horizontal** to set the tab order one after the other for the child controls. • Select **Vertical** to set the tab order one below the other for the child controls. The default value is **Vertical**.
Image Path	Path to the image for the control. The image path is specified as: • A URL starting with **http://** or **https://**. • An image path tag. For information about using image path tags, see *Image Path Tags* in Image. • A path to the **BPMUITemplates** folder as **BPMUITemplates\Default\Images**. **Note:** You can use relative paths as follows: - Use **.** for the current folder where the html page resides. - Use **..** for the folder above the current folder where the html page resides.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Adapt UI	Change the user interface of the control based on the display width. • Select **Yes** to allow the system to change the user interface of the control based on the display width. The **Web View** is rendered for the control when the screen width is greater than or equal to 800 pixels. Otherwise, the **Mobile View** is rendered for the control. • Select **No** to deny the system to change the user interface of the control based on the display width. In this case, the **Web View** is rendered for the control. The default value is **Yes**. **Note:** The responsiveness of the container depends on the value set for the **Desktop View Only** property at the Form level.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Behavior	Edit values at run-time. • Select **Editable** to allow a user to edit values of the child controls. • Select **Read-Only** to deny a user to edit values of the child controls. The default value is **Editable**.

Scripts

For information about using scripts, see scripting guidelines for Panel Form control.

Note: Embed Form is not supported in **Panel Form** with **Tab Order** set to **Horizontal**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Tabs

Use **Tabs** control to group controls placed on the tab in your form. By default, a Tab control is added to the control.

The Tabs container supports only 10 tabs. But in each tab, you can place any number of tab controls and you can also add or delete these controls.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
Description	Description for the control.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**. **Note:** If the value is **No**, then the child controls will be hidden even when the **Visible** property for the child controls is set to **Yes**.
Label Position	Position of the label for the child control. • Select **Default** to allow the system to align the label for the child control. The system aligns the label to the left of the child control. • Select **Left** to align the label to the left of the child control. • Select **Top** to align the label to the top of the child control. • Select **Right** to align the label to the right of the child control. • Select **Bottom** to align the label to the bottom of the child control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Suppress** to hide the label for the child control. The default value is **Default**. **Note:** - This property is applicable only if the **Label Position** property for the child control is set to **Default**. - For example, if the **Label Position** property for the child control is set to **Default** and the **Label Position** property for the **Tab** is set to **Top**, then the label for the child control is aligned to the top of the child control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Consider Available Height	Resize the height of the control or container based on the available height. • Select **Yes** to resize the height based on available height. • Select **No** to retain the space. The default value is **No**. **Note:** - The Consider Available Height property is not applicable for the following: -Designer view -Print form

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	-Panel form with horizontal tab order - If this property is set to **Yes** and a value is specified in the **Height** property, minimum height is considered. - This property is applicable only in web view. - When more than one control or container is set with **Consider Available Height** at the same level of container or form, then the available space is distributed evenly between those controls or containers. For example, consider a Form which contains two Panel forms with **Consider Available Height** set to **Yes**, where Panel form 1 has two TextInput controls and Panel form 2 has one TextInput control. The Panel form 1's height is more than Panel form 2's height as Panel form 1 contains two TextInput controls and the available empty space in the form is evenly distributed between the two Panel forms.
Default Tab Index	The default tab index for the control.
Show Border and Margin	Show or hide the border and margin for the control. • Select **Yes** to show the border and margin for the control. • Select **No** to hide the border and margin for the control. The default value is **Yes**.
Image Path	Path to the image for the control. The image path is specified as: • A URL starting with **http://** or **https://**. • An image path tag. For information about using image path tags, see *Image Path Tags* in Image. • A path to the **BPMUITemplates** folder as **BPMUITemplates\Default\Images**. **Note:** You can use relative paths as follows: - Use **.** for the current folder where the html page resides.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	- Use **..** for the folder above the current folder where the html page resides.
Adapt UI	Change the user interface of the control based on the display width. • Select **Yes** to allow the system to change the user interface of the control based on the display width. The **Web View** is rendered for the control when the screen width is greater than or equal to 800 pixels. Otherwise, the **Mobile View** is rendered for the control. • Select **No** to deny the system to change the user interface of the control based on the display width. In this case, the **Web View** is rendered for the control. The default value is **Yes**. **Note:** The responsiveness of the container depends on the value set for the **Desktop View Only** property at the Form level.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
On Tab Index Change	Run a script when the value of the tab index changes, that is when the selected tab changes. For more information, see scripting guidelines for On Tab Index Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.

Scripts

For information about using scripts, see scripting guidelines for Tabs property.

Keyboard-based Navigation

When the focus is on the Tab, you can use the **Left** and **Right** keyboard arrows to navigate from one Tab element to the next Tab element in the Form having the Tabs control.

Tab

Use **Tab** control to group controls placed on the control (child controls) in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
Description	Description for the control.

Appearance

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**. **Note:** If the value is **No**, then the child controls will be hidden even when the **Visible** property for the child controls is set to **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.

Advanced

Property	Description
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.

Scripts

For information about using scripts, see scripting guidelines for Tab control.

Controls

This section explains the properties of different form controls available in AVEVA Work Tasks.

Arc

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	(Required) XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	example, `@FirstName`. • Use forward slash (**/**) for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent, which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it may affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Enable Column Bindings	Enable this option to set up the lookup source. • Select **Yes** to enable use of the lookup source. • Select **No** if column binding is not necessary. If enabled, specify the column names based on the lookup source definition for **Unit of Measurement**, **Target Value**, and **Default Value** properties**.** The default value is **No**.
Lookup Source	(Required) Name of the item from the lookup list. **Note**: This is only visible when the **Enable Column Bindings** property is set to **Yes**. To add a name of the item for the control from the lookup list: 1. Click the **Lookup Source** button. The Lookup Source Configuration page appears. 2. In the **Data Source Type** list, select a type of data source. **Note**: Supported datasource types are database and Web API lookup source. 3. In the **Data Source** box, select a data source. 4. In the **Parameters** section, select the parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	to include, if applicable. 5. Click **Save & Continue**. The selected name is set for the control and the Lookup Source Configuration page is closed.
Unit of Measurement	Unit of measurement for the value of the control. This supports two lines with 12 characters for each line and a tooltip is shown when you hover over this property. An ellipsis is added if it exceeds the 12-character limit.
Target Value	Target value of the control. This is the value that the target indicator points at. If this property is left blank, the target indicator is not shown at preview or runtime.
Default Value	This is the value of the control. This supports 10 characters and a tooltip is shown when you hover over this property. An ellipsis is added if the value exceeds the 10-character limit.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note**: If the value is 0, then the width of the control is set as per the default value of 360.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Title	Name or information text for the control. It is shown

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	directly above the top-left corner of the control. This supports two lines with 20 characters for each line and a tooltip is shown when you hover over this property. An ellipsis is added if it exceeds the 20-character limit. If you want to add a line break, use the html tag ` `. **Note:** If this property is not configured or blank, nothing is shown, the other control contents use its space, and the control size is adjusted based on the available content within the control. The described behavior also applies to the **Unit of Measurement** property.
Decimal Places	Decimal places for the default value of the control. It also applies to the values in the **Range** property. This displays a maximum number of 3 decimal places. The default value is **0**. **Note**: At runtime, if this property is empty or has a non-numeric value, it resets back to 0. If its value is greater than 3, it stays at its maximum value of 3.
Scale Minimum Value	(Required) Minimum value displayed on the scale. The default value is **0**. **Note**: At runtime, if this property is empty or has a non-numeric value, its value is set to 0.
Scale Maximum Value	(Required) Maximum value displayed on the scale. The default value is **100**. **Note**: At runtime, if this property is empty or has a non-numeric value, its value is set to 100.
Scale Start Angle	Angle at which the scale starts. The default value is **0**, where the angle starts at 9 o 'clock. This accepts a minimum value of 0 and maximum of 360. **Note**: If this property is left blank, it resets back to its default value.
Scale Span Angle	The length at which the angle spans based on the **Scale Start Angle** property. The default value is **180** degrees. This accepts a minimum value of 0 and maximum of

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	360. **Note**: If this property is left blank, it resets back to its default value.
Show Scale	Show or hide the scale in the control. • Select **Yes** to show the scale in the control. • Select **No** to hide scale in the control. If this is set to **Yes**, the **Number of Division** property is visible. The default value is **No**.
Number of Division	Number of division within the scale, from the scale minimum value to the maximum value. This number divides the difference between the **Scale Maximum Value** and **Scale Minimum Value**. For example: • **Scale Minimum Value** = 0 • **Scale Maximum Value** = 200 • **Number of Division** = 5 So the difference between the scale max and min values is 200. The difference divided by the number of division is 40. The scale tick marks are then shown as 0, 40, 80, 120, 160, and 200. **Note:** This is only visible when the **Show Scale** property is set to **Yes**. The default value is **10**.
Control Background Color	The color to represent the background of the control. The default is white.
Background Color	The color to represent the background of the bar in the control. The default is gray.
Bar Color	The color to represent the bar in the control. The default is green.
Range	The breakpoint to apply in the control range bar. Each added row corresponds to a breakpoint and for each breakpoint value, you can assign a color. You can

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	have a maximum of 10 breakpoints. **Note**: If the configured range does not cover the full scale, the undefined range takes the color that is set for the **Bar Color** property. The value for the breakpoint must be between the specified value for the **Scale Minimum Value** and **Scale Maximum Value** properties.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
On Initialize	Run a script after the control has been initialized, but before it is rendered to the UI. For more information, see *Scripting Guidelines*.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Enable Refresh	Enable or disable control refresh. • Select **Yes** to enable control refresh and configure the **Refresh Rate** property. • Select **No** to disable control refresh. The default value is **No**.
Refresh Rate	This is how often the control display is refreshed, in seconds, to fetch value from the data source. **Note**: This is only visible if the **Enable Refresh** property is set to **Yes**. The default value is 30 seconds and the valid refresh rate is from 5 to 600 seconds.

Scripts

For information about using scripts, see scripting guidelines for Arc control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example Arc Rendering

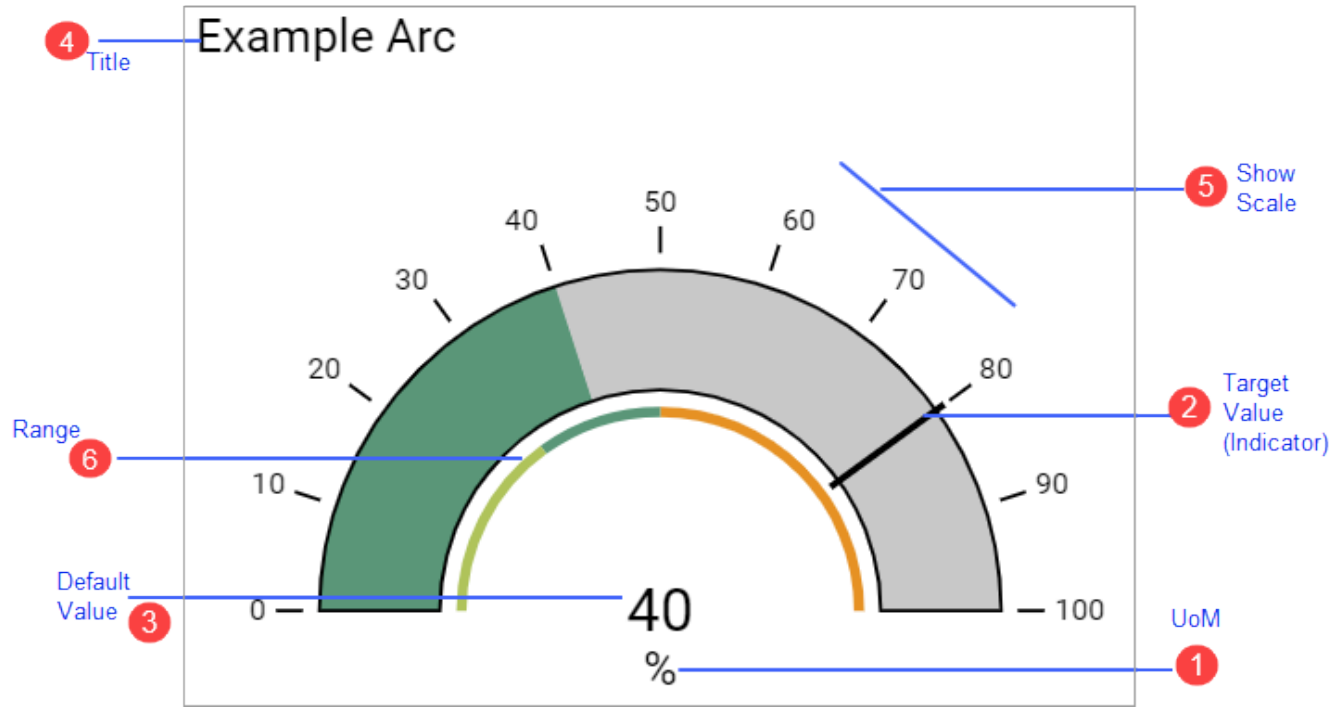
The following rendering examples for Arc control are based on the configuration in the control's properties page. The configured properties are listed in the table and most of the properties that are kept to their default values are no longer mentioned.

Example configuration and rendering

Basic	Appearance
1. **Unit of Measurement**: `%` 2. **Target Value**: `80` 3. **Default Value**: `40`	4. **Title**: `Example Arc` 5. **Show Scale**: `Yes` (selected) • **Number of Division**: `10` (default) 6. **Range** (*with three breakpoints*) • Value 1 - 30 (lime) • Value 2 - 50 (green) • Value 3 - 100 (orange)

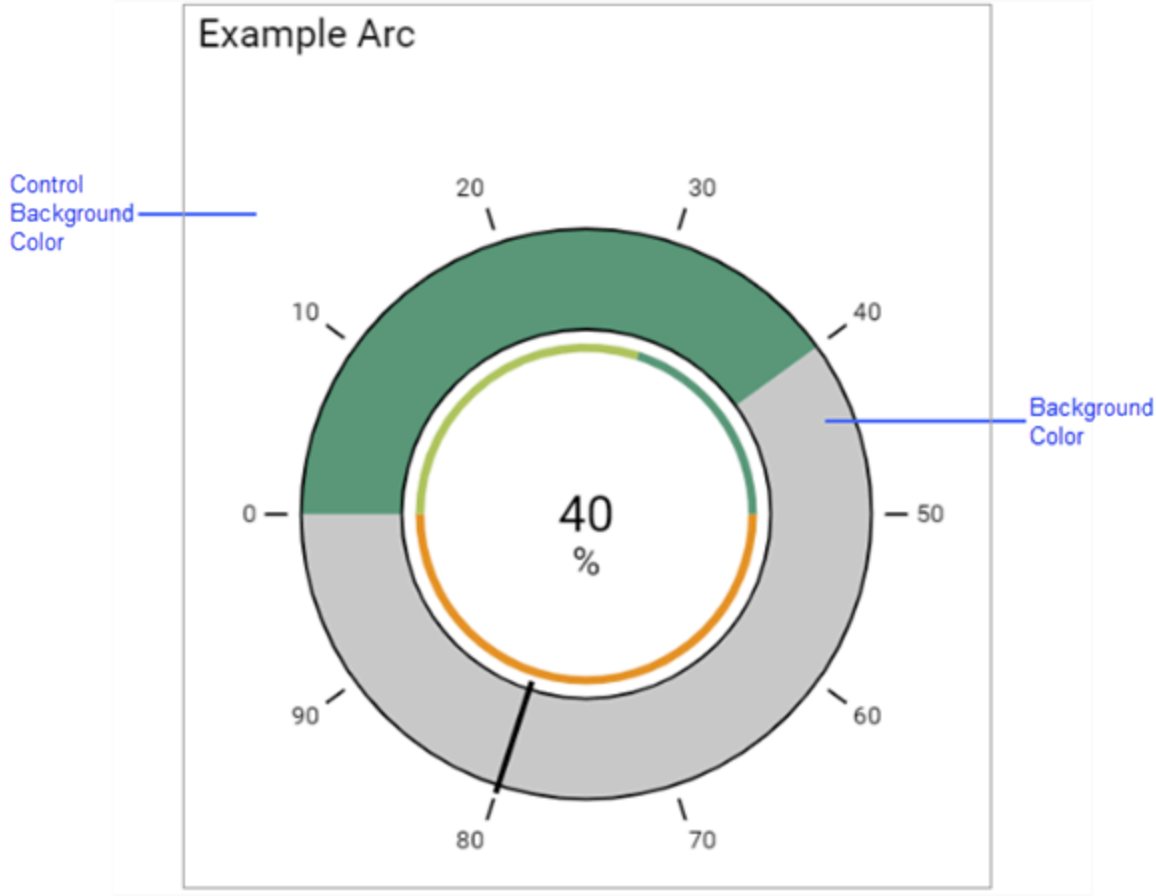
The range bar in the following example shows the three breakpoints as specified in the configuration. The target indicator is pointing at the specified **Target Value** of 80.

For the rendered bar color to show the **Default Value**, notice that it takes the assigned color of the second breakpoint (50, green) because the **Default Value** falls on that breakpoint range.



The following example rendering has the same configuration, but the **Scale Span Angle** property is set to 360.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Attachment

Use **Attachment** control to attach a file in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows. -Prefixing with **@** is not supported when the **Storage Type** property is set to **File System**.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
File Name Length	Length of characters for the file name.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	The default value is 0. For example, if the value is set to 8 and the file name of the attachment has 12 characters, then only the first 8 characters of the file name are retained. If the file name of the attachment has 6 characters, then the file name is retained as is. The abbreviated file name is displayed in the form. However, when you download the file, the name of the file is retained as is. **Note:** If the value is 0, then the file name is retained as is.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Attachment Type	File type of attachment for the control. • Select **All** to attach any file types of attachment. • Select **Images** to restrict the file types of attachment to images (BMP, DIB, GIF, JFIF, JPE, JPEG, JPG, and PNG). • Select **Tagged Images (TIFF)** to restrict the file types of attachment to tagged images (TIF and TIFF). • Select **Portable Documents (PDF)** to restrict the file types of attachment to portable documents. • Select **Office Documents** to restrict the file types of attachment to office documents (DOC, DOCM, DOCX, DOTM, DOTX, MPP, MPT, POTM, POTX, PPAM, PPSM, PPSX, PPT, PPTM, PPTX, SLDM, SLDX, THMX, XLAM, XLS, XLSB, XLSM, XLSX, XLTM, XLTX, and XSN). The default value is **All**. **Note:** - If you select **All** along with other options, then the other options are ignored and you can attach any file types of attachment.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	- If no option is selected, then you can attach any file types of attachment. - You cannot attach the files having .exe file extensions as they are not supported.
Custom Attachment Type	Custom file type of attachment for the control. For example, `.txt;.rtf` to restrict the file types of attachment to text and rich text formats. **Note:** This value disables the **Attachment Type** property and restricts the allowed file type to the value specified.
Show Delete Confirmation Message	Show or hide the custom delete confirmation message for the control. • Select **Yes** to show the custom delete confirmation message for the control. • Select **No** to hide the custom delete confirmation message for the control. The default value is **No**.
Delete Confirmation Message	Delete confirmation message for the control. **Note:** This property is applicable only when the **Show Delete ConfirmationMessage** property is set to **Yes**.

Advanced

Property	**Description**
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Storage Type	Storage type of attachment for the control. • Select **Database** to store attachment in the database. • Select **File System** to store attachment in the file system. The default value is **Database**. **Note:** - **File System** as storage type does not support:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	-- attachments that are part of lists. -- archiving and purging.
Security Mode	Set the security mode to None or Encryption to secure the attachment. • Select **None** to have the value for the attachment in database as Base64 encoded. The value in Form Instance XML will not be encrypted. • Select **Encryption** to have the encrypted value for the attachment in the database as BLOB. The value in Form Instance XML will be encrypted. The default value is **Encryption**. **Note:** This property is applicable only when the **Storage Type** property is set to **Database**.
Persist File Name	Keep the file name of the attachment as the header. • Select **Yes** to keep the file name. • Select **No** to rename the file. The default is **Yes**. **Note:** This property is applicable only when the **Storage Type** property is set to **Database**.
Persist Custom File Name	ID of the Text control whose value is used as the header. If you do not want to persist the file name, then you can persist a custom file name in another control. The **Persist Custom File Name** property takes the ID of a Text control where it can store the attached file name information. You can then get the file name from this control through scripts. **To select the Text control** 1. Click the **Persist Custom File Name** button. The **Text Control Configuration** page appears. 2. In the **Text Control** list, select a Text Control. 3. Click **Save & Continue**. The ID of the selected Text control is set for the **Persist Custom File Name** property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: - This property is applicable only when the **Storage Type** property is set to **Database**. - We recommend to have the **Text** control in the same container where the **Attachment** control is present.
File Path Template	File path template for storing attachment in the file system for the control. For information about using file path template tags, see *File Path Template Tags*. The default value is **<StoragePath>\<Year>\<Month>\<Date>\<GUID>_<FileName>.<FileExtension>**. **Note:** - This property is applicable only when the **Storage Type** property is set to **File System**. - The **<StoragePath>** tag should be the first tag in the file path template, otherwise the file path template will be invalid. - The application pool identity of the web application should have the appropriate rights to the folder where the attachments are stored. - Date and time values resolved in file path template tags are in Universal Coordinated Time (UTC) format, and the time value is in 24-hour time format.
Allow Multiple Attachments	Allow or deny a user to attach multiple files for the control. • Select **Yes** to allow a user to attach multiple files for the control. • Select **No** to deny a user to attach multiple files for the control. The default value is **No**. **Note:** - This property is applicable only when the **Storage Type** property is set to **File System**. - In AVEVA Work Tasks Pro for Android: -- Multiple files can be attached only from the gallery, when the value of the property is set to **Yes**. -- Only the first attachment gets attached when the value of the property is set to **No**.
Maximum Attachments	Maximum number for attachments for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is **5**. **Note:**- This property is applicable only when the **Allow Multiple Attachments** property is set to **Yes**. - By default, the maximum value allowed is 10. If you want to change this, then override the **AttachmentMaximumLimit** entry in the **Properties** column in the **SKRepository** table of the Farm database. -- For example, to override the maximum value allowed to 20, change **[CDATA[<<Maximum Number of Attachments>>]]** as follows: `<property key="AttachmentMaximumLimit"><![CDATA[20]]></property>` -- If the value specified is less than **2**, then the value is taken as **2**. -- If the value specified is greater than **99**, then the value is taken as **99**.
Source Type	Source type of attachment for the control. • Select **Camera** to capture the image and store it in the selected storage type. • Select **Gallery** to select the existing file and store it in the selected storage type. • Select **Camera and Gallery** to either capture the image or select the existing file. The default value is **Camera and Gallery**. **Note:** - This property is available only for **AVEVA Work Tasks Pro** app. - For UWP, this property is not applicable. - For iOS, the Gallery option opens both the camera and gallery file picker options.

Scripts

For information about using scripts, see scripting guidelines for Attachment control.

Note:
- We recommend to keep the file size lesser than 30 MB to avoid errors during the upload. Also, you can download a file only if the size is 30 MB or lesser.
- The request timeout controls the time taken for a request to process, which also controls uploading/ downloading the file. So, it is recommended to set the Request timeout to an appropriate value as per the upload limit set in the environment. The default value for the request timeout is set to 900 secs.
- The upload limit is based on the settings for the `maxAllowedContentLength` setting in the web.config of the Enterprise Console Web Application. The default limit is 4 MB. Refer the Microsoft documentation for any

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

changes to be made for increasing the upload limit.
- The attachment might take some time to upload when the system optimizes the database for storage of attachments.
- When the **Storage Type** is set to **Database**, the attachment is uploaded to the server immediately after clicking **Upload**.
-- The uploaded attachments are stored in the **Data** column of the **SKEListItemDataTemp** table.
-- The Attachment control value securely holds information to the location where the attachment is stored.
-- The uploaded attachment cannot be retrieved or processed through workflows and so on.
- When the **Storage Type** is set to **File System**, the attachment is not uploaded to the server immediately after clicking **Upload**. The attachment is uploaded when you:
-- submit the form (even in preview mode)
-- click **Invoke Workflow** (even in preview mode)
-- submit **Pop-Up** form (even in preview mode)
-- save as drafts
- Special characters " ", ";", and "," in the file name of the attachment will be replaced with "_" in the downloaded attachment.

File Path Template Tags

Tag	Description
<StoragePath>	Resolves to the **Attachment Storage Path** specified while creating the repository. For example, If the specified **Attachment Storage Path** is **C:\ProgramData\Workflow Management\ Attachments**, it is resolved to **C:\ProgramData\Workflow Management\ Attachments** **Note:** The **<StoragePath>** tag should be the first tag in the file path template, otherwise the file path template will be invalid.
<Year>	Resolves to the current year in **yyyy** format. For example, If the current date is **June 7, 2018** it is resolved to **2018**
<Month>	Resolves to the current month in **MM** format. For example, If the current date is **June 7, 2018** it is resolved to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Tag	Description
	06
<Date>	Resolves to the current date in **dd** format. For example, If the current date is **June 7, 2018** it is resolved to **07**
<Hours>	Resolves to the current hour in **HH** format. For example, If the current time is **10:02:03.004 PM**, it is resolved to **22**
<Minutes>	Resolves to the current minute in **mm** format. For example, If the current time is **10:02:03.004 PM**, it is resolved to **02**
<Seconds>	Resolves to the current second in **ss** format. For example, If the current time is **10:02:03.004 PM**, it is resolved to **03**
<Milliseconds>	Resolves to the current millisecond in **mmm** format. For example, If the current time is **10:02:03.004 PM**, it is resolved to **004**
<GUID>	Resolves to a UUID in **RFC4122 version 4 (random)** format (**xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx**) For example, **527ee1d8-2c01-4d0a-b3c1-a8b121ca020b**
<FileName>	Resolves to the original file name of the attachment. For example,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Tag	Description
	If the attached file is **Keywords.doc**, it is resolved to **Keywords**
<FileExtension>	Resolves to the original file extension of the attachment. For example, If the attached file is **Keywords.doc**, it is resolved to **doc**

Boolean

Use **Boolean** control to select either Yes or No in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Default Value	The default value for the control.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the **control**. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	Position property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Options Direction	Direction of the options for the control. • Select **Horizontal** to align the options next to each other for the control. • Select **Vertical** to align the options one below the other for the control. The default value is **Horizontal**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	contain only alphanumeric characters.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Include in Summary	Allow or deny a user to view the value in the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the Summary of a Base Form. • Select **No** to deny a user to view the value in the Summary of a Base Form. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Boolean control.

Button

Use **Button** control to perform an action in your form. You can display both text and image on the control.

Properties

The properties for the control are grouped under the following tabs:

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Button Text	Text for the control. **Note:** - This value is required when the control is used with the **Possible Distinct Outputs** property for **Invoke Form** activity of the workflow. - For information about Invoke Form activity and Possible Distinct Outputs, see *Invoke Form Activity* in *Developer Guide*.
Button Type	Action for the control. • Select **Submit** to allow the control to submit the form. • Select **Validate** to allow the control to validate controls on the form. • Select **Cancel** to allow the control to close or reload the form. • Select **Custom** to allow the control to perform custom actions on the form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Print** to allow the control to print the form. This feature is not available for mobile devices. • Select the **Navigate** option to navigate to another form. The default value is **Custom**. When you select **Print**, the **Print Form** option appears. You can link a form that you want to print. **To link a form that you want to print** 1. Click the **Print Form** button. The **Form Details Configuration** page appears. 2. In the **Form Name** box, click a form name. 3. In the **Form Version** box, click a form version. 4. Click **Save & Continue**. The selected form is linked to the control, and the **Form Details Configuration** page is closed. When you select **Navigate**, the **Navigate To Form** option appears. You can navigate to another form, from the current form. **To navigate to a form** 1. Click the **Navigate to Form** button. The **Form Details Configuration** page appears. 2. In the **Form Name** box, click a form name. 3. In the **Form Version** box, click a form version. 4. Click **Save & Continue**. The selected form is embedded in the current form, and the **Form Details Configuration** page is closed. **Note:** - If the value is **Validate** and the **Form Button** property is set to **Yes**, then the control will validate all the controls on the form. - If the value is **Validate**, the **Form Button** property is set to **Yes**, and the control is placed on **Tab**, then the control will validate all the controls on the **Tab**. -If the value is **Cancel** and the form is loaded in a separate window, then the control closes the form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	- If the value is **Cancel** and the form is loaded within an existing window (inline), then the control reloads the form. - If the value is **Custom**, then the control will perform custom actions on the form as per the script set for the **On Click** property of the control. - The **Print** feature is not supported in mobile devices. - The **Navigate** feature is applicable for Form Links. - Consider a scenario where a **Button** control of Button Type **Navigate** is consumed in a pop-up form. To close the pop-up form, use a custom button and use the SFU.closeForm API in the form to be navigated to. If you are using Invoke Workflow to invoke a workflow, call the SFU.closeForm API in the post-execution script of the Invoke Workflow. You can close the pop-form using this API only if the **Open New Window** property of the **Pop-Up** control is set to **YES**. However, if the value of this property is set to **NO**, then the pop-up form does not get closed. - Ensure that the form you want to navigate to is published, before it is consumed during runtime. - Forms parameters are not supported for the **Navigate** option. - Consider a scenario where a script is written for the **On Click** event with the **Button Type** property set to **Navigate** and the **On Click** script makes an AJAX call to a web API. If the AJAX call is *Synchronous*, the navigation to another form will happen only after the response from the Web API is received. However, if the AJAX call is *Asynchronous*, the navigation might happen before the response from the Web API is received. - Consider a scenario where 'Form1' has been configured to navigate to 'Form2'. If you preview the 'Form1' and click the **Navigate** button, the 'Form2' will be displayed. However, if you try to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	submit the 'Form2', the **instance xml** will not be displayed. - The form configured as **Navigate To** option is a Fill form type. If there is **Submit** button for that form, then associate a workflow to the form.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Alignment	Alignment of the control with respect to the form. • Select **Default** to allow the system to align the control. The system aligns the control to the left of the form or as specified by the control. • Select **Left** to align the control to the left of the form. • Select **Right** to align the control to the right of the form. • Select **Center** to align the control to the center of the form.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Display Mode	Show text and/or image for the control. • Select **Text** Only to show only text for the control. • Select **Image** Only to show only image for the control. • Select **Text and Image** to show both text and image for the control. The default value is **Text Only**.
Button Size	Size of the button for the control. • Select **Standard** to set the size of the control to 120 x 32 pixels. • Select **Small** to set the size of the control to 30 x 30 pixels. • Select **Medium** to set the size of the control to 180 x 86 pixels. • Select **Large** to set the size of the control to 280 x 130 pixels. The default value is **Standard**.
Image Path	Path to the image for the control. The image path is specified as: • A URL starting with **http://** or **https://**. • An image path tag. For information about using image path tags, see *Image Path Tags* in Image. • A path to the **BPMUITemplates** folder as **BPMUITemplates\Default\Images**. **Note:** You can use relative paths as follows: - Use **.** for the current folder where the html page resides. - Use **..** for the folder above the current folder where the html page resides. We recommend the following image sizes when the **Display Mode** property is set to Image. • **Small** – 30 x 30 pixels. • **Medium** – 180 x 86 pixels. • **Large** – 280 x 130 pixels.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	We recommend the following image sizes when the **Display Mode** property is set to Text and Image. • **Small** – 30 x 30 pixels. • **Medium** – 40 x 40 pixels. • **Large** – 50 x 50 pixels. **Note:** - This property is applicable only when the **Display Mode** property is set to **Image** or **Text and Image**, and the **Button Size** property is set to **Small**, **Medium** or **Large**. - If the **Display Mode** property is set to **Text and Image** and the **Button Size** property is set to **Small**, then only the image is displayed.
Image Position	Position of the image with respect to the text for the control. • Select **Left** to align the image to the left of the text for the control. • Select **Right** to align the image to the right of the text for the control. • Select **Top** to align the image to the top of the text for the control. • Select **Bottom** to align the image to the bottom of the text for the control. The default value is **Left**. **Note:** This property is applicable only when the **Display Mode** property is set to **Text and Image**.
Show as Link	Enable or disable the control to show as link. • Select **Yes** to enable the control to show as link. • Select **No** to disable the control to show as link. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	contain only alphanumeric characters.

Validation

Property	Description
Show Confirmation Message	Show or hide the confirmation message for the control. • Select **Yes** to show the confirmation message for the control. • Select **No** to hide the confirmation message for the control. The default value is **No**.
Confirmation Message Text	Confirmation message for the control. **Note:** This property is applicable only when the **Show Confirmation Message** property is set to **Yes**.

Advanced

Property	Description
On Click	Run a script when a button is clicked. For more information, see scripting guidelines for On Click property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Form Button	Enable or disable the control to act as form button. When this property is enabled, the control is displayed at the bottom-right corner of the form. • Select **Yes** to enable the control to act as form button. • Select **No** to disable the control to act as form button. The default value is **No**. **Note:** - If the value is **Yes**, and the form is viewed on a mobile device, then the control appears in the **Action Bar** of the mobile device. -If the Form Button control is placed in the **Container**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	and the value is set to **Yes**, the control is not displayed at the bottom-right corner of the form.

Scripts

For information about using scripts, see scripting guidelines for Button property.

Card

Use **Card** control to display data in a card from a Lookup data source in your form.

Properties

Control properties are grouped into three tabs: **Basic**, **Appearance**, and **Advanced**. The **Scripts** tab is used for properties configuration through scripting.

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	(Required) XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use forward slash (**/**) for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent, which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it may affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Lookup Source	Name of the item from the lookup list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	To add a name of the item for the control from the lookup list: 1. Click the **Lookup Source** button. The Lookup Source Configuration page appears. 2. In the **Data Source Type** list, select a data source type. 3. In the **Data Source** box, select a data source. 4. In the **Parameters** section, select the control properties from the list, which you want to map to each of the given parameters, if applicable. 5. Click **Save & Continue**. The selected name is set for the control and the Lookup Source Configuration page is closed.
Enable Dynamic Label	Enable use of the value (2-5) label properties for data mapping based on the selected data source lookup. • Select **Yes** to enable mapping to a column name in the data source using the value label properties. • Select **No** to disable column name mapping and use the value label properties for static button name or text for localization. The default value is **No**.
HeaderValue Left	A text field associated to a column name in the selected data source lookup. You need to check for the column name in the data source and manually enter it in the control property page to assign it to this property. Example column name: • `entity_name` • `wo_id` The corresponding value for the assigned column name appears on the top left of the control's header bar. **Note**: If this property is blank, nothing is shown, but its space is retained in the control. You have the option to configure the number of displayed characters through **HeaderValue Left Display Limit** property. The default column value

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	display limit is 12. **Note**: If the column value is more than 12 characters, you can adjust the font size to fit in the control through CSS.
HeaderValue Right	A text field associated to a column name in the selected data source lookup. You need to check for the column name in the data source and manually enter it in the control property page to assign it to this property. Example column name: • `state_desc_h` *(Job state)* • `wo_status_desc` *(WO status)* The corresponding value for the assigned column name appears on the top right of the control's header bar. **Note**: If this property is blank, nothing is shown, but its space is retained in the control. The maximum number of displayed characters for the column value is 10. If the column value has more than 10 characters to display, an ellipsis is added to indicate that, and the full column value is shown in a tooltip on mouse hover.
Value1	A numeric or text field associated to a column name in the selected data source lookup. You need to check for the column name in the data source and manually enter it in the control property page to assign to this property. Example column name: • `item_id` • `wo_id` The corresponding value for the assigned column name appears in the middle of the control just below the header bar. If this property is blank, nothing is shown, but its space is retained in the control. The maximum number of displayed characters for the column value is 26. If the column value has more than 26 characters to display, an ellipsis is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	added to indicate that, and the full column value is shown in a tooltip on mouse hover.
Value (2-5)	A text field associated to a column name in the selected data source lookup. You need to check for the column name in the data source and manually enter it in the control property page to assign to this property. Example column name: • `entity_name` • `wo_id` The assigned column name to the property and its corresponding value appear in the control after the **Value1** property. The maximum number of displayed characters for the column value in **Value 2** is 27 and in **Value (3-5)** is 30. If the column value has more characters to display than the maximum number, an ellipsis is added to indicate that, and the full column value is shown in a tooltip on mouse hover.
Value (2-5) Label	Description for the **Value** property, text to support localization, or a value of a lookup column name. For example: • `Entity Name` • `@@entity_name@@` *(for localization)* If the **Enable Dynamic Label** property is set to **No**, you can use each **Label** property to specify a description for a particular **Value** property or use each of them for localization text. For localization, ensure that you add the value you specified in the **Label** property in its corresponding resource file in the **Locales** folder under the product installation path. If the **Enable Dynamic Label** property is set to **Yes**, you can assign a column name to each **Label** property from the selected data source lookup, and the value of that column name is shown in the control as the label for a particular **Value** property**.** If the value of the column name is empty, nothing is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	shown, but its space is retained. **Note**: If a particular **Label** property is not configured or blank, that field is not shown. The other values use its space, and the size of the control is adjusted based on the available data (values) within the control. The maximum number of displayed characters for the column value in **Value2 Label** is 34 and in **Value (3-5) Label** is 36. If the column value has more characters to display than the maximum number of characters, an ellipsis is added to indicate that, and the full column value is shown in a tooltip on mouse hover.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Width	Width in pixels for the control. The default value is **0**. **Note**: If the value is 0, then the width of the control is set as per the default value.
Height	Height in pixels for the control. The default value is **0**. **Note**: If the value is 0, then the height of the control is set as per the default value and is automatically adjusted based on the content within the control.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. This is applicable only when the **Label Position** is set to **Left** or **Right**.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Image Path	The location of the image that you want to use in the control. This can be a URL that starts with **http** or **https**, or path with **<BPMJITemplates>**, **<ThemePath>**, or **<ThemeName>** tags.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	Note: The supported icon size is 45 pixels.
HeaderValue Left Display Limit	Number of characters to show for the **HeaderValue Left** property before the ellipsis. This accepts a maximum display limit of 25. If the column value has more characters to display than the specified limit, an ellipsis is added to indicate that, and the full column value is shown in a tooltip on mouse hover. The default value is **12**. **Note**: If the column value is more than 12 characters, you can adjust the font size to fit in the control through CSS.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
On Click	Run a script after the control is clicked.
On Initialize	Run a script after the control has been initialized, but before it is rendered to the UI.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Enable Lookup Color	Enable use of color defined in the selected data source lookup for the control's header bar. • Select **Yes** to enable use of the color from the data source lookup. • Select **No** to enable the color picker in the **Color** property and select the color to use. The default value is **Yes**.
Color	Color for the control's header bar. If the **Enable Lookup Color** property is set to **Yes**, specify the column name from the lookup that contains the color in this property, otherwise use the color picker to assign the color for the control's header bar.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Auto-load Card	Allow or deny the system to automatically load the control. • Select **On Form Load and Parameter Change** to allow the system to automatically load the control when the form is loaded and when the parameter is changed. • Select **Never** to deny the system to automatically load the control when the form is loaded and when the parameter is changed. The default value is **On Form Load and Parameter Change**. **Note**: For parameterized lookup, when setting the value to **Never**, use the **control.refreshCard()** method to reload records on demand in the Card control.
Enable Refresh	Enable or disable control display refresh. The default value is **No**.
Refresh Rate	The number of times the control display is refreshed in seconds. The default value is **30** seconds.
Enable Dynamic Action Buttons Label	Enable use of the action button (1-3) label properties for data mapping based on the selected data source lookup. • Select **Yes** to enable mapping to a column name in the data source using the action button properties. • Select **No** to disable column name mapping and use the action button properties for static button name or text for localization. The default value is **No**.
Enable Action (1-3)	Call the script defined in a particular **Action Script** property in the **Scripts** tab. If you select **No**, the action button is not visible in the control. The default value is **No**.
ActionButton (1-3) Label	Name for **ActionButton (1-3)** or text to support localization. If the **Enable Dynamic Action Buttons Label** property is set to **No**, you can use each **Action Button Label**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	property to specify a name for a particular **ActionButton** or use it for localization text. For localization, ensure that you add the value you specified here in its corresponding resource file in the **Locales** folder under the product installation path. If the **Enable Dynamic Action Buttons Label** property is set to **Yes**, you can assign a column name to each **Action Button Label** property from the selected data source lookup, and the value of that column name is shown in the control as the label for the action button. **Note**: If a particular **Action Button Label** property is not configured or blank, nothing is shown, and the way the other action buttons are shown is adjusted accordingly. The maximum number of displayed characters for the column value is 10. If the column value has more than 10 characters to display, an ellipsis is added to indicate that, and the full column value is shown in a tooltip on mouse hover.

Scripts

For information about using scripts, see scripting guidelines for Card control.

Chart

Use **Chart** control to view a chart in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
Description	Description for the control.
Lookup Source	Name of the item in the **Lookup List**. **To add a name of the item in the Lookup List for the control** 1. Click the **Lookup Source** button. The **Lookup**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	Source Configuration page appears. 2. In the **Data Source Type** list, click a data source type. 3. In the **Data Source box**, click a data source. 4. In the **Parameters** list, click and select a control on the form, if applicable. 5. In the **X-Axis** list, click a data series with unique values for the category (horizontal) axis. 6. In the **Y-Axis** list, click one or more integer or numeric data series for the value (vertical) axis. 7. Click **Save & Continue**. The selected name is set for the control and the **Lookup Source Configuration** page is closed. To configure the Lookup Source for Sankey chart, see Lookup Source Configuration for Sankey Chart. **Note:** - Parameterized and non-parameterized lookups are supported for Database. Only non-parameterized lookups are supported for Web API. - To render donut or pie chart, only one integer or numeric data series should be selected for the y-axis of the chart. - For information about Lookup List, see List Lookups.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	• Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **450**. **Note:** If the value is 0, then the height of the control is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Chart Type	Type of chart for the control. • Select **Area Chart** to show area chart for the control. • Select **Bar Chart** to show bar chart for the control. • Select **Donut Chart** to show donut chart for the control. • Select **Line Chart** to show line chart for the control. • Select **Pie Chart** to show pie chart for the control. • Select **Sankey Chart** to show a flow from one set of values to another. • Select **Scatter Chart** to show scatter chart for the control. The default value is **Area Chart**. **Note:** - To render donut or pie chart, only one integer or numeric data series should be selected for the y-axis of the chart. - For Sankey chart, the chart type has to be selected before selecting the Datasource. - In **AVEVA Work Tasks Pro** app, the Sankey chart width does not get adjusted in the container when the orientation is changed.
Show Axis Titles	Show or hide the axis titles for the control. • Select **Yes** to show the axis titles for the control. • Select **No** to hide the axis titles for the control. The default value is **Yes**.
Show Legend	Show or hide the legend for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	• Select **Yes** to show the legend for the control. • Select **No** to hide the legend for the control. The default value is **Yes**.
Sankey Chart Node Unit	Displays the unit used for Sankey chart nodes.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
On Initialize	Run a script after the control has been initialized, but before it is rendered to the UI.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.

Scripts

For information about using scripts, see scripting guidelines for Chart control.

Note: In **AVEVA Work Tasks Pro** app, the chart is rendered after sampling down the data to 1000 points, if the data exceeds 1000 points.

Lookup Source Configuration for Sankey Chart

Sankey chart is a visualization used to depict a flow from one set of values to another. The things being connected are called nodes and the connections are called links. Sankey chart is used when you want to show a many-to-many mapping between two categorical dimensions.

To configure Lookup Source for Sankey chart

1. Click the **Lookup Source** icon. The **Lookup Source Configuration** page appears.
2. In the **Data Source Type** list, select a data source type.
3. In the **Data Source for Sankey Nodes**, select a data source for the Sankey nodes.
4. In the **Sankey Node Id**, select a node Id that will be displayed in the Sankey chart. These Ids can be used to create links between source and target nodes. The node Id should have unique value.
5. In the **Sankey Node Names**, select a node name that will be displayed in the Sankey chart. The Sankey Node name should be a string.
6. In the **Data Source for Sankey Links**, select a data source for the Sankey links.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. In the **Sankey Source Nodes**, select the source node for the Sankey chart. The source node should have the value specified in the Sankey Node Id field.
8. In the **Sankey Target Nodes**, select the target node for the Sankey chart. The Target node should have the value specified in the Sankey Node Id field.
9. In the **Sankey Link Value**, select a link value between the source and the target nodes. The link value should be an integer.
10. Click **Save & Continue**. The selected name and Lookup Source configuration is set for the control and the **Lookup Source Configuration** page is closed.

Note:
- Parameterized and non-parameterized lookups are supported for Database. Only non-parameterized lookups are supported for Web API.
- For information about Lookup List, see List Lookups.

Check Box

Use **Check Box** control to turn on or turn off an option by selecting or clearing the check box in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Options	Options for the control. **To add an option to the control** 1. Click **Add Record**. A new record appears. 2. In the **Text** box, enter the options text. 3. In the **Value** box, enter the options value. 4. In the **Fore Color** palette, click a color (or in the **Color Code** box, enter hexadecimal code of the color), and then click **Apply**. The selected color is applied to the foreground of the options text. 5. In the **Back Color** palette, click a color (or in the **Color Code** box, enter hexadecimal code of the color), and then click **Apply**. The selected color is applied to the background of the options text. **To clear fore color for an option of the control** • On the **Fore Color** palette, in the **Color Code** box, enter **#1b1b1b**, and then click **Apply**. The **Fore Color** is reset to the default color. **To clear back color for an option of the control** • On the **Back Color** palette, in the **Color Code** box, enter **#1b1b1b**, and then click **Apply**. The **Back Color** is reset to the default color. **To delete an option from the control** • Click the **Delete** button. The record is deleted. **Note:** The options **Text** and **Value** parameters are mandatory and must be unique across all options.
Default Value	The default value for the control. **Note:** - The value must be one of the values set for the **Value** parameter in the **Options** property. -Multiple values must be separated with **;#**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to Top, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Options Per Row	Number of options per row for the control. Enter a value from 0 to 10.
Options Direction	Direction of the options for the control. • Select **Horizontal** to align the options next to each other for the control. • Select **Vertical** to align the options one below the other for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is **Horizontal**.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**. **Note:** Validation message appears only on submission of the form.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. Important: The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Text Parameter	Name that identifies the options text to use in scripts for the control.
Value Parameter	Name that identifies the options value to use in scripts for the control.
Forms 2011 Compatible	Make the control compatible with Forms 2011, that is the forms used in BPM 2011. • Select **Yes** to allow the system to make the control

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	compatible with Forms 2011. • Select **No** to deny the system to make the control compatible with Forms 2011. The default value is **No**. **Note:** When a control on a form is made Forms 2011 compatible, the XML instance of the form with the control is changed to ensure compatibility with Forms 2011.
Include in Summary	Allow or deny a user to view the value in the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the Summary of a Base Form. • Select **No** to deny a user to view the value in the Summary of a Base Form. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Check Box control.

Data Grid

Use **Data Grid** control to display data in tabular form from a data source in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Configuration Name	Specifies grid view configuration. **To specify grid view configuration** 1. Click the **Configuration Name** button. The **Grid View Configuration** page appears. 2. In the **Grid Configuration** dropdown list, select the required grid configuration. The data grid preview with columns and their values appears. The **Parameters** list is also displayed if there are any parameters associated with the grid configuration. In the **Parameters** list, click and select a control on the form, if applicable. On the data grid column, you can perform the following actions: • Sorting single or multiple columns • Sort column values in Ascending order

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Sort column values in Descending order • Resizing columns 3. Click **Save & Continue**. The selected name is set for the control and the **Configuration Name** page is closed. **Note:** - For **Parameters**, the **Title** of the control on the form is displayed, if set for the control. Otherwise, the **XML Node** of the control on the form is displayed. - In forms designer, when you try to parameterize on a hierarchical grid, only the root level columns are listed. - In the **Grid View Configuration** page, resizing the last data grid column to a smaller size is not supported due to Third Party Control limitations. - Ensure that you map a **Number** control to an **Integer** or **Numeric** type of Data Grid parameter.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Read-Only	Allow or deny a user to enter a value for the control. • Select **Yes** to deny a user to enter a value for the control. • Select **No** to allow a user to enter a value for the control. The default value is **No**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Suppress**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to Top, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Consider Available Height	Resize the height of the control or container based on the available height. • Select **Yes** to resize the height based on available height. • Select **No** to retain the space. The default value is **No**. **Note:** - The Consider Available Height property is not applicable for the following: -Designer view -Print form -Panel form with horizontal tab order - If this property is set to **Yes** and a value is specified in the **Height** property, minimum height is considered. - This property is applicable only in web view. - When more than one control or container is set with **Consider Available Height** at the same level of container or form, then the available space is distributed evenly between those controls or containers. For example, consider a Form which contains two Panel forms with **Consider Available Height** set to **Yes**, where Panel form 1 has two TextInput controls and Panel form 2 has one TextInput control. The Panel form 1's height is more than Panel form 2's height as Panel form 1 contains two TextInput controls and the available empty space in the form is evenly distributed between the two Panel forms.
Rearrange Columns	Allow or deny a user to reorder the columns by dragging the header cells. • Select **Yes** to allow a user to reorder the columns. • Select **No** to deny a user to reorder the columns. The default value is **No**. **Note**: At the runtime, the reordering of the columns is not

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	persistent. In hierarchical grid, the column reordering of the child grid, at runtime, is not persistent during the following conditions: -Pagination -Grid Refresh -In mobile view
Resize Columns	Allow or deny a user to resize the columns by dragging the edges of the header cells. • Select **Yes** to allow a user to resize the columns.. • Select **No** to deny a user to resize the columns. The default value is **No**.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Multiple Selection	Allow or deny a user to select multiple records for the control. • Select **Yes** to allow a user to select multiple records for the control. • Select **No** to deny a user to select multiple records for the control. The default value is **Yes**. For information on behavior of the data grids on data change with the **Multiple Selection** property, see *Multiple and Parametrized Data Grids*.
Show Export Button	Show or hide the export button for the control. • Select **Yes** to show the export button for the control. • Select **No** to hide the export button for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is **No**. **Note:** - The export button is available at every level in a hierarchical grid. - You cannot export all levels at same time. - If you select **No** to hide the export button,then the footer with paging buttons is visible only if the data of the grid is greater than the page size. This is applicable at every level. - If you select **Yes** to show the export button, then the footer is visible even if the data of the grid is lesser than or equal to the page size. - Export is not supported in Work Tasks Pro UWP App.
Export Type	File type of exported data for the control. • Select **All** to show both Excel document and Portable document export buttons for the control to export data to an Excel document or a Portable document respectively. • Select **Excel** to show the Excel document export button for the control to export data to an Excel document. • Select **PDF** to show the Portable document export button for the control to export data to a Portable document. The default value is **All**. **Note:** - PDF saves only the screenshots for the current page of the control. - When you export a Data Grid with large data as a pdf, the browser crashes. To avoid this issue, in the **Scripts** tab of the Forms container, set the script for the **On Form Load** property as follows: `$.getScript("js/pako_deflate.min.js");` This enables compression in the pdf, and it is required if your dataset is very large.
Export All Pages	Allow or deny a user to export all pages for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Yes** to allow a user to export all pages for the control. • Select **No** to deny a user to export all pages for the control. However, the system allows the user to export only the current page for the control. The default value is **No**.
Adapt UI	Change the user interface of the control based on the display width. • Select **Yes** to allow the system to change the user interface of the control based on the display width. The **Web View** is rendered for the control when the screen width is greater than or equal to 800 pixels. Otherwise, the **Mobile View** is rendered for the control. • Select **No** to deny the system to change the user interface of the control based on the display width. In this case, the **Web View** is rendered for the control. The default value is **Yes**. **Note:** The responsiveness of the container depends on the value set for the **Desktop View Only** property at the Form level.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Advanced

Property	Description
On Data Bound	Run a script when data is bound to the control. For more information, see scripting guidelines for On Data Bound property.
On Data Change	Run a script when value of the control changes. **Note:** In a data grid, if the persist column is of bit type, then the **On Data Change** event does not trigger as expected. For more information, see scripting guidelines for On Data Change property.
On Selection Change	Run a script when selection of the record changes. For more information, see scripting guidelines for On Selection Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Behavior	Show or hide data for the control. • Select **Data Grid** to show data for the control. • Select **Data Source** to hide data for the control. The default value is **Data Grid**. For information on using **Data Grid** values in other controls, see Using Data Grid Values in Options script. For more information on using a using data from a data grid set as a data source, see Use Data from Data Grid Set As Data Source.
Execution Mode	Allow or deny a user to retrieve data for server calls immediately. • Select **Asynchronous** to deny a user to retrieve data for server calls immediately. • Select **Synchronous** to allow a user to retrieve data for server calls immediately. The default value is **Asynchronous**. **Note:** In a hierarchical grid,this property is applicable only at the root level. The child levels are always in the **Asynchronous** mode.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Dynamic Page Size	Display records per page based on the available height. • Select **Yes** to display records per page based on available height. • Select **No** to display all the records. The default value is **No**. **Note**: - This property is not applicable to Print -If **Dynamic Page Size** property is set to **Yes**, then the value set for the **Page Size** property is considered as the minimum value. - The row height is fixed with **Dynamic Page Size** set to **Yes**. -The Data Grid with **Execution Mode** property set to **Synchronous** is not supported with **Dynamic Page Size** property set to **Yes**.
Page Size	Number of records added at run-time for the control. By default, the page size is set to greater than 1.
Extended Information	Allow or deny a user to access the extended information (that is columns other than the display and value columns of the result set) based on the selection for the control. • Select **Yes** to allow a user to access the extended information based on the selection for the control. • Select **No** to deny a user to access the extended information based on the selection for the control. The default value is **No**.
Select First Record	Allow or deny the system to automatically select the first record when the control is loaded. • Select **Yes** to allow the system to automatically select the first record when the control is loaded. • Select **No** to deny the system to automatically select the first record when the control is loaded. The default value is **No**. **Note:** - In the **Data Grid** control, the **Select First Record** property takes precedence over the script to select a

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	record in the **On Data Bound** event. - For example, in a **Data Grid** control when the **Select First Record** property is set to **Yes** and a script in the **On Data Bound** event is set to select the second record, the first record is selected - Therefore, control.selectRecord() method and the **Select First Record** property must not be used together. - In a hierarchical grid, if the **Allow Selection** property is applied at the child level, then the parent record expands automatically and the child record is selected.
Auto-Select Record	Allow or deny the system to automatically select the record when the check box is selected and vice versa. • Select **Yes** to allow the system to automatically select the record when the check box is selected and vice versa. • Select **No** to deny the system to automatically select the record when the check box is selected and vice versa. The default value is **No**. **Note:** - If the **Multiple Selection** property is set to **Yes** and the check box for the record is previously selected, then selecting the record does not clear the check box. - In such cases, select the check box for the record to clear the check box for the record.
Summary Columns	Columns to view in the summary of a **Base Form**. **Note:** - When the **Extended Information** property is set to **Yes**, any column can be specified. Otherwise, only the **Display** columns and the **Value** columns can be specified. Multiple values must be separated with a comma. - In a hierarchical grid, this property is applicable only for the root level columns.
Auto-load Grid	Allow or deny the system to automatically load the control. • Select **On Form Load and Parameter Change** to allow the system to automatically load the control

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	when the form is loaded and when the parameter is changed. • Select **On Form Load** to allow the system to automatically load the control only when the form is loaded. • Select **Never** to deny the system to automatically load the control when the form is loaded and when the parameter is changed. The default value is **On Form Load and Parameter Change**. **Note:** - For non-parameterized data grids, setting the value to **On Form Load and Parameter Change** is the same as setting the value to **On Form Load**. - Use the control.refreshGrid() method to reload records on demand in the data grid .
Include in Summary	Allow or deny a user to view the value in the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the Summary of a Base Form. • Select **No** to deny a user to view the value in the **Summary** of a Base Form. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.
Use Culture Specific Values	Enable values of the grid to be stored as culture specific values. • Select **Yes** if you want to store the values of the grid as culture specific values. • Select **No** if you do not want to store the values of the grid as culture specific values. The default value is **No**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: - Stored values of different data types such as **Number**, **String**, **Boolean**, **Date**, **Time**, and **DateTime** will not be persisted across different cultures. For example: If you submit a form from 'Fill Form' in 'en-US' culture and open the form from the 'Inbox' in 'pt-BR' culture, then the culture specific values are not persisted. - This property appears in **Data Grid** only for upgraded repository from versions prior to Wonderware Skelta BPM 2017. For a new repository, this property will not be displayed and the value will be **No** internally. - The **SFU.getGridValue()** API and the script having scriptable value as **selectedValue** and **extendedInformation** for data grid, now returns **DateTime** object. - The **DateTime** values are displayed as per the browser time zone.

Scripts

For information about using scripts, see scripting guidelines for Data Grid control.

For an example on accessing values of a column of a selected row in a Data Grid, see Accessing Values of a Selected Row Through Other Controls

Note:
- The format and culture for date and time value displayed in the control is as per the current server and not as stored in the database.
- The number values are displayed in the data grid as per the culture, considering only the decimal separator.
- When the data type is not a string but a number, date, time and so on, the data type is first converted to string for comparison and then filtered accordingly.
- If the persist column is of **DateTime** type, then in runtime the grid displays only the seconds. The milliseconds are neither displayed nor persisted. Filter options such as **equal to** will not work if milliseconds is persisted in the database for the **Date**, **Time**, and **DateTime** data types.
- If **Date** type is configured as Database column in SQL server, the **Date** type is displayed in the **DateTime** format in the UI.
- While filtering data, it is recommended to enter only valid inputs for **Date**, **Time**, and **DateTime** data types.
- If the Boolean type has Null data in the database and if the **Use Culture Specific Values** property for the **Data Grid** is set to **Yes**, then the Null data for the Boolean type is displayed as **False** in the UI.

Multiple and Parameterized Data Grids

You can use two data grids, and populate the destination data grid based on the record selected in the source data grid. The destination data grid is also called as the parameterized data grid as it is populated using the parameters from the source data grid.

The behavior of the data grids on data change with the **Multiple Selection** property is as follows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Value of the Multiple Selection Property set for the Source Data Grid	Value of the Multiple Selection Property set for the Destination Data Grid	Behavior of the data grids on data change
No	No	If you select a record in the source data grid, then the destination data grid is populated and you can select a record in the destination grid.
No	Yes	If you select a record in the source data grid, then the destination data grid is populated and you can select a record or multiple records in the destination grid.
Yes	No	If you select a record in the source data grid, then the destination data grid is populated and you can select a record in the destination grid. If you select multiple records in the source data grid, then the destination data grid is not updated and results in an invalid scenario.
Yes	Yes	If you select a record or multiple records in the source data grid, then the destination data grid is populated and you can select a record or multiple records in the destination grid.

Configure Grids

Use **Grid Configuration** to configure a data source for populating a data grid in forms.

To add a Grid Configuration

1. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Grid Configuration.**

   The **Grid Configuration** page appears.
2. Click **New**, and then click **Grid Configuration**. The **New Grid Configuration** dialog box appears.
3. In the **Title** box, type a name for the grid configuration, and in the **Description** box, type a description for the grid configuration.

   Note: In the **Title**, the allowed special characters are **_** and **-** only. In the **Description**, the allowed special characters are **- _ .** and **,** only.
4. In the **Hierarchical Level** drop-down list, select the number of levels you want the **Data Grid** control to display.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Click the **More** button for the **Data Source**, and then in the **Data Source** list, select the the data source. For the first data source all the lookups appear, but for the data sources after the first, only the parameterized lookups appear in the list.
6. In the **Display Columns** list, click the column names to show in the data grid. Double-click the column name to set the grid captions.

   To provide run time localized values, see Localize Column Names in Data Grid

   Note: For Grid Captions, the special characters **< > [ ] ‘ “ &** are not allowed.
7. In the **Persist Columns** list, click the column names to store in the instance XML of the form.
8. Do one of the following.

Task	Action
• Show check box for the records in the grid.	• Click **Yes** for the **Show Check Box**.
• Hide check box for the records in the grid.	• Click **No** for the **Show Check Box**.

9. Select Allow Selection option to enable or disable row selection in the Data Grid through:
   a. Mere selection (that is click to select the row)
   b. Check boxes (that is select checkboxes to select rows)

   If you select **Allow Selection** for a data source lookup, the user can select the row through mere selection or through check boxes, and the columns selected in the **Persist Columns** field are persisted in the form instance XML. The checkboxes are disabled for the rows in the parent grid, and no checkboxes are available for the rows in the child grid.

   Select **Disable Selection** if you do not want to persist any values in the form instance XML and instead want to use the Data Grid as a reference grid. The user can select the row through mere selection at that level for the Data Grid and only at the last level for the Hierarchical Data Grid.
10. If the **Hierarchical Level** is more than 1, click **Parameter Mapping**.

    The **Parameter Mapping** window appears.

    To map the parameters of the lookups, select the parameters in the drop-down list.
11. Click **Save** to save the parameter mapping. The **Parameter Mapping** window is closed.
12. Click **Advanced Settings** to highlight records, if required.

    For more information, see Highlight Records.
13. Click **Save**. The grid configuration is saved.
14. Click **Close**. The **New Grid Configuration** dialog box is closed.

Note:
- Database column names with space or special characters are not supported. This is due to the limitation in grid control of **Kendo UI for jQuery**.

- If a data source lookup, value field for a lookup, or a parameter associated with a grid configuration changes, then you must reconfigure the Grid and save the changes again.

- The Hierarchical Data Grid is not supported in mobile devices.

- When the database is not available, the tree lookup item will be disabled and cannot be selected.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The default number of database connection entries for each repository is 1. If you want to change the entry, override the **NumberOfDatabaseConnectionRetries** entry in the **Properties** column in the **SKRepository** table of the Farm database.

For example, to override the number of database connection entries to 20, change **[CDATA[0]] to [[CDATA[20]]** for **NumberOfDatabaseConnectionRetries** as follows.

```
<property key="NumberOfDatabaseConnectionRetries"><![CDATA[20]]></property>
```

- Backward compatibility exists for grid configuration, with support for column header localization based on column names, when the localization key is not specified with "@@" prefix and suffix.

- When you edit the grid configuration, the errors are logged in the Browser Console due to the usage of third-party controls in the page.

- The Persist columns should have data types such as numbers (without decimals), GUID, and string.

Update Grid Configuration for Lookup Modification

The **Requires Update** column in the **Grid Configuration** page displays the value as **0** or **1** to indicate if you need to revisit and update the grid configuration based on the Lookup modification. By default, it is set to **0** and indicates that no update is required. The value is set to **1**, if there is any change in the Lookup configuration.

The **Requires Update** column is set to **1,** when the Grid Configuration's **Display Columns**, **Persist Columns**, **Parameter Mapping**, and/or **Highlighting Records** settings are affected with the Lookup configuration change.

To update Grid Configuration for Lookup Modification

1. In the **Enterprise Console**, click menu icon, click **Settings**, click **Data Grid**, and then click **Grid Configuration**. The **Grid Configuration** page appears.
2. Right-click on a grid configuration for which the **Requires Update** column displays the value **1**, and then click **Edit** (or click on a grid configuration, and then on the action bar, click **Edit**). The **Edit Grid Configuration** dialog box appears.
3. Update the **Display Columns**, **Persist Columns**, **Parameter Mapping**, and **Highlighting Records** settings whichever applicable and then save the Grid Configuration.

   The Grid Configuration is saved with the updated Lookup configuration.

Highlight Records

Use **Highlight Cells/Rows** to highlight cells or rows of a data grid based on business conditions.

To highlight Cell or Row

1. In the **Enterprise Console**, click menu icon, click **Settings**, click **Data Grid**, and then click **Grid Configuration**. The **Grid Configuration** page appears.
2. Right-click on a grid configuration, and then click **Edit** (or click on a grid configuration, and then on the action bar, click **Edit**). The **Edit Grid Configuration** dialog box appears.
3. Click **Advanced Settings**. The **Advanced Settings** dialog box appears.
4. Do one of the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
• Apply fore color for a cell in the grid.	• In the **Apply To** list, click **Cell ForeColor**.
• Apply back color for a cell in the grid.	• In the **Apply To** list, click **Cell BackColor**.
• Apply fore color for a row in the grid.	• In the **Apply To** list, click **Row ForeColor**.
• Apply back color for a row in the grid.	• In the **Apply To** list, click **Row BackColor**.

5. In the **Color** list, click a color or enter hexadecimal code of the color.
6. In the **Data Source** list, click a column of the lookup.
7. In the **Operator** list, click an operator. The operators are populated based on the data type of the selected column of the lookup.

Task	Action
Boolean	• True • False
Date	• After • Before • Last Month (considers days in the previous month, that is from the first day of the previous month to the last day of the previous month) • Last X Years (considers days in the past years, that is from the last day of December of the previous year to the first day of January of the current year minus X) • Last Year (considers days in the last year, that is from the first day of January to the last day of December of the last year) • Next Month (considers days in the next month, that is from the first day of the next month to the last day of the next month) • Next X Years (considers days in the future years, that is from the first day of January of the next year to the last day of December of the current year plus X) • Next Year (considers days in the next year, that is from the first day of January to the last day of December of the next year) • On Or After • On Or Before • This Month (considers days in the current month, that is from the first day of the month to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
	the last day of the month) • This Week (considers days under the current ISO 8601 week number) • This Year (considers days in the current year, that is from the first day of January to the last day of December of the year) • Today • Tomorrow • Yesterday
Numeric	• Equals • Not Equals • Greater Than • Greater Than or Equal To • Lesser Than • Lesser Than or Equal To
String	• Begin With • Does Not Begin With • Contains Data • Does Not Contain Data • Ends With • Does Not Ends With • Equals • Not Equals

8. In the **Value** box, type a value for the column.
9. Click the **Add Row** button to add an additional row if required.
10. Click **Save**. The conditions for highlighting records are saved.
11. Click **Save**. The **Edit Grid Configuration** dialog box is closed.

Note:
- Database column names with special characters except underscore are not supported.
- The appropriate data type must be set for the selected columns of the Web API lookup in the **Lookup Settings**. Otherwise, only operators for the **String** data type will be available in the **Operator** list for the column.
- Boolean value can be entered as `0` or `false` for false value and `1` or `true` for true value.
- Date and time values must be set as per the server culture in the format, MM/DD/YYYY. The date time values should only contain date part.
- Date is not validated and must be entered only in `MM/DD/YYYY` format without the preceding zeros. For example, **September 8, 2007** should be entered as `9/8/2007` and not `09/08/2007`.
- Value column does not support single quotation mark (') special character.

When multiple conditions are specified the last specified condition takes precedence.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- To ensure optimal performance, restrict the total number of conditions to a maximum of 10. The condition is applied only if the column is displayed in the grid.

Data Lookup

Use **Data Lookup** control to search for data from a data source in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note**: - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Lookup Source	Name of the item in the **Lookup List**. **To add a name of the item in the Lookup List for the control** 1. Click the **Lookup Source** button. The **Lookup**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Source Configuration page appears. 2. In the **Data Source Type** list, click a data source type. 3. In the **Data Source box**, click a data source. 4. In the **Parameters** list, click and select a control on the form, if applicable. 5. Click **Save & Continue**. The selected name is set for the control and the **Lookup Source Configuration** page is closed. **Note:** - For **Parameters**, the **Title** of the control on the form is displayed, if set for the control. Otherwise, the **XML Node** of the control on the form is displayed. - For information about Lookup List, see List Lookups.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Read-Only	Allow or deny a user to enter a value for the control. • Select **Yes** to deny a user to enter a value for the control. • Select **No** to allow a user to enter a value for the control. The default value is **No**.
Label Position	Position of the label for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Suppress**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Watermark	Hint text that appears when no value is entered for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Show as Drop-Down	Enable or disable the control to show as Drop-Down. • Select **Yes** to enable the control to show as Drop-Down. • Select **No** to disable the control to show as Drop-Down. The default value is **No**. **Note:** If you select **Yes**, then a user will not be able to type in the control at run-time.
Drop-Down Height	Drop-down height in pixels for the control. The default value is **300**. **Note:** - The minimum value for drop-down height is 300. The value below 300 is not considered at runtime. - If the value is 0, then the drop-down height of the control is set as per the default values.
Adapt UI	Change the user interface of the control based on the display width. • Select **Yes** to allow the system to change the user interface of the control based on the display width. The **Web View** is rendered for the control

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	when the screen width is greater than or equal to 800 pixels. Otherwise, the **Mobile View** is rendered for the control. • Select **No** to deny the system to change the user interface of the control based on the display width. In this case, the **Web View** is rendered for the control. The default value is **Yes**. **Note:** The responsiveness of the container depends on the value set for the **Desktop View Only** property at the Form level.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**. **Note:**If you configure a **Data Lookup** control as mandatory, then when you select an item from the drop-down, the mandatory message appears. The message disappears after you have selected an item.
Autocomplete Length	Number of characters to start autocomplete. The default value is **3**. **Note:** If the value is 0, then the autocomplete length for the control is set as per the default values.
Show Confirmation Message	Show or hide the confirmation message for clearing the value of the control in devices.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Yes** to show the confirmation message for the control. • Select **No** to hide the confirmation message for the control. The default value is **Yes**.
Confirmation Message Text	Confirmation message for clearing the value of the control in devices. The default value is **Are you sure you want to clear the value?** **Note:** This property is applicable only when the **Show Confirmation Message** property is set to **Yes**.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Default Number of Records	Number of records added at run-time for the control. The default value is **50**. **Note:** If the value is 0, then the default number of records for the control is set as per the default values.
Retain Selected Value	Keep or lose the selected value for the control when the current parameter values from the source get changed. • Select **Yes** to keep the selected value for the control. • Select **No** to lose the selected value for the control. The default value is **No**.
Resolve Value in Edit Mode	When the Form is in the edit mode like opening a Form via WorkItem in Inbox, then update or keep the Lookup display value and the extended information (if the property Extended Information is Yes) based on

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	the selection for the control. • Select **Yes** to update the Lookup Display Value and the Extended Information. • Select **No** to keep the existing Lookup Display Value and the Extended Information. The default value is **No** for performance improvement.
Persist Display Value	Keep or lose the display value for the control on submission of the form. • Select **Yes** to keep the display value for the control. • Select **No** to lose the display value for the control. The default value is **Yes**.
Extended Information	Allow or deny a user to access the extended information (that is columns other than the display and value columns of the result set) based on the selection for the control. • Select **Yes** to allow a user to access the extended information based on the selection for the control. • Select **No** to deny a user to access the extended information based on the selection for the control. The default value is **No**.
Include in Summary	Allow or deny a user to view the value in the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the Summary of a Base Form. • Select **No** to deny a user to view the value in the Summary of a Base Form. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Data Lookup control.

Note:
- The format and culture for date and time value displayed in the control is as per the current server and not as stored in the database.
- On Web, autocomplete uses the **Starts With** filter to complete the string, and on AVEVA Work Tasks Pro app, autocomplete uses the **Contains** filter to complete the string.
- When the data type is not a string but a number, date, time and so on, the data type is first converted to string for comparison and then filtered accordingly.
- When **Persist Display Value** is set to **No**, date and date time values are not supported as value fields of the lookup.
- The default data lookup time out is 50 seconds. You can change this value in the **web.config** file.

```
<appSettings>
<add key="lookupTimeout" value="50" />
</appSettings>
```

- When **Data Lookup** is the first control on a form and the form is viewed on a mobile device, then the lookup with populated data (ready for entry or selection) is shown instead of the complete form with other controls.
- Ensure that you map a **Number** control to an **Integer** or **Numeric** type of Data Grid parameter.

Keyboard-based Navigation

You can use the keyboard to perform the following actions:

- To open the drop-down list,
  - use the **Alt** + **Down Arrow** keys (in case, **Yes** option is selected for the **Show as Drop-Down** property)
  - type the search string (in case, **No** option is selected for the **Show as Drop-Down** property)
- To navigate and highlight the listed option, use the **Up** and **Down Arrow** keys.
- To load more options in the list,
  - use the **Alt** + **Page Down** keys for Windows 10 and Windows 2016 machines
  - use the **Shift + Alt + Page Down** keys for Windows 11 machine
- To select highlighted option in the list, use the **Enter** key.
- To close the list, use the **Esc** key.

Date

Use **Date** control to select a date in your form.

Properties

The properties for the control are grouped under the following tabs:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Default Value	The default value for the control. **Note:** Ensure the value is within the **Minimum Value** and the **Maximum Value**.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Read-Only	Allow or deny a user to enter a value for the control. • Select **Yes** to deny a user to enter a value for the control. • Select **No** to allow a user to enter a value for the control. The default value is **No**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is Default. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	control. The default value is **No**.
Minimum Value	Minimum date value for the control. The default value is **01/01/1901**.
Maximum Value	Maximum date value for the control. The default value is **12/31/2099**.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Date Format	Date format for the control. For example, MM/dd/yyyy. Always use M for month. The default value is taken from the current server culture. For more information, see Date and Time Formats. **Note:** Date format must contain at least month, date, and year components. Date format is not validated.
Forms 2011 Compatible	Make the control compatible with Forms 2011, that is the forms used in BPM 2011. • Select **Yes** to allow the system to make the control compatible with Forms 2011. • Select **No** to deny the system to make the control compatible with Forms 2011. The default value is **No**. **Note:** When a control on a form is made Forms 2011 compatible, the XML instance of the form with the control is changed to ensure compatibility with Forms 2011.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Include in Summary	Allow or deny a user to view the value in the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the Summary of a Base Form. • Select **No** to deny a user to view the value in the Summary of a Base Form. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Date control.

Date Time

Use **Date Time** control to select a date and time in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Default Value	The default value for the control. **Note:** Ensure the value is within the Minimum Value and the Maximum Value.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Read-Only	Allow or deny a user to enter a value for the control. • Select **Yes** to deny a user to enter a value for the control. • Select **No** to allow a user to enter a value for the control. The default value is **No**.
Label Position	Position of the label for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select Left to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: If the value is 0, then the width of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**.
Minimum Value	Minimum date and time value for the control.
Maximum Value	Maximum date and time value for the control.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Date Format	Date format for the control. For example, MM/dd/yyyy. Always use M for month. The default value is taken from the current server culture. For more information, see Date and Time Formats. **Note:** - Date format must contain at least month, date, and year components. - Date format is not validated.
Time Format	Time format for the control. For example, hh:mm:ss tt. Use HH for 24-hour time. The default value is taken from the current server culture. For more information, see Date and Time Formats. **Note:** - Time format must contain at least: --- Hour and minute components for 24-hour time. --- Hour, minute, and period components for 12-hour time. - Time format is not validated. - For cultures, which do not support AM/PM, the "tt" or "t" must not be specified in the format.
Time Interval	Interval between two consecutive time slots to show in the list for the control.
Include in Summary	Allow or deny a user to view the value in the **Summary** of a **Base Form**. • Select **Yes** to allow a user to view the value in the Summary of a Base Form. • Select **No** to deny a user to view the value in the Summary of a Base Form. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scripts

For information about using scripts, see scripting guidelines for Date Time control.

Note:
- Value is saved in the Universal Coordinated Time (UTC) format.
- The Display Values are based on the browser time zone.
- Format and culture are taken from the current server.
- However, if the **Date Format** and the **Time Format** are specified, then the given formats are used but applied using the current server culture.

Drop-Down

Use **Drop-Down** control to select an option from a list in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Options Caption	Header for the options for the control.
Options	Options for the control. **To add an option to the control** 1. Click **Add Record**. A new record appears. 2. In the **Text** box, enter the options text. 3. In the **Value** box, enter the options value. 4. In the **Fore Color** palette, click a color (or in the **Color Code** box, enter hexadecimal code of the color), and then click **Apply**. The selected color is applied to the foreground of the options text. 5. In the **Back Color** palette, click a color (or in the **Color Code** box, enter hexadecimal code of the color), and then click **Apply**. The selected color is applied to the background of the options text. **To clear fore color for an option of the control** • On the **Fore Color** palette, in the **Color Code** box, enter **#1b1b1b**, and then click **Apply**. The **Fore Color** is reset to the default color. **To clear back color for an option of the control** • On the **Back Color** palette, in the **Color Code** box, enter **#1b1b1b**, and then click **Apply**. The **Back Color** is reset to the default color. **To delete an option from the control** • Click the **Delete** button. The record is deleted. **Note:** - The options **Text** and **Value** parameters are mandatory and must be unique across all options. - The **Fore Color** and **Back Color** properties are currently not supported in devices.
Default Value	The default value for the control. **Note:** The value must be one of the values set for the **Value** parameter in the **Options** property.

Appearance

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is Default. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is **No**. **Note:** Validation message appears only on submission of the form.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Text Parameter	Name that identifies the options text to use in scripts for the control.
Value Parameter	Name that identifies the options value to use in scripts for the control.
Include in Summary	Allow or deny a user to view the value in the **Summary** of a **Base Form**. • Select **Yes** to allow a user to view the value in the **Summary** of a **Base Form**. • Select **No** to deny a user to view the value in the **Summary** of a **Base Form**. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Drop-Down control.

Email

Use **Email** control to enter email addresses in your form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Default Value	The default value for the control. **Note:** Ensure the value is within the **Maximum Length**. Multiple values must be separated with a semicolon.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Read-Only	Allow or deny a user to enter a value for the control. • Select **Yes** to deny a user to enter a value for the control. • Select **No** to allow a user to enter a value for the control. The default value is **No**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Watermark	Hint text that appears when no value is entered for the control.
Width	Width in pixels for the control. The default value is **0**. Note: If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**.
Maximum Length	Maximum length of the value for the control. The default value is **0**. **Note:** If the value is 0, then the maximum length for the control is not set.
Multiple Email IDs	Allow or deny a user to enter multiple email IDs for the control. • Select **Yes** to allow a user to enter multiple email IDs for the control. • Select **No** to deny a user to enter multiple email IDs for the control. The default value is **No**.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Include in Summary	Allow or deny a user to view the value in the **Summary** of a **Base Form**. • Select **Yes** to allow a user to view the value in the Summary of a Base Form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **No** to deny a user to view the value in the Summary of a Base Form. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Email control.

Embed Form

Use **Embed Form** control to use the entire form as a part of another form. You can design the common section in a Form and use it in other forms through the Embed Form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. The default value is the value for the **Name** property without the spaces. **Note:** If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Form Details	Details of the form that is to be linked. **To add Form Details for the control** 1. Click the **Form Details** button. The **Form Details Configuration** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	2. In the **Form Name** box, click a form name. 3. In the **Form Version** box, click a form version. 4. Click **Save & Continue**. The selected form details is set for the control, and the **Form Details Configuration** page is closed. **Note:** - **Embed Form** control does not support forms with **Embed Form** controls. -We recommend using unique XML nodes for controls in source forms to aid in scripting. - We recommend using the `control.findByXmlNode()` method instead of the `control.findById()` method in source forms to avoid conflict of IDs between forms. -The definition of the source form is fetched and used for the embed form. The source form is no longer referenced after the properties of the control are saved. Therefore, the source form does not have to be available in the repository after the properties are configured. - If both the main and embed form contains pop-up control, then ensure that the IDs are different. - The draft version of the selected form is considered.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: If the value is 0, then the height of the control is set as per the default values.
Consider Available Height	Resize the height of the control or container based on the available height. • Select **Yes** to resize the height based on available height. • Select **No** to retain the space. The default value is **No**. **Note:** - The Consider Available Height property is not applicable for the following: -Designer view -Print form -Panel form with horizontal tab order - If this property is set to **Yes** and a value is specified in the **Height** property, minimum height is considered. - This property is applicable only in web view. - When more than one control or container is set with **Consider Available Height** at the same level of container or form, then the available space is distributed evenly between those controls or containers. For example, consider a Form which contains two Panel forms with **Consider Available Height** set to **Yes**, where Panel form 1 has two TextInput controls and Panel form 2 has one TextInput control. The Panel form 1's height is more than Panel form 2's height as Panel form 1 contains two TextInput controls and the available empty space in the form is evenly distributed between the two Panel forms.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Show Panel Details	Show or hide the Title, Description, and Margin. • Select **Yes** to show the panel details.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **No** to hide the panel details. The default value is **Yes**.
Adapt UI	Change the user interface of the control based on the display width. • Select **Yes** to allow the system to change the user interface of the control based on the display width. The **Web View** is rendered for the control when the screen width is greater than or equal to 800 pixels. Otherwise, the **Mobile View** is rendered for the control. • Select **No** to deny the system to change the user interface of the control based on the display width. In this case, the **Web View** is rendered for the control. The default value is **Yes**. **Note:** The responsiveness of the container depends on the value set for the **Desktop View Only** property at the Form level.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.

Scripts

For information about using scripts, see scripting guidelines for Embed Form control.

Note:
- This control might impact performance as the data loaded determines the performance of the control.
- If the form specified in the **Embed Form** control, contains an **On Form Load** script, then the script does not get executed. To overcome this, use a **Hidden Field** control with the script added to the value script.
- **Embed Form** control does not support forms with **Embed Form** controls.
- **Embed Form** is not supported in **Panel Form** when with **Tab Order** set to **Horizontal**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Embed Form** is not supported for MQTT.

Embed Page

Use **Embed Page** control to show a webpage in your form.

Note:
- The Embed Page control uses an iframe to load the url. If the source Url has an HTTP Response header X-Frame-Options, it might not load in the Embed Page control. Refer to X-Frame-Options documentation on the W3C website.
- This control is not supported in AVEVA Work Tasks Pro.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
Description	Description for the control.
URL	URL for the webpage for the control. Start the URL with **http://** or **https://** or the path with the **<BPMUITemplates>** tag. The **<BPMUITemplates>** tag replaces the path with the application path including the Default folder. For example, **<BPMUITemplates>/Report.html** is replaced as **/EnterpriseConsole/BPMUITemplates/Default/Report.html**

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Top**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Consider Available Height	Resize the height of the control or container based on the available height. • Select **Yes** to resize the height based on available height. • Select **No** to retain the space. The default value is **No**. **Note:** - The Consider Available Height property is not applicable for the following: -Designer view -Print form -Panel form with horizontal tab order - If this property is set to **Yes** and a value is specified in the **Height** property, minimum height is considered. - This property is applicable only in web view. - When more than one control or container is set with **Consider Available Height** at the same level of container or form, then the available space is distributed evenly between those controls or containers. For example, consider a Form which contains two Panel forms with **Consider Available Height** set to **Yes**, where Panel form 1 has two TextInput controls and Panel form 2 has one TextInput control. The Panel form 1's height is more than Panel form 2's height as Panel form 1 contains two TextInput controls and the available empty space in the form is evenly distributed between the two Panel forms.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is **No**.
Show Scrollbars	Show or hide scrollbars for the control. • Select **Yes** to show scrollbars for the control. • Select **No** to hide scrollbars for the control. The default value is **Yes**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. On Data Change script always works with the value script for the Embed Page. To set the Embed Page Url and also trigger the Embed Page's On Data Change script, see Set Embed Page Url and Trigger the On Data Change Script. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.

Scripts

For information about using scripts, see scripting guidelines for Embed Page control.

Note: This control might impact performance as the data loaded determines the performance of the control.

Set Embed Page Url and Trigger the On Data Change Script

Use the below script to set the Embed Page Url and also trigger the Embed Page's On Data Change script.

```
// Below line would set the url of the Embed Page to load
control.findById("E1").url = "http://www.skelta.com";
// Below line would set the value of the Embed Page control, so that
// the On Data Change script of the Embed Page is fired.
control.findById("E1").value = "http://www.skelta.com";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

E-signature

Use **E-signature** control to authenticate the identity of a user who signed the document and to confirm the integrity of the signed document in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. The default value is the value for the **Name** property without the spaces. **Note:** If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Show Header	Show or hide the header for the control. • Select **Yes** to show the header for the control. • Select **No** to hide the header for the control. The default value is **Yes**.
Signed By Text	Signed By text for the control.
User Name Text	User Name text for the control.
Password Text	Password text for the control.
Signed On Text	Signed On text for the control.
Comments Text	Comments text for the control.
Image Path	Path to the image for the control. The image path is specified as: • A URL starting with **http://** or **https://**. • An image path tag. For information about using image path tags, see *Image Path Tags* in Image. • A path to the **BPMUITemplates** folder as **BPMUITemplates\Default\Images**. **Note:** - You can use relative paths as follows: --- Use **.** for the current folder where the html page resides. --- Use **..** for the folder above the current folder where the html page resides.
Adapt UI	Change the user interface of the control based on the display width. • Select **Yes** to allow the system to change the user interface of the control based on the display width. The **Web View** is rendered for the control when the screen width is greater than or equal to 800 pixels. Otherwise, the **Mobile View** is rendered for the control. • Select **No** to deny the system to change the user interface of the control based on the display width. In this case, the **Web View** is rendered for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is **Yes**. **Note:** The responsiveness of the container depends on the value set for the **Desktop View Only** property at the Form level.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Comments Required	Enforce or bypass entering comments for the control. • Select **Yes** to enforce entering comments for the control. • Select **No** to bypass entering comments for the control. The default value is **No**.

Advanced

Property	Description
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Date Format	Date format for the control. For example, MM/dd/yyyy. Always use M for month. The default value is taken from the current server culture. For more information, see Date and Time Formats. **Note:** Date format must contain at least month, date, and year components. Date format is not validated.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Time Format	Time format for the control. For example, hh:mm:ss tt. Use HH for 24-hour time. The default value is taken from the current server culture. For more information, see Date and Time Formats. **Note:** - Time format must contain at least: --- Hour and minute components for 24-hour time. --- Hour, minute, and period components for 12-hour time. - Time format is not validated.
User Mode	Number of users required to sign using the control. • Select **Single** User to allow only one user to sign using the control. • Select **Multiple** User to allow multiple users to sign using the control. The default value is **Single User**.

Scripts

For information about using scripts, see scripting guidelines for E-signature control.

Note:
- Only one **E-signature** control is supported per form.
- Use **Pop-Up** control to apply e-signature multiple times or use progressive forms for multi-stage approvals.
- The E-signature control is not supported when you access AVEVA Work Tasks Pro through Azure Active Directory authentication or AVEVA Connect.

Gantt Chart

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	(Required) XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Use forward slash (**/**) for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use a forward slash (**/**) to denote a parent, which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it may affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Lookup Source	(Required) Name of the item from the lookup list. To add a name of the item for the control from the lookup list: 1. Click the **Lookup Source** button. The Lookup Source Configuration page appears. 2. In the **Data Source Type** list, select a type of data source. **Note**: Supported datasource types are database and Web API lookup source. 3. In the **Data Source** box, select a data source. 4. In the **Parameters** section, select the parameters to include, if applicable. 5. Click **Save & Continue**. The selected name is set for the control and the Lookup Source Configuration page is closed.
Event State	Column name that corresponds to the state of an event in the selected lookup source. The maximum number of displayed characters for the value of this property is 25. If the value exceeds the character limit, an ellipsis is added and the full column value is shown in a tooltip on mouse hover. **Note**: This property is shown in the details pane with the **Event Color** as part of the header for a Gantt Chart's selected segment. In the body of the details

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	pane, the **Event Start Time**, **Event End Time,** and **Event Name** properties are shown.
Event Start Time	Column name that corresponds to an event's start time in the selected lookup source. **Note**: This must be in UTC format in the source and in the lookup configuration, its data type must be set to DateTime.
Event End Time	Column name that corresponds to an event's end time in the selected lookup source. **Note**: This must be in UTC format in the source and in the lookup configuration, its data type must be set to DateTime.
Event Name	Column name that corresponds to the name or description of an event in the selected lookup source. The maximum number of displayed characters for the value of this property is 25. If the value exceeds the character limit, an ellipsis is added and the full column value is shown in a tooltip on mouse hover.
Event Color	Column name that corresponds to the color of an event in the selected lookup source.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note**: If the value is 0, then the width of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Bar Thickness	Thickness of the Gantt bar in pixels. The default value is 0. The value other than the default must be from 10 to 150 pixels.
Time Period	This controls the period of data to be displayed in the Gantt Chart control. • Select **Duration** to specify a time period in hour. This is based on the last segment's end time relative to now. If this is selected, the **Duration** property is shown after the **Time Period** property with 8 hours as the default value. Otherwise, **Duration** property is hidden. • Select **Time Range** to enable the use of **ganttTimeRangeStartTime** and **ganttTimeRangeEndTime** properties in scripting. You can use the **Time Range** start and end time properties with **Date Time** or **Text** controls when scripting. For example scripts, see *Scripting Guidelines* for **Time Period** property. For more information on Date Time and Text controls, see Date Time and Text. The default value is **Duration**.
Duration	See **Time Period** property. **Note**: If this property is updated through scripting or **refreshData()** is triggered, the Gantt Chart control is redrawn. Currently selected segment and its details pane are no longer shown.
Show Details Pane	Show or hide the Gantt Chart segment's details pane. • Select **Yes** to show the details pane when a segment in the control is selected. • Select **No** to hide the details pane and remove the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	space it occupies. The default value is **Yes**.
Show Time Range	Show or hide the start and end date/time. • Select **Yes** to show the start and end date/time in the control. • Select **No** to hide the start and end date/time in the control. The default value is **No**. The start time is calculated based on the specified value in the **Duration** property subtracted from the last segment end time (end time - duration).
Title Position	Position of the title for the control. • Select **Left** to align the title to the left of the control. • Select **Right** to align the title to the right of the control. • Select **Suppress** to hide the title for the control. The default value is **Left**.
Title	Title of the Gantt control. It is shown directly above the top-left corner of the control. This supports two lines with 35 characters for each line. An ellipsis is added if it exceeds the 35-character limit. If you want to add a line break, use the html tag ` `. **Note:** If this property is not configured or blank, nothing is shown, the other control contents use its space, and the control size is adjusted based on the available content within the control. If this property is configured for column bindings, even if the configured data column returns a blank value, where nothing is shown, the space is retained.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	contain only alphanumeric characters.

Advanced

Property	Description
On Click	Run a script after the control is clicked.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Enable Refresh	Enable or disable control auto refresh. • Select **Yes** to enable control refresh and configure the **Refresh Rate** property. • Select **No** to disable control refresh. The default value is **No**. **Note**: This property will not work on the control even if it is set to **Yes** if the control segment's details pane is shown.
Refresh Rate	This is how often the control display is refreshed, in seconds, to fetch value from the data source. **Note**: This is only visible if the **Enable Refresh** property is set to **Yes**. The default value is 30 seconds and the valid refresh rate is from 10 to 600 seconds.
Enable Action (1 - 3)	Call the script defined in a particular **Action Script** property in the **Scripts** tab. The default value is **No**. If you keep the default, the action button is not visible in the segment details window. If you set this to **Yes**, its corresponding **Action** property appears, where you can specify the name or label of that particular action.
Action (1-3)	Name or label for **Action (1-3).** The maximum number of displayed characters is 8. If it exceeds the character limit, an ellipsis is added and the full label is shown in a tooltip on mouse hover.

Scripts

For information about using scripts, see scripting guidelines for Gantt Chart control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

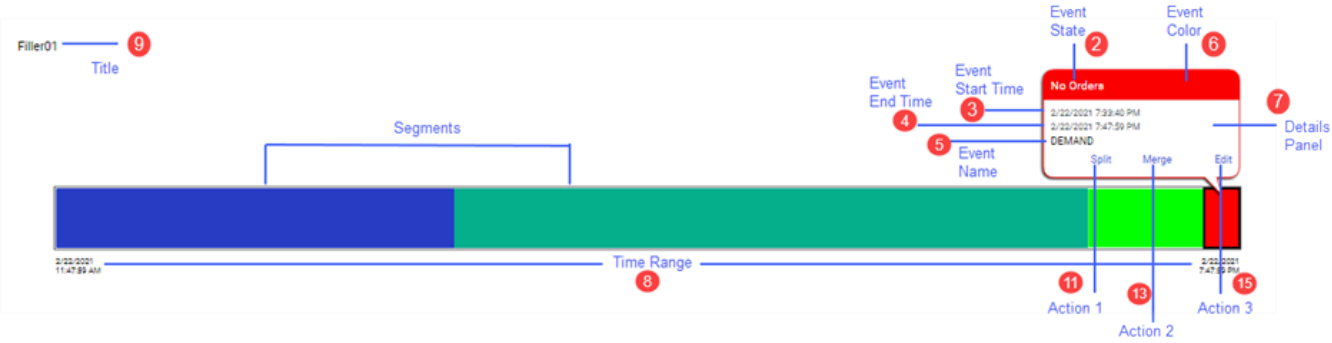
Example Gantt Chart Rendering

The following rendering example for Gantt Chart control are based on the configuration in the control's properties page. The configured properties are listed in the table and most of the properties that are kept to their default values are no longer mentioned.

Example configuration and rendering

Basic	Appearance	Advanced
1. **Lookup Source**: `entity_23_B` 2. **Event State**: `reas_desc` 3. **Event Start Time**: `event_time_utc` 4. **Event End Time**: `event_end_time_utc` 5. **Event Name**: `state_desc` 6. **Event Color**: `state_color`	7. **Show Detail Panel**: `Yes` (selected) 8. **Show Time Range**: `Yes` (selected) 9. **Title**: `Filler01`	10. **Enable Action 1**: `Yes` (selected) 11. **Action 1**: `Split` 12. **Enable Action 2**: `Yes` (selected) 13. **Action 2**: `Merge` 14. **Enable Action 3:** `Yes` (selected) 15. **Action 3**: `Edit`

The data displayed in the segment details window is from the defined lookup source and the mapped column name for each of the displayed properties.



When you select a Gantt Chart segment, it is highlighted with a black border, and the corresponding segment details pane appears as shown in the example rendering. You can click the segment again, or click anywhere within the area of the control if you want to hide the segment details and deselect the segment.

Gauge

Use **Gauge** control to display data in a gauge in your form.

Note: This control is not supported in AVEVA Work Tasks Pro.

Properties

The properties for the control are grouped under the following tabs:

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.

Appearance

Property	Description
Visible	Show or hide the **control**. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is **No**.
Alignment	Alignment of the control with respect to the form. • Select **Left** to align the control to the left of the form. • Select **Right** to align the control to the right of the form. • Select **Center** to align the control to the center of the form.
Scale Minimum Value	Minimum value displayed on the scale. The default value is 0.
Scale Maximum Value	Maximum value displayed on the scale. The default value is 100.
Scale Start Angle	The angle at which the scale starts. The default value is -30 degrees.
Scale End Angle	The angle at which the scale ends. The default value is 210 degrees.
Scale Major Unit	Value of major tick marks on the scale. For example, if the **Unit of Measurement** is set to lb, and the **Scale Major Unit** is set to 50, then every major tick mark will denote a value of 50 lb. The gauge is limited to a maximum of 100 major tick marks at runtime.Therefore, the value for the major unit must be set such that no more than 100 units would be rendered on the gauge. The default value is 20.
Scale Minor Unit	Value of minor tick marks on the scale. For example, if the **Unit of Measurement** is set to lb, and the **Scale Minor Unit** is set to 10, then every minor tick mark will denote a value of 10 lb. The gauge is limited to a maximum of 200 minor tick marks at runtime.Therefore, the value for the minor unit must be set such that no more than 200 units would be rendered on the gauge. The default value is 10.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Value Position	Position of the value for the control. • Select **Top** to align the label to the top of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Bottom**.
Show Target Indicator	Show or hide the target indicator of the control. • Select **Yes** to show the target indicator of the control. • Select **No** to hide the target indicator of the control. The default value is **Yes**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
On Initialize	Run a script after the control has been initialized, but before it is rendered to the UI. For more information, see scripting guidelines for On Initialize property.
Range Color	Run a script to apply a color to a range of values on the the gauge. For more information, see scripting guidelines for Range Color property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Target Value	Target value of the gauge. This is the value that the Target indicator points at.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is 0.
Unit of Measurement	Unit of measurement for value of the control.
Decimal Places (Precision)	Decimal places for the value of the control. The default value is 0.

Note: The Gauge control is not supported in mobile devices and in responsive views in containers.

Scripts

For information about using scripts, see scripting guidelines for Gauge control.

Geolocation

Use **Geolocation** control to get the geographical position of the user.

Note:You need to enable the location services on your browser or device to access the Geolocation feature.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
Description	Description for the control.
Button Text	Text for the control.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	- This is applicable only when the **Label Position** is set to **Left** or **Right**.
Alignment	Alignment of the control with respect to the form. • Select **Default** to allow the system to align the control. The system aligns the control to the left of the form or as specified by the control. • Select **Left** to align the control to the left of the form. • Select **Right** to align the control to the right of the form. • Select **Center** to align the control to the center of the form.
Display Mode	Show text and/or image for the control. • Select **Text** Only to show only text for the control. • Select **Image** Only to show only image for the control. • Select **Text and Image** to show both text and image for the control. The default value is **Text Only**.
Button Size	Size of the button for the control. • Select **Standard** to set the size of the control to 120 x 32 pixels. • Select **Small** to set the size of the control to 30 x 30 pixels. • Select **Medium** to set the size of the control to 180 x 86 pixels. • Select **Large** to set the size of the control to 280 x 130 pixels. The default value is **Standard**.
Image Path	Path to the image for the control. The image path is specified as: • A URL starting with **http://** or **https://**. • An image path tag. For information about using

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	image path tags, see *Image Path Tags* in Image. • A path to the **BPMUITemplates** folder as **BPMUITemplates\Default\Images**. **Note:** You can use relative paths as follows: - Use **.** for the current folder where the html page resides. - Use **..** for the folder above the current folder where the html page resides. We recommend the following image sizes when the **Display Mode** property is set to Image. • **Small** – 30 x 30 pixels. • **Medium** – 180 x 86 pixels. • **Large** – 280 x 130 pixels. We recommend the following image sizes when the **Display Mode** property is set to Text and Image. • **Small** – 30 x 30 pixels. • **Medium** – 40 x 40 pixels. • **Large** – 50 x 50 pixels. **Note:** - This property is applicable only when the **Display Mode** property is set to **Image** or **Text and Image**, and the **Button Size** property is set to **Small**, **Medium** or **Large**. - If the **Display Mode** property is set to **Text and Image** and the **Button Size** property is set to **Small**, then only the image is displayed.
Image Position	Position of the image with respect to the text for the control. • Select **Left** to align the image to the left of the text for the control. • Select **Right** to align the image to the right of the text for the control. • Select **Top** to align the image to the top of the text for the control. • Select **Bottom** to align the image to the bottom of

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	the text for the control. The default value is **Left**. **Note:** This property is applicable only when the **Display Mode** property is set to **Text and Image**.
Show as Link	Enable or disable the control to show as link. • Select **Yes** to enable the control to show as link. • Select **No** to disable the control to show as link. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Form Button	Enable or disable the control to act as form button. When this property is enabled, the control is displayed at the bottom-right corner of the form. • Select **Yes** to enable the control to act as form button. • Select **No** to disable the control to act as form button. The default value is **No**. **Note:** If the value is **Yes**, and the form is viewed on a mobile device, then the control appears in the **Action Bar** of the mobile device.

Scripts

For information about using scripts, see scripting guidelines for Geolocation property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Hidden Field

Use **Hidden Field** control to submit custom data in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Default Value	The default value for the control.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Include in Summary	Allow or deny a user to view the value in

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the Summary of a Base Form. • Select **No** to deny a user to view the value in the Summary of a Base Form. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form..

Scripts

For information about using scripts, see scripting guidelines for Hidden Field control.

Note:
- **Hidden Field** control is visible on the form only at design-time.
- At run-time, the width of space occupied by **Hidden Field** controls on a record in the **Grid** view of a **Base Form** is distributed proportionately to other controls on the record of the **Base Form**.

Hierarchical List

Use **Hierarchical List** control to view a hierarchical list in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
Description	Description for the control.
Lookup Source	Name of the item in the **Lookup List**. **To add a name of the item in the Lookup List for the control** 1. Click the **Lookup Source** button. The **Lookup**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Source Configuration page appears. 2. In the **Data Source Type** list, click a data source type. 3. In the **Data Source box**, click a data source. 4. In the **Parameters** list, click and select a control on the form, if applicable. 5. Design your hierarchical list by dragging and dropping the column names. 6. Click **Save & Continue**. The selected name is set for the control and the **Lookup Source Configuration** page is closed. **Note:** - Only non-parameterized Web API lookups are supported. - The first node is used for generating the schema. If the elements are missing, it will not be shown in the Lookup Source Configuration. - For information about Lookup List, see List Lookups.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	the control. • Select **Suppress** to hide the label for the control. The default value is **Top**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **450**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Default Number of Records	Number of records added at run-time for the control. The default value is **50**. **Note:** If the value is 0, then the default number of records for the control is set as per the default values.

Scripts

For information about using scripts, see scripting guidelines for Hierarchical List control.

Hyperlink

Use **Hyperlink** control to create a link to a URL in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Hyperlink Text	Clickable text in the hyperlink for the control.
URL	URL for the hyperlink for the control. Start the URL with **http://** or **https://** or the path with the **<BPMUITemplates>** tag.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Include in Summary	Allow or deny a user to view the value in the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the **Summary** of a **Base Form**. • Select **No** to deny a user to view the value in the **Summary** of a **Base Form**. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Hyperlink control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Image

Use **Image** control to insert an image in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Image Path	Path to the image for the control. The image path is specified as: • A URL starting with **http://** or **https://**. • An image path tag. For information about using image path tags, see *Image Path Tags*. A path to the **BPMUITemplates** folder as **BPMUITemplates/**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Default/NextGenForms/custom/Images/. **Note**: - If the image name is tagged with **FormCustomPath:**, the location is **<Installpath/Web/BPMUITemplates/Default/NextGenForms/custom/images/MyImage.png** - If the image name is tagged with **FormCustomThemePath:**, the location is **<Installpath/Web/BPMUITemplates/Default/NextGenForms/custom/Themes/<ThemeName>/Images/MyImage.png** - If the image name is not tagged with **FormCustomPath:** or **FormCustomThemePath:**, the location of image of earlier version will be considered i.e. **<Installpath>/Web/BPMUITemplates/Default/Themes/<ThemeName>/NextGenForms/images** folder. - For AVEVA Work Tasks Pro, mapping of themes is not considered. - You can use relative paths as follows: --- Use **.** for the current folder where the html page resides. --- Use **..** for the folder above the current folder where the html page resides.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Suppress**. **Note:** -If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Alignment	Alignment of the control with respect to the form. • Select **Default** to allow the system to align the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	control. The system aligns the control to the left of the form or as specified by the control. • Select **Left** to align the control to the left of the form. • Select **Right** to align the control to the right of the form. • Select **Center** to align the control to the center of the form.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Downscale Image	Reduce size of the image for the control based on the display width. • Select **Yes** to allow the system to reduce size of the image for the control based on the display width. • Select **No** to deny the system to reduce size of the image for the control based on the display width. The default value is **No**.
Alternate Text	Alternate text for the control.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.

Scripts

For information about using scripts, see scripting guidelines for Image control.

Image Path Tags

Tag	Description
<BPMUITemplates>	Replaces the path with the application path excluding the theme name attached to the repository. For example, **<BPMUITemplates>/Images/imagename.png** is replaced as **/EnterpriseConsole/BPMUITemplates/Default/Images/imagename.png**
<ThemePath>	Replaces the path with the application path including the theme name attached to the repository. For example, **<ThemePath>/Images/imagename.png** for the Lite theme is replaced as **/EnterpriseConsole/BPMUITemplates/Default/Themes/Lite/Images/imagename.png** When there is no theme attached to the repository, the theme name folder along with the parent themes folder is removed from the path. For example, **/EnterpriseConsole/BPMUITemplates/Default/**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Tag	Description
	Images/imagename.png
<ThemeName>	Replaces the path with the theme name attached to the repository. For example, **<BPMUITemplates>/Themes/<ThemeName>/Images/imagename.png** for the Lite theme is replaced as **/EnterpriseConsole/BPMUITemplates/Default/Themes/Lite/Images/imagename.png** When there is no theme attached to the repository, the **<ThemeName>** tag along with the parent themes folder is removed from the path. When there is no parent themes folder, the theme name folder is removed from the path. When there is no theme name folder, the **<ThemeName>** tag is removed from the path.

Invoke Workflow

Use **Invoke Workflow** control to invoke a workflow from your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Button Text	Text for the Invoke Workflow button.
Workflow Details	Details of the workflow that is to be invoked. **To add Workflow Details for the control** 1. Click the **Workflow Details** icon. The **Workflow Details Configuration** page appears. 2. In the **Workflow Name** box, click a workflow name. 3. In the **Workflow Version** list, click a workflow version. 4. Click **Save & Continue**. The selected workflow details is set for the control and the **Workflow Details Configuration** page is closed. **Note:** - Workflow details are displayed in **<Workflow Name> ~ <Workflow Version>** format. - When the selected version of the workflow is in published state, and all or any one of the following properties are modified: -Workflow Details -XML Variable Name - XML Schema then, the workflow must be unpublished, or a different version of the workflow must be selected, or a new version of the workflow must be created. - For more information, see Overwriting Workflows.
XML Variable Name	XML variable name to store the form XML Schema in workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	To add an XML Variable Name for the control 1. Click the **XML Variable Name** icon. The **XML Variable Configuration** page appears. 2. Select the existing XML variable name or create a new XML variable. • To select the existing variable name, click **Select XML Variable** drop-down list and select the name. • To create a new variable, a. Select **Create New...** option from the **Select XML Variable** drop-down list. b. Enter the name for the XML Variable in the **Specify Name** text box. The specified name gets saved to the XML Variable list. 3. Click **Save & Continue**. The selected XML variable name is set for the control and the **XML Variable Configuration** page is closed. **Note**: The **Selected XML Variable** drop-down lists all the existing XML variables. You can either select the existing variable or create a new variable. For more information, see About XML Variables.
XML Schema	Form XML schema to create the XML variable for workflow. **To add an XML Schema for the control** 1. Click the **XML Schema** icon. The **XML Schema Configuration** page appears. 2. Click **Form** for a form or click **Grid** for a grid in the **Schema**. 3. If you select **Grid** in the schema, then select a grid in the **Grid** box. The Grid box lists all the grid controls in the form including the ones present in the Embed form controls. 4. Click **Save & Continue**. The selected XML schema is set for the control and the **XML Schema Configuration** page is closed. **Note:** - When the XML schema set for the Form or Grid does not match with the XML schema set for the XML

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Variable of the selected workflow, the XML Variable must be overwritten. - For more information, see Overwriting XML Variables.
Mapping Schema	Mapping Schema to create mapping definition for workflow. You can map the schema only when the XML schema set for the Form or Grid does not match with the XML schema set for the XML Variable of the selected workflow. **To map the schema for the control** 1. Configure the **XML Schema**. Refer steps 1 to 4 of **XML Schema** in the above row. 2. Click the **Mapping Schema** icon to map the Workflow XML schema with the control XML schema. **Note**: Clear the existing mapping definition (if exists), in case: - a new XML Variable is selected - a Form XML schema is changed to a Grid XML schema or vice-versa Click the **Remove** icon to clear the existing mapping definition. The **XML Variable Overwrite Configuration** page appears. 3. Select the **Create a mapping schema** option and then click **Save & Continue**. The I**nvoke Workflow XML Mapping** page appears. 4. Select the elements or attributes from the **Workflow XML** column to be mapped with the elements or attributes of the **Invoke Workflow XML**. 5. Click **Save**. The selected XML schema is mapped with the control and the **Invoke Workflow XML Mapping** page is closed. **Note:** When the XML schema set for the Form or Grid does not match with the XML schema set for the XML Variable of the selected workflow, the XML Variable must be overwritten or mapped. For more information, see Overwriting XML Variables.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Alignment	Alignment of the button with respect to the form. • Select **Default** to allow the system to align the button. The system aligns the button to the left of the form or as specified by the control. • Select **Left** to align the button to the left of the form. • Select **Right** to align the button to the right of the form. • Select **Center** to align the button to the center of the form.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Display Mode	Show text and/or image for the button. • Select **Text Only** to show only text for the button. • Select **Image Only** to show only image for the button. • Select **Text and Image** to show both text and image for the button. The default value is **Text Only**.
Button Size	Size of the button for the control. • Select **Standard** to set the size of the control to 120 x 32 pixels. • Select **Small** to set the size of the control to 30 x 30 pixels. • Select **Medium** to set the size of the control to 180 x 86 pixels. • Select **Large** to set the size of the control to 280 x 130 pixels. The default value is **Standard**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Image Path	Path to the image for the control. The image path is specified as: • A URL starting with **http://** or **https://**. • An image path tag. For information about using image path tags, see *Image Path Tags* in Image. • A path to the **BPMUITemplates** folder as **BPMUITemplates\Default\Images**. **Note:** - You can use relative paths as follows: --- Use **.** for the current folder where the html page resides. --- Use **..** for the folder above the current folder where the html page resides. We recommend the following image sizes when the **Display Mode** property is set to **Image**. • **Small** – 30 x 30 pixels. • **Medium** – 180 x 86 pixels. • **Large** – 280 x 130 pixels. We recommend the following image sizes when the **Display Mode** property is set to **Text and Image**. • **Small** – 30 x 30 pixels. • **Medium** – 40 x 40 pixels. • **Large** – 50 x 50 pixels. **Note:** - This property is applicable only when the **Display Mode** property is set to **Image** or **Text and Image**, and the **Button Size** property is set to **Small**, **Medium** or **Large**. - If the **Display Mode** property is set to **Text and Image** and the **Button Size** property is set to **Small**, then only the image is displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Image Position	Position of the image with respect to the text for the control. • Select **Left** to align the image to the left of the text for the control. • Select **Right** to align the image to the right of the text for the control. • Select **Top** to align the image to the top of the text for the control. • Select **Bottom** to align the image to the bottom of the text for the control. The default value is **Left**. **Note:** This property is applicable only when the Display Mode property is set to Text and Image.
Show as Link	Enable or disable the control to show as link. • Select **Yes** to enable the control to show as link. • Select **No** to disable the control to show as link. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Show Confirmation Message	Show or hide the confirmation message for the control. • Select **Yes** to show the confirmation message for the control. • Select **No** to hide the confirmation message for the control. The default value is **No**.
Confirmation Message Text	Confirmation message for the control. **Note:** This property is applicable only when the **Show Confirmation Message** property is set to **Yes**.
Show Success Message	Show or hide the success message for the control. • Select **Yes** to show the success message for the control. • Select **No** to hide the success message for the control. The default value is **No**.
Success Message Text	Success message for the control. **Note:** This property is applicable only when the **Show Success Message** property is set to **Yes**.

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Execution Mode	Allow or deny a user to act on consecutive tasks immediately without navigating to the **Inbox**. • Select **Asynchronous** to deny a user to act on consecutive tasks immediately without navigating to the **Inbox**. • Select **Synchronous** to allow a user to act on consecutive tasks immediately without navigating to the **Inbox**. The default value is **Asynchronous**. **Note:** - The default timeout for synchronous execution is 20 seconds to ensure optimal performance. The execution status of the workflow gets updated to FE (Finished with Error) in case there is a timeout. - When Invoke workflow control is configured to initiate a Quickflow, the **Execution Mode** is set to **Synchronous** and is read-only. - If you wish to change the timeout duration, override the **defaultredirectiontimeout** entry in the **Properties** column of the row with the name **WorkflowManagementGlobalSettings** in the **SkService** table of the Farm database. For example, to override the timeout duration to 60 seconds, change **[CDATA[0]]** to **[[CDATA[60]]** for **defaultredirectiontimeout** as follows: `<property key="defaultredirectiontimeout"><![CDATA[60]]></property>` Reset IIS and restart Workflow Engine for the changes to reflect. - When the **Execution Mode** is set to **Synchronous**, ensure that none of the activities initiated through the Invoke Workflow control have a sleep behavior. For instance, activities like Human or Scheduler activities that enter sleep mode should not be included.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Close Form	Allow or deny the system to close the form when the workflow is invoked. • Select **Yes** to allow the system to close the form when the workflow is invoked. • Select **No** to deny the system to close the form when the workflow is invoked. The default value is **No**. **Note:** When a form is loaded in a separate window, the system closes the form when the workflow is invoked. When a form is loaded within an existing window (inline), the system reloads the form when the workflow is invoked. This property is not supported in InTouch HMI and InTouch OMI.
Form Button	Enable or disable the control to act as form button. When this property is enabled, the control is displayed at the bottom-right corner of the form. • Select **Yes** to enable the control to act as form button. • Select **No** to disable the control to act as form button. The default value is **No**. **Note:** If the value is **Yes**, and the form is viewed on a mobile device, then the control appears in the **Action Bar** of the mobile device.

Scripts

For information about using scripts, see scripting guidelines for Invoke Workflow control.

Overwrite Workflows

When the selected version of the workflow is in published state, and one or all of the following Invoke Workflow control properties are modified
-Workflow Details
-XML Variable Name
-XML Schema
then the workflow must be unpublished, or a different version of the workflow must be selected, or a new version of the workflow must be created.

Use the **Workflow Overwrite Configuration** page to select an overwrite configuration.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Follow the steps for Editing Properties of Controls till you a get the **Properties** dialog box for the **Invoke Workflow** control, edit properties of the control, and then click **Apply**.

To overwrite a workflow

1. In the **Workflow Overwrite Configuration** page, do one of the following:

Task	Action
• Unpublish the selected version of the workflow.	a. Under **Overwrite Configuration**, click **Unpublish current version**. b. Follow the steps for *Changing the state of running or paused workflow instances*.
• Create a new draft version of the selected workflow.	Under **Overwrite Configuration**, click **Create a new draft version**.
• Select a specific draft version of the selected workflow.	a. Under **Overwrite Configuration**, click **Select a draft version**. b. In the **Workflow Version** list, click a workflow version to select a specific draft version of the selected workflow. **Note:** This option is available only when the selected workflow has multiple draft versions.

2. Click **Apply**. The **Workflow Overwrite Configuration** page is closed.

Change the state of running or paused workflow instances

1. In the **Workflow Overwrite Configuration** page, do one of the following:

Task	Action
• Unpublish the selected workflow by pausing all running instances of the workflow.	• In the **Workflow Instances** box, click **Pause**. All running instances of the selected workflow are paused and the workflow is unpublished.
• Unpublish the selected workflow by terminating all running and paused instances of the workflow.	• In the **Workflow Instances** box, click **Kill**. All running and paused instances of the selected workflow are terminated and the workflow is unpublished.

2. Click **Apply**. The **Workflow Overwrite Configuration** page is closed.

Overwrite XML Variables

When the XML schema set for the **Invoke Workflow** control does not match with the XML schema set for the XML Variable of the selected workflow, the XML Variable must be overwritten or mapped with the selected workflow.

Use the **XML Variable Overwrite Configuration** page to select an overwrite configuration.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Follow the steps for Editing Properties of Controls till you a get the **Properties** dialog box for the **Invoke Workflow** control,
2. Edit the properties of the control, and then click **Apply**.

To map or overwrite an XML Variable

1. In the **XML Variable Overwrite Configuration** page, do one of the following:

Task	Action
• Map the XML Variable schema to the selected workflow.	• Under **Overwrite Configuration**, select the **Create a mapping schema** option. For information on how to map the schema, refer Mapping Schema property in Invoke Workflow.
• Overwrite the XML Variable in the current draft version of the selected workflow.	• Under **Overwrite Configuration**, select the **Use current version** option.
• Save the XML Variable in the new draft version of the selected workflow.	• Under **Overwrite Configuration**, select the **Create a new draft version** option.
• Overwrite the XML Variable in the specific draft version of the selected workflow.	1. Under **Overwrite Configuration**, select the **Select a draft version** option. 2. In the **Workflow Version** list, click a workflow version to select a specific draft version of the selected workflow. **Note:** This option is available only when the selected workflow has multiple draft versions.

2. Click **Apply**. The XML variable is mapped, overwritten, or saved for the selected version of the workflow, the **XML Variable Overwrite Configuration** page is closed, and the **Properties** dialog box for the **Invoke Workflow** control is closed.

Label

Use **Label** control to show a label in your form. You can use labels to add information in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers.For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Display Text	Text for the control.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the **Name** for the control. • Select **Default** to allow the system to align the Name. The system aligns the Name to the left of the control area or as specified by the container control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Left** to align the Name to the left of the control area. • Select **Top** to align the Name to the top of the control area. • Select **Right** to align the Name to the right of the control area. • Select **Bottom** to align the Name to the bottom of the control area. • Select **Suppress** to hide the Name for the control area. The default value is **Suppress**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the **Name** for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the **Name** is aligned to the top of the control area.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Alignment	Alignment of the **Display Text** with respect to the form. • Select **Default** to allow the system to align

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	the **Display Text**. The system aligns the **Display Text** to the left of the form or as specified by the control. • Select **Left** to align the Display Text to the left of the form. • Select **Right** to align the Display Text to the right of the form. • Select **Center** to align the Display Text to the center of the form.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Show Line	Show or hide a line below the text for the control. • Select **Yes** to show a line below the text for the control. • Select **No** to hide a line below the text for the control. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. Important: The value must start with an alphabet and

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	can contain only alphanumeric characters.

Advanced

Property	Description
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Include in Summary	Allow or deny a user to view the value in the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the Summary of a Base Form. • Select **No** to deny a user to view the value in the Summary of a Base Form. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Label control.

List

Use **List** control to select one or many options from a list in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Options	Options for the control. **To add an option to the control** 1. Click **Add Record**. A new record appears. 2. In the **Text** box, enter the options text. 3. In the **Value** box, enter the options value. 4. In the **Fore Color** palette, click a color (or in the **Color Code** box, enter hexadecimal code of the color), and then click **Apply**. The selected color is applied to the foreground of the options text. 5. In the **Back Color** palette, click a color (or in the **Color Code** box, enter hexadecimal code of the color), and then click **Apply**. The selected color is applied to the background of the options text. **To clear fore color for an option of the control** • On the **Fore Color** palette, in the **Color Code** box, enter **#1b1b1b**, and then click **Apply**. The **Fore Color** is reset to the default color.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	To clear back color for an option of the control • On the **Back Color** palette, in the **Color Code** box, enter **#1b1b1b**, and then click **Apply**. The **Back Color** is reset to the default color. **To delete an option from the control** • Click the **Delete** button. The record is deleted. **Note:** - The options **Text** and **Value** parameters are mandatory and must be unique across all options. - The **Fore Color** and **Back Color** properties are currently not supported in devices.
Default Value	The default value for the control. **Note:** - The value must be one of the values set for the **Value** parameter in the **Options** property. - If the **Selection Mode** property is set to **Single**, then the value must be only one of the values set for the **Value** parameter in the **Options** property. - If the **Selection Mode** property is set to **Multiple**, then the value can be one or many values set for the **Value** parameter in the **Options** property. Multiple values must be separated with **;#**

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: If the value is 0, then the width of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Rows	Number of rows for the control. The default value is **4**.
Selection Mode	Selection mode for the control. • Select **Single** to allow a user to select only one option for the control. • Select **Multiple** to allow a user to select one or many options for the control. The default value is **Single**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**. **Note:** Validation message appears only on submission of the form.

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Text Parameter	Name that identifies the options text to use in scripts for the control.
Value Parameter	Name that identifies the options value to use in scripts for the control.
Forms 2011 Compatible	Make the control compatible with Forms 2011, that is the forms used in BPM 2011. • Select **Yes** to allow the system to make the control compatible with Forms 2011. • Select **No** to deny the system to make the control compatible with Forms 2011. The default value is **No**. **Note:** When a control on a form is made Forms 2011 compatible, the XML instance of the form with the control is changed to ensure compatibility with Forms 2011.
Include in Summary	Allow or deny a user to view the value in the **Summary** of a **Base Form**. • Select **Yes** to allow a user to view the value in the Summary of a Base Form. • Select **No** to deny a user to view the value in the Summary of a Base Form. The default value is **No**. **Note:** This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for List control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Number

Use **Number** control to enter and view numerical data in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Default Value	The default value for the control. **Note:** Ensure the value is within the Minimum Value and the Maximum Value.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Read-Only	Allow or deny a user to enter a value for the control. • Select **Yes** to deny a user to enter a value for the control. • Select **No** to allow a user to enter a value for the control. The default value is **No**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Watermark	Hint text that appears when no value is entered for the control.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**.
Auto-update Value	Allow or deny the system to automatically update the value to the nearest minimum or maximum value for the control. • Select **Yes** to allow the system to automatically update the value to the nearest minimum or maximum value for the control. • Select **No** to deny the system to automatically update the value to the nearest minimum or maximum value for the control. The default value is **No**. For example, assume the **Minimum Value** property is set to 5 and the **Maximum Value** property is set to 10. Now if a user enters the value as 2, then the system automatically updates the value to 5. Similarly, if a user enters the value as 12, then the system automatically updates the value to 10. This behavior is applicable only when the value is set via the User Interface (UI), not via script.
Minimum Value	Minimum value for the control. **Note**: The decimal value supported is up to two decimal places.
Maximum Value	Maximum value for the control. **Note**: The decimal value supported is up to two decimal places.

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Number Format	Number format for the value of the control. The default value is taken from the current server culture. For more information, see Number Formats.
Decimal Places	Decimal places for the value of the control. The default value is **0**.
Decimal Separator	Decimal separator for the value of the control. The default value is taken from the current server culture.
Group Sizes	Group sizes for the value of the control. The default value is taken from the current server culture.
Group Separator	Group separator for the value of the control. The default value is taken from the current server culture. **Note:** This property is applicable only when the **Show Group Separator** property is set to **Yes**.
Show Group Separator	Show or hide the group separator for the value of the control. • Select **Yes** to show the group separator for the value of the control. • Select **No** to hide the group separator for the value of the control. The default value is **No**.
Negative Pattern	Negative pattern for the value of the control Enter `(n)`, `-n`, `- n`, `n-` or `n -`. The default value is taken from the current server culture.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Include in Summary	Allow or deny a user to view the value in the **Summary** of a **Base Form**. • Select **Yes** to allow a user to view the value in the **Summary** of a **Base Form**. • Select **No** to deny a user to view the value in the **Summary** of a **Base Form**. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Number property.

Note:

- If culture is not specified in the **web.config** file, then the values for the culture specific properties are as per the current server culture.

ill

Password

Use **Password** control to enter a password in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Read-Only	Allow or deny a user to enter a value for the control. • Select **Yes** to deny a user to enter a value for the control. • Select **No** to allow a user to enter a value for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is **No**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label Control-Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	- This is applicable only when the **Label Position** is set to **Left** or **Right**.
Watermark	Hint text that appears when no value is entered for the control.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**.
Text Pattern	Regular expression to validate the text for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	For example, use the following regular expression to validate a text string as a phone number: `^[2-9]\d{2}-\d{3}-\d{4}$`
Minimum Length	Minimum length of the value for the control. The default value is **0**. **Note:** If the value is 0, then the minimum length for the control is not set.
Maximum Length	Maximum length of the value for the control. The default value is **0**. **Note:** If the value is 0, then the maximum length for the control is not set.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Security Mode	Retain or secure the value for the control. • Select **None** to retain the value for the control in plain text. • Select **Encryption** to secure the value for the control with AES-128. • Select **Form Type** to secure the value of the control with the encryption algorithm used in Forms. • Select **List Type** to secure the value of the control with algorithms similar to the encryption mechanism used to encrypt the login password. The default value is **Encryption**.
Persist Password	Keep or lose the value for the control on submission of the form. • Select **Yes** to keep the value for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **No** to lose the value for the control. The default value is **No**.

Scripts

For information about using scripts, see scripting guidelines for Password control.

Note:
- **Password** control does not populate the password automatically even if the autofill or autocomplete option in the browser is turned on.
- Text pattern, minimum length, maximum length, and custom validations are applicable only for plain text password.
- We recommend using HTTPS with SSL encryption for additional security.

Pop-Up

Use **Pop-Up** control to open a form as a pop-up window in your form. You can reuse your existing form using this control.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. The default value is the value for the **Name** property without the spaces. **Note:** If you edit this value later, then it might affect the mapping between forms, lists, and workflows..
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Button Text	Text for the control.
Persist Form Definition	Allow or deny a user to persist the form definition. The definition of the source form is fetched and used for the pop-up form. The source form is no longer

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	referenced after the properties of the control are saved. Therefore, the source form does not have to be available in the repository after the properties are configured. • Select **Yes** to allow a user to persist the form definition. • Select **No** to deny a user to persist the form definition. The default value is **No**. The pop-up form pointing to a form with Persist Form Definition set to **No** shows the draft version of the form in the Preview only rather than the published version. The pop-up form pointing to a form with Persist Form Definition set to **Yes** is the draft version of the form.
Form Details	Details of the form that is to be linked. **To add Form Details for the control** 1. Click the **Form Details** button. The **Form Details Configuration** page appears. 2. In the **Form Name** box, click a form name. 3. In the **Form Version** box, click a form version. 4. In the **Parameters** list, click and select a control on the form, if applicable. This maps a pop-up form parameter to a parent form control. The Parameters list appears only if the version of the selected form contains **Forms Parameters**. 5. Click **Save & Continue**. The selected form details is set for the control, and the **Form Details Configuration** page is closed. **Note:** - Ensure the selected form has a **Button** control with the **Button Type** set to **Cancel**, otherwise you will not be able to close the form. - If you click **Finish**, then the form will be submitted and closed. However, an empty instance XML will be generated. - For **Parameters**, the **Title** of the control on the form is displayed, if set for the control. Otherwise, the **XML Node** of the control on the form is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	displayed. - For information about using **Forms Parameters**, see Forms Parameters.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Alignment	Alignment of the control with respect to the form. • Select **Default** to allow the system to align the control. The system aligns the control to the left of the form or as specified by the control. • Select **Left** to align the control to the left of the form. • Select **Right** to align the control to the right of the form. • Select **Center** to align the control to the center of the form.
Dimension	Dimension in percentage for the pop-up window of the control. The default value is **0**. **Note:** If the value is 0, then the dimension for the pop-up window of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Display Mode	Show text and/or image for the control. • Select **Text Only** to show only text for the control. • Select **Image Only** to show only image for the control. • Select **Text and Image** to show both text and image for the control. The default value is **Text Only**.
Button Size	Size of the button for the control. • Select **Standard** to set the size of the control to 120 x 32 pixels. • Select **Small** to set the size of the control to 30 x 30 pixels. • Select **Medium** to set the size of the control to 180 x 86 pixels. • Select **Large** to set the size of the control to 280 x 130 pixels. The default value is **Standard**.
Image Path	Path to the image for the control. The image path is specified as: • A URL starting with **http://** or **https://**. • An image path tag. For information about using image path tags, see *Image Path Tags* in Image. • A path to the **BPMUITemplates** folder as **BPMUITemplates\Default\Images**. **Note:** You can use relative paths as follows: - Use **.** for the current folder where the html page resides. - Use **..** for the folder above the current folder where the html page resides.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	We recommend the following image sizes when the **Display Mode** property is set to Image. • **Small** – 30 x 30 pixels. • **Medium** – 180 x 86 pixels. • **Large** – 280 x 130 pixels. We recommend the following image sizes when the **Display Mode** property is set to Text and Image. • **Small** – 30 x 30 pixels. • **Medium** – 40 x 40 pixels. • **Large** – 50 x 50 pixels. **Note:** - This property is applicable only when the **Display Mode** property is set to **Image** or **Text and Image**, and the **Button Size** property is set to **Small**, **Medium** or **Large**. - If the **Display Mode** property is set to **Text and Image** and the **Button Size** property is set to **Small**, then only the image is displayed.
Image Position	Position of the image with respect to the text for the control. • Select **Left** to align the image to the left of the text for the control. • Select **Right** to align the image to the right of the text for the control. • Select **Top** to align the image to the top of the text for the control. • Select **Bottom** to align the image to the bottom of the text for the control. The default value is **Left**. **Note:** This property is applicable only when the **Display Mode** property is set to **Text and Image**.
Show as Link	Enable or disable the control to show as link. • Select **Yes** to enable the control to show as link. • Select **No** to disable the control to show as link. The default value is **No**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Open New Window	Allow or deny a user to open form in a new browser window. • Select **Yes** to allow a user to open form in a new browser window. • Select **No** to deny a user to open form in a new browser window. However, the system allows the user to open form in the same browser window. The default value is **No**. **Note:** Opening the form in a new browser window improves form performance, but the values for the controls on submission of the form will not be persisted.
Persist Value	Keep or lose the value for the control on submission of the form opened as a pop-up window. • Select **Yes** to keep the value for the control on submission of the form opened as a pop-up window. • Select **No** to lose the value for the control on submission of the form opened as a pop-up window. The default value is **Yes**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Form Button	Enable or disable the control to act as form button. When this property is enabled, the control is displayed at the bottom-right corner of the form. • Select **Yes** to enable the control to act as form button. • Select **No** to disable the control to act as form button. The default value is **No**. **Note:** If the value is **Yes**, and the form is viewed on a mobile device, then the control appears in the **Action Bar** of the mobile device.

Scripts

For information about using scripts, see scripting guidelines for Pop-Up control.

Note:
- **Pop-Up** control does not support forms with **Pop-Up** controls.
- **Pop-Up** control does not support **Chart**, **Slider**, **Card**, and **Hierarchical List** controls when the **Persist Form Definition** property is set to **Yes**.

Progress Bar

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	(Required) XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use forward slash (**/**) for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent, which is not a

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it may affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Enable Column Bindings	Enable this option to set up the lookup source. • Select **Yes** to enable the use of the lookup source. • Select **No** if column binding is not necessary. If enabled, specify the column names based on the lookup source definition for **Unit of Measurement**, **Target Value**, and **Default Value** properties**.** The default value is **No**.
Lookup Source	(Required) Name of the item from the lookup list. **Note**: This is only visible when the **Enable Column Bindings** property is set to **Yes**. To add a name of the item for the control from the lookup list: 1. Click the **Lookup Source** button. The Lookup Source Configuration page appears. 2. In the **Data Source Type** list, select a type of data source. **Note**: Supported datasource types are database and Web API lookup source. 3. In the **Data Source** box, select a data source. 4. In the **Parameters** section, select the parameters to include, if applicable. 5. Click **Save & Continue**. The selected name is set for the control and the Lookup Source Configuration page is closed.
Unit of Measurement	Unit of measurement for the value of the control. This supports a single line with 24 characters and a tooltip is shown when you hover over this property. An ellipsis is added if it exceeds the 24-character limit.
Target Value	Target value of the control. This is the value that the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	target indicator points at. However, if this property is left blank, the target indicator is not shown at preview or runtime. The default value is **0**.
Default Value	This is the value of the control. This supports 10 characters and a tooltip is shown when you hover over this property. An ellipsis is added if the value exceeds the 10-character limit.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note**: If the value is 0, then the width of the control is set as per the default value of 360.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Title	Name or information text for the control. It is shown directly above the top-left corner of the control. This supports two lines with 35 characters for each line and a tooltip is shown when you hover over this property. An ellipsis is added if it exceeds the 35-character limit. If you want to add a line break, use the html tag . **Note:** If this property is not configured or blank, nothing is shown, the other control contents use its space, and the control size is adjusted based on the available content within the control. The described behavior also applies to the **Unit of Measurement** property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Decimal Places	Decimal places for the default value of the control. It also applies to the values in the **Range** property. This displays a maximum number of 3 decimal places. The default value is **0**. **Note**: At runtime, if this property is empty or has a non-numeric value, it resets back to 0. If its value is greater then 3, it stays at its maximum value of 3.
Scale Minimum Value	(Required) Minimum value displayed on the scale. The default value is **0**. **Note**: At runtime, if this property is empty or has a non-numeric value, its value is set to 0.
Scale Maximum Value	(Required) Maximum value displayed on the scale. The default value is **100**. **Note**: At runtime, if this property is empty or has a non-numeric value, its value is set to 100.
Scale	Show or hide the scale, or use the **Range** property values in the control. • Select **Hide** to hide the scale in the control. • Select **Show** to show the scale in the control. • Select **Ranges** to use the breakpoint values specified under the **Range** property instead of the scale values. **Note**: If **Ranges** is selected, the scale value shows the decimal places based on what is specified in the **Decimal Places** property. If this is set to **Show**, the **Number of Division** property is visible. The default is **Hide**.
Number of Division	Number of division within the scale, from the minimum value to the maximum value. This number divides the difference between the **Scale Maximum Value** and **Scale Minimum Value**. For example: • **Scale Minimum Value** = 0 • **Scale Maximum Value** = 200 • **Number of Division** = 5

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	So the difference between the scale max and min values is 200. The difference divided by the number of division is 40. The scale tick marks are then shown as 0, 40, 80, 120, 160, and 200. **Note:** This is only visible when the **Scale** property is set to **Show**. The default value is **10**.
Control Background Color	The color to represent the background of the control. The default is white.
Background Color	The color to represent the background of the bar in the control. The default is gray.
Bar Color	The color to represent the bar in the control. The default is green.
Range	The breakpoint to apply in the control range bar. Each added row corresponds to a breakpoint and for each breakpoint value, you can assign a color. You can have a maximum of 10 breakpoints. **Note**: If the configured range does not cover the full scale, the undefined range takes the color that is set for the **Bar Color** property. The value for the breakpoint must be between the specified value for the **Scale Minimum Value** and **Scale Maximum Value** properties.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
On Initialize	Run a script after the control has been initialized, but before it is rendered to the UI. For more information, see *Scripting Guidelines.*
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Enable Refresh	Enable or disable control refresh. • Select **Yes** to enable control refresh and configure the **Refresh Rate** property. • Select **No** to disable control refresh and hide the **Refresh Rate** property. The default value is **No**.
Refresh Rate	This is how often the control display is refreshed, in seconds, to fetch value from the data source. **Note**: This is only visible if the **Enable Refresh** property is set to **Yes**. The default is 30 seconds and the valid refresh rate is from 5 to 600 seconds.

Scripts

For information about using scripts, see scripting guidelines for Progress Bar control.

Example Scripting Scenarios and Rendering

The following sections provide example scenarios where scripting is use with the Progress Bar control. It also has example configuration with its corresponding Progress Bar control rendering.

Example Progress Bar Rendering

The following rendering examples for Progress Bar control are based on the configuration in the control's properties page. The configured properties are listed in the table and most of the properties that are kept to their default values are no longer mentioned.

Example configuration and rendering

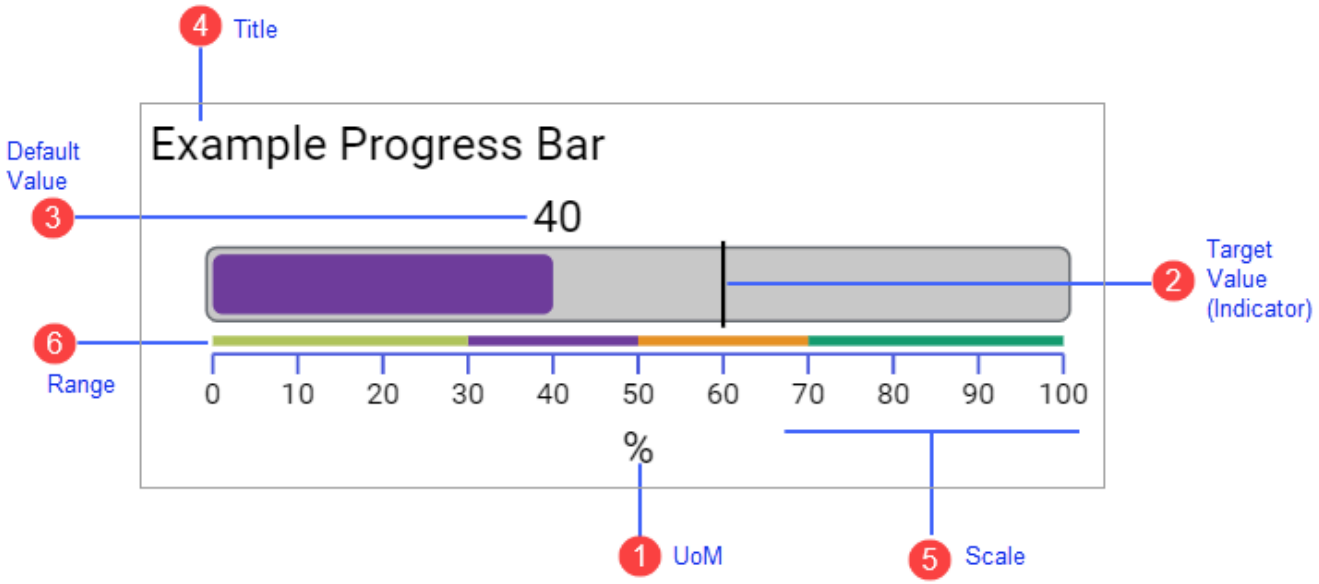
Basic	Appearance
1. **Unit of Measurement**: `%` 2. **Target Value**: `60`	4. **Title**: `Example Progress Bar` 5**. Scale**: `Show` (selected)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Basic	Appearance
3. **Default Value**: `40`	• **Number of Division**: `10` (default) 6. **Range** (*with three breakpoints*) • Value 1 - 30 (lime) • Value 2 - 50 (purple) • Value 3 - 70 (orange)

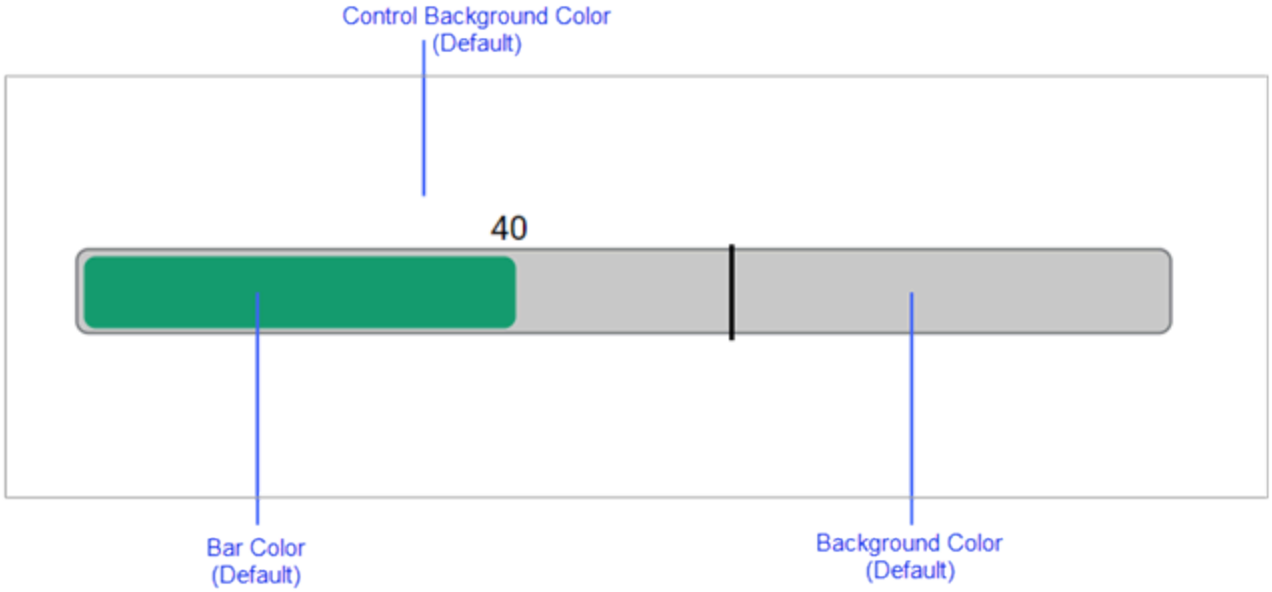
In the following rendering example, the configured range does not cover the full scale, so the undefined range takes the color that is set for the **Bar Color** properties, which in the current configuration is set to the default green color.

For the rendered bar color to show the **Default Value**, notice that it takes the assigned color of the second breakpoint (50, purple) because the **Default Value** falls on that breakpoint range.



The next example rendering has a configuration where most of the properties are kept to their default values. The properties that are empty by default are also left the same way, except for the **Default Value** and **Target Value** properties, so 40 is shown as well as the target indicator. However, the **Target Value** of 60 is not shown since the **Scale** property is set to **Hide (**by default), so the scale is hidden.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



QR Code

Use **QR Code** control to scan a QR code. This control is applicable only for **AVEVA Work Tasks Pro** App.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Button Text	Text for the control.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Alignment	Alignment of the control with respect to the form. • Select **Default** to allow the system to align the control. The system aligns the control to the left of the form or as specified by the control. • Select **Left** to align the control to the left of the form. • Select **Right** to align the control to the right of the form. • Select **Center** to align the control to the center of the form.
Display Mode	Show text and/or image for the control. • Select **Text** Only to show only text for the control. • Select **Image** Only to show only image for the control. • Select **Text and Image** to show both text and image for the control. The default value is **Text Only**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Button Size	Size of the button for the control. • Select **Standard** to set the size of the control to 120 x 32 pixels. • Select **Small** to set the size of the control to 30 x 30 pixels. • Select **Medium** to set the size of the control to 180 x 86 pixels. • Select **Large** to set the size of the control to 280 x 130 pixels. The default value is **Standard**.
Image Path	Path to the image for the control. The image path is specified as: • A URL starting with **http://** or **https://**. • An image path tag. For information about using image path tags, see *Image Path Tags* in Image. • A path to the **BPMUITemplates** folder as **BPMUITemplates\Default\Images**. **Note:** You can use relative paths as follows: - Use **.** for the current folder where the html page resides. - Use **.** for the folder above the current folder where the html page resides. We recommend the following image sizes when the **Display Mode** property is set to Image. • **Small** – 30 x 30 pixels. • **Medium** – 180 x 86 pixels. • **Large** – 280 x 130 pixels. We recommend the following image sizes when the **Display Mode** property is set to Text and Image. • **Small** – 30 x 30 pixels. • **Medium** – 40 x 40 pixels. • **Large** – 50 x 50 pixels. **Note:** - This property is applicable only when the **Display**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Mode property is set to **Image** or **Text and Image**, and the **Button Size** property is set to **Small**, **Medium** or **Large**. - If the **Display Mode** property is set to **Text and Image** and the **Button Size** property is set to **Small**, then only the image is displayed.
Image Position	Position of the image with respect to the text for the control. • Select **Left** to align the image to the left of the text for the control. • Select **Right** to align the image to the right of the text for the control. • Select **Top** to align the image to the top of the text for the control. • Select **Bottom** to align the image to the bottom of the text for the control. The default value is **Left**. **Note:** This property is applicable only when the **Display Mode** property is set to **Text and Image**.
Show as Link	Enable or disable the control to show as link. • Select **Yes** to enable the control to show as link. • Select **No** to disable the control to show as link. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
On Data Change	Run a script when the value of the control changes.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Form Button	Enable or disable the control to act as form button. When this property is enabled, the control is displayed at the bottom-right corner of the form. • Select **Yes** to enable the control to act as form button. • Select **No** to disable the control to act as form button. The default value is **No**. **Note:** If the value is **Yes**, and the form is viewed on a mobile device, then the control appears in the **Action Bar** of the mobile device.

Scripts

For information about using scripts, see scripting guidelines for Button property.

Note:

AVEVA Work Tasks utilizes a third-party library to enable barcode scan and support scanning. Note that all variations of a code, such as GS1 based on Code 128, may not be supported.

The supported barcode types for iOS, Android, and UWP are listed below:

Supported Barcode Types for iOS	Supported Barcode Types for Android and UWP
EAN-8, EAN-13	EAN-8, EAN-13, UPC-A, UPC-E
Code 39, Code 93, Code 128	Code 39, Code 93, Code 128
ITF	ITF
AZTEC	AZTEC
QR Code	QR Code
Data Matrix	Data Matrix
PDF 417	PDF 417
Codabar	Codabar
	RSS-14 (all variants), RSS Expanded (most variants)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

MaxiCode

Radio Button

Use **Radio Button** control to select one of a group of mutually exclusive options in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Options	Options for the control. **To add an option to the control** 1. Click **Add Record**. A new record appears. 2. In the **Text box**, enter the options text.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	3. In the **Value box**, enter the options value. 4. In the **Fore Color** palette, click a color (or in the **Color Code** box, enter hexadecimal code of the color), and then click **Apply**. The selected color is applied to the foreground of the options text. 5. In the **Back Color** palette, click a color (or in the **Color Code** box, enter hexadecimal code of the color), and then click **Apply**. The selected color is applied to the background of the options text. **To clear fore color for an option of the control** • On the **Fore Color** palette, in the **Color Code** box, enter **#1b1b1b**, and then click **Apply**. The **Fore Color** is reset to the default color. **To clear back color for an option of the control** • On the **Back Color** palette, in the **Color Code** box, enter #1b1b1b, and then click Apply. The **Back Color** is reset to the default color. **To delete an option from the control** • Click the **Delete** button. The record is deleted. **Note:** The options **Text** and **Value** parameters are mandatory and must be unique across all options.
Default Value	The default value for the control. **Note:** The value must be one of the values set for the **Value** parameter in the **Options** property.

Appearance

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a Base Form with **Label Position** property set to Top, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Options Per Row	Number of options per row for the control. Enter a value from 0 to 10.
Options Direction	Direction of the options for the control. • Select **Horizontal** to align the options next to each other for the control. • Select **Vertical** to align the options one below the other for the control. The default value is **Horizontal**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**. **Note:** Validation message appears only on submission of the form.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Text Parameter	Name that identifies the options text to use in scripts for the control.
Value Parameter	Name that identifies the options value to use in scripts for the control.
Include in Summary	Allow or deny a user to view the value in the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the Summary of a Base Form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **No** to deny a user to view the value in the Summary of a Base Form. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Radio Button control.

Report Viewer

Use **Report Viewer** control to view reports in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
Description	Description for the control.
Report Name	Name that identifies the report. **Note:** This property is applicable only if the **Report View Option** in the **Report Configuration** is set to Popup.
Report Configuration	Configure the report view. **To configure the report view** 1. Click the **Report Configuration** button. The **Report Configuration** page appears. 2. Configure the report view, and then click **Submit**. The configuration is saved and the **Report Configuration** page is closed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	For more information, see Report Viewer Configuration.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Top**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.

Scripts

For information about using scripts, see scripting guidelines for Report Viewer control.

Note:
- The Report Viewer control is currently not supported in mobile devices.
- When a parameter of a Report in SSRS is defined with Specified Default Values and if any of the default values specified does not exist in the Available Values of the parameter, then none of the valid default values is selected. For example, if the Available Values for the Report Parameter is specified as 1, 2, 3, 4 and the Default Values for the Report Parameter is specified as 3, 4, 5, 6; the Report displayed in the **Report Viewer** control does not consider the valid Default Values for the parameter.
- When a Boolean Parameter has Available and Default Specified values, you can define only one default value and two Available values. The value should be either true or false.
- DateTime Multiple Values Parameter type is not supported.
- In the SSRS Report Builder, set the Report with the **Language** property to **=User!Language**. This will display the Report in the **Report Viewer** control as per the locale or language of the web application.

Report Viewer Configuration

The **Report Viewer Configuration** control allows you to configure the report view.

In the **Report Viewer properties** window, click the **Report Configuration** option. The Report Configuration page appears and configure the following:

- **Report View Option**: The following options are available to view the report in forms:
  - Embed
  - Popup
  - Export

  Embed: The report view can be embedded in the form using the following settings:
- **Display Options**: Displays the following options:
  - Parameter Prompts
  - ToolBar
  - Back Button
  - Page Navigation Controls
  - Refresh Button
  - Find Controls
  - Export Controls
  - Print Button

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Report Template**: Select the required report template from the dropdown list to generate the report view.
- **Parameters**: Allows you to set the values for the selected report template parameters. On selecting the required Report Template, the parameters defined for this particular template will be listed. All the parameters with the * mark are mandatory.

  By default, only the unhidden parameters are displayed. You can check the **Include Hidden Parameters** option to view all the hidden parameters.

  You can specify a value for these parameters in two ways:

  - **Controls**: Select **Controls** from the parameter dropdown and click the button to select a form control. At runtime, the value of the control will be reflected on the parameter, based on which the reports are fetched from the Reports list.

    Note: It is recommended to have controls, which are used as parameters, in the same level as that of the report viewer control.

  - **Values**: Users can select the Values option in the parameter drop-down and specify the value.

    Among the parameters displayed here, the Time Zone ID and the TagName parameters have specific icons, Time Zone and Historian Tag, displayed against them respectively. Users should note that it is a prerequisite to have a Historian Connection established before selecting a Time Zone or a Historian Tag. For more information on how to establish a Historian Connection, refer to Creating a Historian Connection under the Report Activities topic in *Developer Guide*.

Time Zone: If a report parameter name is the Time Zone ID, then the Time Zone icon will be displayed for that particular parameter.

On clicking of this icon, the Time Zone screen pops up. Here, the user can select the required Historian Connection. On selection of the connection, different time zones will be populated in the Time Zone drop-down list. The user can select an appropriate time zone and click Select.

Historian Tag: If a report parameter name is TagName, then the Historian Tag icon is displayed for that particular parameter.

On clicking of this icon, the Historian Tags screen pops up. Here, the user can select the required Historian Connection. On selection of the connection, all the Historian tags will be displayed in the left-hand side list box of Select Tag(s) section.

The user can select the required tags, click the > button to move the tags to the right-hand side of the Select Tag(s) section and click Select. If the user wants remove any tag from the right-hand side list box, then the user has to select the tag(s) and click the < button. The tag will be placed in the left-hand side list box.

A parameter can also be set to a null value, based on the parameter definition in the Report Template. In such cases the parameter would be displayed as shown in the image given below:

This is available only when the Values option is selected in the Parameters drop-down list. As seen in the above image, selecting the null check box would disable the value-specific controls of that parameter.

Note: Embed option is recommended for small reports as large reports would degrade the performance of the form.

Pop-up: This option works very similar to the Embed option. Users can select the Popup radio button to configure a report to be viewed in a pop-up screen. For more details, refer to Embed section.

Export: Select this option to export <>:

Report Source: Specify the source of the report. If the Report radio button is selected, all reports from the Reports list will be listed in the Report drop-down.

If the Report Generator Activity radio button is selected, then the user has to set the Report Reference Variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The user can specify a value for these parameters in two ways:

Users can select Controls option in the parameter drop-down and click the button to select a Form control. At runtime, the value of the control would reflect on the parameter, based on which the report view is generated.

Users can select the Values option in the parameter drop-down and specify the value. The value should be the Title or GUID of the report.

Report: Select the report from the drop-down list which has all the available Reports.

Appearance

Field	Description
Is control enabled?	Select **Yes**, if the control has to be enabled, else select **No**.
Label Position	Select the positioning of the label from the drop-down list. The available options are Default, Left, Top, Right, Bottom, and Suppress.
Is control read only?	Select **Yes**, if the control has to be visible, else select **No**.
Width	Enter an integer value in this field, to set the width of the field in pixels. **Note**: A value '0' will set the width of the field in default percentage value.
Height	Enter an integer value in this field, to set the height of the field in pixels. **Note**: A value '0' will set the height of the field in default percentage value.
Mobile Display Mode	Select the required mode from the drop-down list to display this control in the mobile devices. The available options are: Default, Show only in Web, and Show only in Mobile

Validation

Field	Description
Watermark	Enter a suitable value in this field, which is the default value that appears, when the controls value is empty.

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Id	The reference identification code in this field is generated automatically.
On Data Changed Script	Click the icon and enter the script to run when a data change event occurs. The event occurs when the value of this control changes.

Note: Report View Control does not support scripts.

Scripts

For more information, refer to *Using Scripts in the Forms*.

Viewing a Report as an embedded report in a form using Report Viewer Control

Given below are the steps to view a Report as an embedded report:

1. Create a new form using the **Tools** > **Manage Forms** > **New**.
2. Drag and drop a **Date Time Input** control.
3. Enter the **Tag name** and click in the **Mapped XML Node** field to populate the value.
4. Click **Save**.
5. Drag and drop the **Report Viewer** control to the form designer.
6. Enter the **Tag name** and click in the **Mapped XML Node** field to populate the value.
7. Click the **Report Configuration** icon to open the window.
8. Specify all the required properties in this window. Refer to Embed section for more details.
9. Click **Submit** in the Report Configuration window.
10. Click **Save** in the Report Viewer properties window.
11. Click the **Preview** button in the ribbon bar.
12. In the preview window, enter a value in the **Date Time Input** field. The report will be generated on the screen.

    To view the report in a full screen mode, click the **View Report** in full screen link.

Viewing a Report in a Pop-up window using Report Viewer Control

Given below are the steps to view a Report in a pop-up window:

1. Create a new form using the **Tools** > **Manage Forms** > **New**.
2. Drag and drop a **Date Time Input** control.
3. Enter the **Tag name** and click in the **Mapped XML Node** field to populate the value.
4. Click **Save**.
5. Drag and drop the **Report Viewer** control to the form designer.
6. Enter the **Tag name** and click in the **Mapped XML Node** field to populate the value.
7. Click the **Report Configuration** icon to open the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

8. Specify all the required properties in this window. Refer to Pop-up section for more details.
9. Click **Submit** in the Report Configuration window.
10. Click **Save** in the **Report Viewer** properties window.
11. Click the **Preview** button in the ribbon bar.

    In the preview window, enter a value in the **Date Time Input** field and click the **View Report** link.

    The report appears in a pop-up window as shown in the image given below:

Viewing a Report through a Download link in a form using Report Viewer Control

Given below are the steps to view a Report through a download link:

1. Create a new form using the **Tools** > **Manage Forms** > **New**.
2. Drag and drop the **Report Viewer** control to the form designer.
3. Enter the Tag name and click in the **Mapped XML Node** field to populate the value.
4. Click the **Report Configuration** icon to open the window.
5. Specify all the required properties in this window. Refer to Download section for more details.
6. Click **Submit** in the Report Configuration window.
7. Click **Save** in the Report Viewer properties window.
8. Click the **Preview** button in the ribbon bar.

    In the preview window, click the **Download Report** link.

    By clicking the **Download** button, you can either save the report to the required location on your computer or open it to view its contents.

Note: The Report Viewer control settings within an embedded form control will be overridden with the settings of the Report Viewer Control present in the same form. This occurs if the Report Viewer Control and the embedded form control having a Report Viewer Control are present in the same form and both the Report Viewer Control in the Form and the Report Viewer controls within the embedded form control have the same Id.

To resolve this, export the main form and change the Report Viewer control Id and ensure it is different from the Id of Report viewer control present in the embedded form control. Import the edited XML back to the forms designer.
Report Viewer Control does not support scripts.

Embed Reports

Embed a report view to show the report view in your form.

- Follow the steps for Editing Properties of Controls till you a get the **Properties** dialog box for the Report Viewer control.

To embed a report

1. Click the **Report Configuration** button. The **Report Configuration** page appears.
2. Under **Report View Option**, click **Embed**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Under **Display Options**, click the required options to show the selected UI elements in the report view.
4. In the **Report Template** list, click a report template based on which the report view is generated.
5. Under **Parameters**, set the values for the required report template parameters.
6. Click **Include Hidden Parameters** to show the hidden report template parameters, and then set the values, if required.
7. Click **Submit**. The configuration is saved and the **Report Configuration** page is closed.

Note:
- If you select the **Parameter Prompts** option (under **Display Options** in **Report Configuration**), then ensure the **Width** and **Height** properties set for the control is at least 250 pixels to display the report view properly.
- Similarly, if the **Parameter Prompts** option is not selected, then ensure the **Width** and **Height** properties set for the control is at least 200 pixels to display the report view properly.

Link Pop-up Reports

Show as a link to a report view in your form to open the report view as a pop-up window.

- Follow the steps for Editing Properties of Controls till you a get the Properties dialog box for the Report Viewer control.

To link a report

1. Click the **Report Configuration** button. The **Report Configuration** page appears.
2. Under **Report View Option**, click **Popup**.
3. Under **Display Options**, click the required options to show the selected UI elements in the report view.
4. In the **Report Template** list, click a report template based on which the report view is generated.
5. Under **Parameters**, set the values for the required report template parameters.
6. Click **Include Hidden Parameters** to show the hidden report template parameters, and then set the values, if required.
7. Click **Submit**. The configuration is saved and the **Report Configuration** page is closed.

Note:
- If you select the **Parameter Prompts** option (under **Display Options** in **Report Configuration**), then ensure the **Width** and **Height** properties set for the control is at least 250 pixels to display the report view properly.
- Similarly, if the **Parameter Prompts** option is not selected, then ensure the **Width** and **Height** properties set for the control is at least 200 pixels to display the report view properly.

Link Downloadable Reports

Show as a link to a report view in your form to download the report view.

- Follow the steps for Editing Properties of Controls till you a get the **Properties** dialog box for the **Report Viewer** control.

To download a report

1. Click the **Report Configuration** button. The **Report Configuration** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Under **Report View Option**, click **Download**.
3. Under **Report Source**, do one of the following:

Task	Action
• Show all reports to select in the **Reports** list.	• Click **Report**.
• Set the Report Reference Variable.	a. Click **Report Generator Activity**. b. In the **Report Reference Variable** list, click **Controls**, and then click the **More** button to click a form control. OR In the **Report Reference Variable** list, click **Values**, and then enter a **GUID** or **Title** of the report.

4. In the **Report** list, click a report.
5. Click **Submit**. The configuration is saved and the **Report Configuration** page is closed.

RFID

Use **RFID** control to scan a RFID tag using Near Field Communication. This control is applicable only for AVEVA Work Tasks Pro App.

Note:
- RFID control supports only RFID of plain text type.
- RFID control is not supported in AVEVA Work Tasks Pro app for UWP.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	**Description**
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Button Text	Text for the control.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Alignment	Alignment of the control with respect to the form. • Select **Default** to allow the system to align the control. The system aligns the control to the left of the form or as specified by the control. • Select **Left** to align the control to the left of the form. • Select **Right** to align the control to the right of the form. • Select **Center** to align the control to the center of the form.
Display Mode	Show text and/or image for the control. • Select **Text** Only to show only text for the control. • Select **Image** Only to show only image for the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	control. • Select **Text and Image** to show both text and image for the control. The default value is **Text Only**.
Button Size	Size of the button for the control. • Select **Standard** to set the size of the control to 120 x 32 pixels. • Select **Small** to set the size of the control to 30 x 30 pixels. • Select **Medium** to set the size of the control to 180 x 86 pixels. • Select **Large** to set the size of the control to 280 x 130 pixels. The default value is **Standard**.
Image Path	Path to the image for the control. The image path is specified as: • A URL starting with **http://** or **https://**. • An image path tag. For information about using image path tags, see *Image Path Tags* in Image. • A path to the **BPMUITemplates** folder as **BPMUITemplates\Default\Images**. **Note:** You can use relative paths as follows: - Use **.** for the current folder where the html page resides. - Use **..** for the folder above the current folder where the html page resides. We recommend the following image sizes when the **Display Mode** property is set to Image. • **Small** – 30 x 30 pixels. • **Medium** – 180 x 86 pixels. • **Large** – 280 x 130 pixels. We recommend the following image sizes when the **Display Mode** property is set to Text and Image.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• **Small** – 30 x 30 pixels. • **Medium** – 40 x 40 pixels. • **Large** – 50 x 50 pixels. **Note:** - This property is applicable only when the **Display Mode** property is set to **Image** or **Text and Image**, and the **Button Size** property is set to **Small**, **Medium** or **Large**. - If the **Display Mode** property is set to **Text and Image** and the **Button Size** property is set to **Small**, then only the image is displayed.
Image Position	Position of the image with respect to the text for the control. • Select **Left** to align the image to the left of the text for the control. • Select **Right** to align the image to the right of the text for the control. • Select **Top** to align the image to the top of the text for the control. • Select **Bottom** to align the image to the bottom of the text for the control. The default value is **Left**. **Note:** This property is applicable only when the **Display Mode** property is set to **Text and Image**.
Show as Link	Enable or disable the control to show as link. • Select **Yes** to enable the control to show as link. • Select **No** to disable the control to show as link. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
On Data Change	Run a script when the value of the control changes.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Form Button	Enable or disable the control to act as form button. When this property is enabled, the control is displayed at the bottom-right corner of the form. • Select **Yes** to enable the control to act as form button. • Select **No** to disable the control to act as form button. The default value is **No**. **Note:** If the value is **Yes**, and the form is viewed on a mobile device, then the control appears in the **Action Bar** of the mobile device.

To use the value coming from RFID control, use the **On Data Change** script of RFID Control. For example,

```
control.findByXmlNode("Textinput").value = currentValue;
```

"Textinput" is the value coming from RFID Control.

Scripts

For information about using scripts, see scripting guidelines for Button property.

Rich Text

Use **Rich Text** control to enter and view rich text data in your form. You can style the text, align the paragraphs, increase or decrease indentation, insert hyperlinks, images, and tables. To access the inserted hyperlink, press the Control key, and then click on the hyperlink text.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use / for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the Name property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.

Appearance

Property	Description
Show Toolbar	Show or hide the toolbar. • Select **Yes** to show the toolbar. • Select **No** to hide the toolbar. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Read - Only	Allow or deny a user to enter a value for the control. • Select **Yes** to deny a user to enter a value for the control. • Select **No** to allow a user to enter a value for the control. • The default value is **No**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: If the value is **Yes**, then the toolbar will not be visible, irrespective of the value of the **Show Toolbar** property.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** -If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **100%.** The recommended value is **200px**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **250 px**. **Note:** -If the value is 0, then the height of the control is set as per the default values. - In iOS devices, the height of the **Rich Text** control automatically resizes to fit contents. You cannot customize the height nor can you set the height to the default value.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	control. • Select **No** to bypass entering a value for the control. The default value is **No**.

Advanced

Property	Description
Id	Reference identification code for the control. This value is automatically generated and cannot be changed.
Include in Summary	Allow or deny a user to view the value in the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the Summary of a Base Form. • Select **No** to deny a user to view the value in the Summary of a Base Form. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form. -When the control is used in a **Base Form**, the formatting applied to the text is not reflected in the summary. The text appears as plain text.
Toolbar Options	Select the type of toolbar based on the toolbar styles and commands displayed. • Select **Standard** to display the standard toolbar styles and commands. • Select **Advanced** to display all the toolbar styles and commands. The default type is **Standard**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: In an iOS device, to use the **Advanced** toolbar in the **Rich Text** control outside any of the **Container** controls, make sure the height and width is set to values higher than the default values.

Note:
- Entering or editing values in the **Rich Text** control is not supported in the **Mozilla Firefox** browser.

- When you select a color from the color picker, make sure the form window is large enough to display the color picker completely to be able to select the text color and background color. If the window is small and the color picker is not displayed completely, you might not be able to select a color.

- Toolbar is not supported in devices.

- **Rich Text** control has the following limitations in iOS devices:
- You cannot open Hyperlinks present in the **Rich Text** control.
- The height of the **Rich Text** control automatically resizes to fit contents. You can neither customize the height nor can you set the height to the default value.
- When you set the width of the **Rich Text** control to the default value, the right hand side border is not displayed. Also, when you switch from landscape mode to portrait mode or vice versa, the text you type goes beyond the border of the control.
- Images take time to load.

- The script, object, and iframe tags are not supported in **Rich Text** control when added through **View HTML** window and cannot view it in the Form.

Scripts

For information about using scripts, see scripting guidelines for Rich Text control.

Role Lookup

Use **Role Lookup** control to search for roles, in the providers mapped to the repository, in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Default Value	The default value for the control. **To add the default value for the control** 1. Click the **Role Lookup** button. The **Default Value** dialog box appears. 2. In the **Search For** box, enter a search keyword. 3. Click **Search**. The search results appears. 4. Click a role in the search results, and then click **Save**. The selected role is set as the default value and the **Default Value** dialog box is closed. **To delete the default value for the control** • Click the **Clear** button. The selected role is deleted.

Appearance

Property	Description
Enable	Enable or disable the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Watermark	Hint text that appears when no value is entered for the control.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Include in Summary	Allow or deny a user to view the value in the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the **Summary** of a **Base Form**. • Select **No** to deny a user to view the value in the **Summary** of a **Base Form**. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Role Lookup control.

Signature

Use **Signature** control to capture a user's signature in a form by drawing on a touch enabled device. The signature is rendered as a picture and shown to the user in edit mode of the form (i.e. before save or submit). The Signature is saved in an encrypted format after the form is saved or submitted. The user cannot edit the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

saved signature, but has the option to undo/clear the Signature before the form is saved or submitted.

This control works across browsers in web and via the AVEVA Work Tasks Pro app.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. The default value is the value for the **Name** property without the spaces. **Note:** If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Top**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **400**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: If the value is 400, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass drawing signature for the control. • Select **Yes** to enforce drawing signature for the control. • Select **No** to bypass drawing signature for the control. The default value is **No**.

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Security Mode	Set the security mode to None or Encryption to secure the signature. • Select **None** to have the value of signature as Base64 encoded image. • Select **Encryption** to have the encrypted value for the signature. The default value is **Encryption**.

Scripts

For information about using scripts, see scripting guidelines for Signature control.

Note:
- The value for the Signature control can be more than 1024 characters, but this can restrict it from being injected as an attribute value in the Intouch OMI.
- Validation for the other controls, before the Signature control, are skipped due to Signature control's validation.

Slider

Use **Slider** control to set and view numerical data in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Default Value	The default value for the control. **Note:** Ensure the value is between 0 and 100.

Appearance

Property	Description
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	contain only alphanumeric characters.

Advanced

Property	Description
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.

Scripts

For information about using scripts, see scripting guidelines for Slider property.

Text

Use **Text** control to enter and view textual data in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Default Value	The default value for the control. **Note:** Ensure the value is within the **Minimum Length** and the **Maximum Length**.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Read-Only	Allow or deny a user to enter a value for the control. • Select **Yes** to deny a user to enter a value for the control. • Select **No** to allow a user to enter a value for the control. The default value is **No**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** -If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Watermark	Hint text that appears when no value is entered for the control.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Display Mode	Number of lines to show for the control. • Select **Single Line** to allow a user to enter one line of text for the control. • Select **Multi Line** to allow a user to enter multiple lines of text for the control. The default value is **Single Line**. **Note:** The **Multi Line** option is applicable only when the **Rows** property is set to **2** or more.
Rows	Number of rows for the control.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	The default value is **No**.
Text Pattern	Regular expression to validate the text for the control. For example, use the following regular expression to validate a text string as a phone number: `^[2-9]\d{2}-\d{3}-\d{4}$`
Minimum Length	Minimum length of the value for the control. The default value is **0**. **Note:** If the value is 0, then the minimum length for the control is not set.
Maximum Length	Maximum length of the value for the control. The default value is **0**. **Note:** If the value is 0, then the maximum length for the control is not set.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Include in Summary	Allow or deny a user to view the value in the **Summary** of a **Base Form**. • Select **Yes** to allow a user to view the value in the **Summary** of a **Base Form**. • Select **No** to deny a user to view the value in the **Summary** of a **Base Form**. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	- For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Text control.

Time

Use **Time** control to select a time in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Default Value	The default value for the control. **Note:** The value must be in 24-hour time format. Ensure the value is within the **Minimum Value** and the **Maximum Value**.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Read-Only	Allow or deny a user to enter a value for the control. • Select **Yes** to deny a user to enter a value for the control. • Select **No** to allow a user to enter a value for the control. The default value is **No**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**.
Minimum Value	Minimum time value for the control. The default value is **00:00:00**.
Maximum Value	Maximum time value for the control. The default value is **23:59:59**.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Time Format	Time format for the control. For example, hh:mm:ss tt. Use HH for 24-hour time. For more information, see Date and Time Formats. **Note:** - Time format must contain at least: --- Hour and minute components for 24-hour time. --- Hour, minute, and period components for 12-hour time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	- Time format is not validated. - For cultures, which do not support AM/PM, the “tt” or “t” must not be specified in the format.
Time Interval	Interval between two consecutive time slots to show in the list for the control. The default value is **01:00:00**.
Include in Summary	Allow or deny a user to view the value in the Summary of a Base Form. • Select **Yes** to allow a user to view the value in the **Summary** of a **Base Form**. • Select **No** to deny a user to view the value in the **Summary** of a **Base Form**. The default value is **No**. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for Time control.

Note:
- Time value is saved in text format and not in Universal Coordinated Time (UTC) format.
- Time value is in 24-hour time format.
- **Minimum Value** and **Maximum Value** properties together specify a range, hence their values cannot be the same.
- However, you can set a value for the **Default Value** property and then set the **Enable** property to **No** to restrict the time to a fixed value.

Tree

Use **Tree** control to display data in a tree view representation from a data source in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Data Source	Lookup data source used to populate the tree. **To configure the lookup source** 1. Click **Data Source** button to select the lookup data source. 2. In the **Data Source Type** list, select the type of data source. 3. In the **Data Source** list, select the data source. 4. Click **Save & Continue**.
Node ID Column	The column name of the column that contains IDs to identify every node used in the tree.
Node Parent Column	The column name of the column that contains IDs to identify parents of the nodes used in the tree. **Note:** If the node parent column is blank in the configuration, or if the parent ID specified does not

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	correspond to any node in the tree, then that node is placed on the root of the tree.
Node Label Column	The column name of the column that contains text to label the nodes used in the tree.

Appearance

Property	Description
Visible	Show or hide the **control**. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.
Expand Nodes	Expand or collapse the nodes of the tree by default when the tree loads. • Select **Yes** if you want the nodes of the tree to expand by default when the tree loads. • Select **No** if you want the nodes of the tree to collapse by default when the tree loads. The default value is **Yes**.

Advanced

Property	Description
On Data Bound	Run s script when data is bound to the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	For more information, see scripting guidelines for On Data Bound property.
On Selection Change	Run a script when selection of the record changes. For more information, see scripting guidelines for On Selection Change property.
On Check Change	Run a script when the state of the checkbox changes, that is when the checkbox is selected or cleared. For more information, see scripting guidelines for On Check Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Checkbox Behavior	Allow or deny a user to select single or multiple records for the control. • Select **Single** to allow a user to select a single record for the control. • Select **Multiple** to allow a user to select multiple records for the control. • Select **None** to deny a user to select a record for the control. The default value is **None**.
Node Checked Column	Column name for selecting the checkbox in the control.
Inherit Checked State	Allow or deny the system to automatically inherit the state of the parent checkbox for the child nodes when the checkbox of the parent node is selected or cleared. • Select **Yes** to allow the system to automatically inherit the state of the parent checkbox for the child nodes. For example,consider that **Oil** is the parent node

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	with **Crude** and **Refined** as child nodes. When you set **Inherit Checked State** to **Yes**, and if you select **Oil**, then **Crude** and **Refined** inherit the state of the parent node **Oil**, and are automatically selected. If **Oil** is unchecked, and if you select **Crude** and **Refined**, then **Oil** automatically sets to an indeterminate state (dash). If **Oil** is checked, and if you select **Crude** and **Refined**, then **Oil** remains in a checked state. **Note:** There is no visual indication of the parent node in an indeterminate state. • Select **No** to deny the system to automatically inherit the state of the parent checkbox for the child nodes. When you set **Inherit Checked State** to **No**, and if you select **Oil**, then **Crude** and **Refined** do not inherit the state of the parent node **Oil**, and do not get selected. The default value is **No**

Note: The **Tree** control is not supported in AVEVA Work Tasks Pro App.

Scripts

For information about using scripts, see scripting guidelines for Tree control

User Lookup

Use **User Lookup** control to search for users, in the providers mapped to the repository, in your form.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces. **Note:** - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Tooltip	Message that appears when the pointer hovers over the control.
Default Value	The default value for the control. **To add the default value for the control** 1. Click the **User Lookup** button. The **Default Value** dialog box appears. 2. In the **Provider** box, click a provider. 3. In the **Search For** box, enter a search keyword. 4. In the **In** list, click a search criteria. 5. Click **Search**. The search results appears. 6. Click a user in the search results, and then click **Save**. The selected user is set as the default value, and the **Default Value** dialog box is closed. **To delete the default value for the control** • Click the **Clear** button. The selected user is deleted.
Show Default User	Show or hide the logged on user as the default user for the control. • Select **Yes** to show the logged on user as the default user for the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **No** to hide the logged on user as the default user for the control. The default value is **Yes**.

Appearance

Property	Description
Enable	Enable or disable the control. • Select **Yes** to enable the control. • Select **No** to disable the control. The default value is **Yes**.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	placed on a **Base Form** with **Label Position** property set to **Top**, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Watermark	Hint text that appears when no value is entered for the control.
Width	Width in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the width of the control is set as per the default values.
Height	Height in pixels for the control. The default value is **0**. **Note:** If the value is 0, then the height of the control is set as per the default values.
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Validation

Property	Description
Mandatory	Enforce or bypass entering a value for the control. • Select **Yes** to enforce entering a value for the control. • Select **No** to bypass entering a value for the control. The default value is **No**.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Extended Information	Allow or deny a user to access the extended information (that is columns other than the display and value columns of the result set) based on the selection for the control. • Select **Yes** to allow a user to access the extended information based on the selection for the control. • Select **No** to deny a user to access the extended information based on the selection for the control. The default value is **No**.
Include in Summary	Allow or deny a user to view the value in the **Summary** of a **Base Form**. • Select **Yes** to allow a user to view the value in the **Summary** of a **Base Form**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Select **No** to deny a user to view the value in the **Summary** of a **Base Form**. The default value is No. **Note:** - This property is applicable only when the control is used on a **Base Form** with **Repeating** property set to **True**, and the form is viewed on a small form factor mobile device. - For information about **Summary** of a **Base Form**, see Base Form.

Scripts

For information about using scripts, see scripting guidelines for User Lookup control.

Widget

Use **Widget** control to create reusable graphical components. You can maintain a library of widgets for the other users to use.

Note: If you add or modify any standard widget files, it should be manually added to the package in the 'Additional files' section of the packages.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Description
Name	Name that identifies the control. This value is also used for the label of the control.
XML Node	XML node mapped to the control. Use this value to map a list to a form and a form to a workflow. • Prefix with **@** to use the value of the control as an attribute with the immediate parent node. For example, `@FirstName`. • Use **/** for a parent-child node configuration to uniquely identify a control in a form with multiple containers. For example, `Employee/FirstName`. The default value is the value for the **Name** property without the spaces.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	Note: - If you use **/** to denote a parent which is not a container and the parent is at the same level as that of the child, then the XML node for the control is not generated. - If you edit this value later, then it might affect the mapping between forms, lists, and workflows.
Description	Description for the control.
Default Value	The default value for the control.
Widget Type	Select the widget type from the drop-down list. The widget types available are: Inline, Embed, and Form.
Widget	Select the available widget. This property is available when the selected widget type is **Inline** and **Embed**.
Configuration	Select the available configuration. This property is available when the selected widget type is **Inline** and **Embed**.
Form Details	Applicable for **Form** widget type. Provide the **Form Details**. To select the form, 1. In the **Form Name** box, select a form name. 2. In the **Form Version** box, select a form version. 3. Click **Save & Continue**. The selected form is embedded in the widget, and the **Form Details Configuration** page is closed. For more information, see Configure Communication between Widget Control and Form.

Appearance

Property	Description
Style	Height and Width in pixels for the control. The default value for Height and Width is **0**. **Note:** If the value is 0, then the height and width of the control is set as per the default values.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	- The characters **< > &** are not supported. - The only supported value for height and width is the pixel.
In Designer	Option to view the space taken by the Widget as per the Style property in Design mode.
Visible	Show or hide the control. • Select **Yes** to show the control. • Select **No** to hide the control. The default value is **Yes**.
Label Position	Position of the label for the control. • Select **Default** to allow the system to align the label. The system aligns the label to the left of the control or as specified by the container control. • Select **Left** to align the label to the left of the control. • Select **Top** to align the label to the top of the control. • Select **Right** to align the label to the right of the control. • Select **Bottom** to align the label to the bottom of the control. • Select **Suppress** to hide the label for the control. The default value is **Suppress**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label Position** property of the container control determines the position of the label for the control. - For example, if the value is **Default** and the control is placed on a **Base Form** with **Label Position** property set to Top, then the label is aligned to the top of the control.
Label-Control Area	Area occupied by the label and the control in percentage. • Select **Default** to allow the system to set the label

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	and the control areas. • Select **Custom** to specify the label area. The control area is automatically set based on the label area. The default value is **Default**. **Note:** - If the value is **Default** and the control is placed on a container control, then the value of the **Label-Control Area** property of the container control determines the area of the label for the control. - This is applicable only when the **Label Position** is set to **Left** or **Right**.
Consider Available Height	Resize the height of the control or container based on the available height. • Select **Yes** to resize the height based on available height. • Select **No** to retain the space. The default value is **No**. **Note:** - The Consider Available Height property is not applicable for the following: -Designer view -Print form -Panel form with horizontal tab order - If this property is set to **Yes** and a value is specified in the **Height** property, minimum height is considered. - This property is applicable only in web view. - When more than one control or container is set with **Consider Available Height** at the same level of container or form, then the available space is distributed evenly between those controls or containers. For example, consider a Form which contains two Panel forms with **Consider Available Height** set to **Yes**, where Panel form 1 has two TextInput controls and Panel form 2 has one TextInput control. The Panel form 1's height is more than Panel form 2's height as Panel form 1 contains two TextInput controls and the available empty space in the form is evenly distributed between the two Panel forms.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Retain Space	Reserve or remove space when the control is hidden. • Select **Yes** to keep space when the control is hidden. • Select **No** to remove space when the control is hidden. The default value is **No**.
Custom Style Sheet Identifier	Prefix for the CSS class name in the custom style sheet. For information about using custom style sheets, see Custom Styles. **Note:** The value must start with an alphabet and can contain only alphanumeric characters.

Advanced

Property	Description
On Data Change	Run a script when value of the control changes. For more information, see scripting guidelines for On Data Change property.
ID	Reference identification code for the control. This value is automatically generated and cannot be changed.
Add Values as Url	Option to add values as URL parameters. • Select **Yes** to enable adding values as URL parameters. • Select **No** to disable adding values as URL parameters. For more information, see Configure Communication between Widget Control and Form.
Widget Properties	The widget properties are listed based on the **Widget** selected in the control, when the **Widget Type** is **Inline** or **Embed**. See the respective Widget documentation for the available widget properties. • Grid Widget

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	• Hierarchy Selector Widget If the **Widget Type** is **Form**, the Form Parameters are listed based on the **Form Details** property of the control. For more information, see Configure Communication between Widget Control and Form. **Note**: Form Parameters are considered as **Widget Properties** for the Widget control irrespective of the data type. The **Widget Properties** are always represented as String data type.

Scripts

For information about using scripts, see scripting guidelines for Widget control.

Dynamic Widget Properties and Events

The widgets support dynamic properties and events. The following dynamic properties and events are defined in a **wjson** file. Localizing the dynamic properties and events is supported through a localization text file in AVEVA Work Tasks.

Widget	Property	Event
Grid	notify	onCellChange
	data	onInitialize
	prevRowValue	onRowSelection
	newCellValue	
	modifiedRows	
	selectedRow	
	selectTopGridItem	
Hierarchy Selector	data	onInitialize
	selectedObj	onSelectionChange
	selectedvalue	
	selectNode	

Each widget has its own **wjson** file (**BaseGrid.wjson**, **HierarchySelector.wjson**). You can use the corresponding widget wjson file to update the names of its properties and events accordingly.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

You can access the **wjson** file in the following folders:

- *<WorkTasksInstallPath>***\Web\BPMUITemplates\Default\NextGenForms\custom\Widgets\BaseGrid**
- *<WorkTasksInstallPath>***\Web\BPMUITemplates\Default\NextGenForms\custom\Widgets\HierarchySelector**

Add or Edit a Dynamic Property or Event

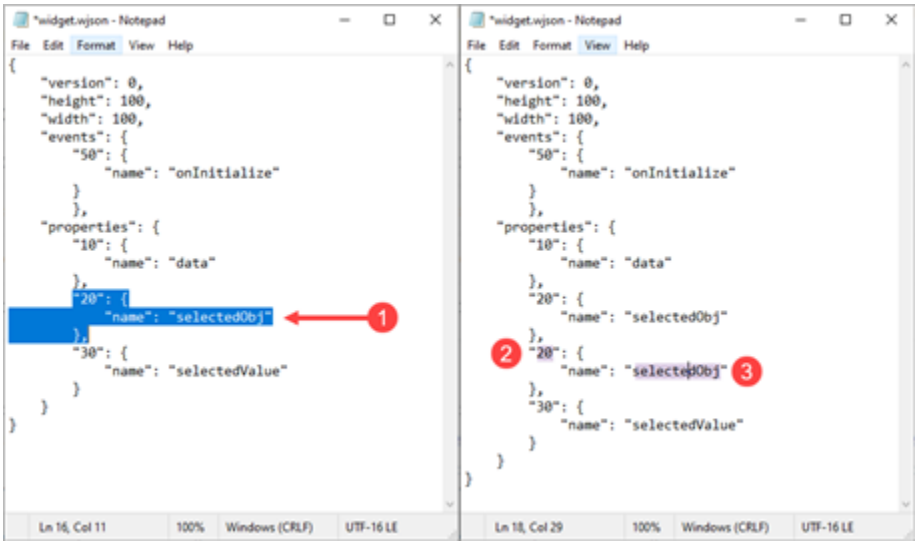
Update the corresponding section of the **wjson** file to add or edit a dynamic property or event.

Note: Whenever you update the **wjson** and localization text files, you need to reset the IIS.

1. Depending on the task you want to do, perform either of the following:
   - To add a new dynamic property or event:
     a. Copy an existing item and paste into the section where you want to insert it.
     b. Assign a unique sequence number. Any number that isn't used yet in the file. If there's a duplicate sequence number, the application displays the first property in the list and ignores the other one.
     c. Add the new name for the property or event and save the file.

     Note: Ensure that you don't miss a syntax element in the file such as comma (,). If something is missed, the properties are not shown in the Properties page. The issue related to this is logged in the WTLogger Console.

     The following example is a **wjson** file for the Hierarchy Selector widget.



   - To edit an existing property or event entry, update the name of the corresponding property or event, and save the file.
2. If you are editing or updating the name of an existing property or event, you also need to update the localization text file.

   You can access these localization files from *<AVEVAWorkTasksInstallPath>***\Locales\en-US\Forms2\Standard**.
   - **Forms_AWT***<VersionNumber>***_BaseGrid.txt**
   - **Forms_AWT***<VersionNumber>***_HierarchySelector.txt**

     If you want to set a newly created property or event for localization, add it to the localization file. Follow

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the existing text format to avoid encountering some issues.

Note: If the name of the new property or event isn't added in the localization file, the application takes the name defined in the **wjson** file.

Grid Widget

The Grid widget allows you to display a dynamic data based on a configured dataset. With this widget, you can configure:

- How data is rendered within the grid
- What text to display in the column headers
- Whether the data in a cell or column is editable or read only. If it is editable, whether it is an input field or a dropdown list
- A tooltip to display the full details of a truncated data in a cell
- The column width

  Note: An option to adjust the column width using the mouse is available.
- Column data filtering, sorting, and reordering
- The number of records to display per page
- Column data types and cell background color

Properties

The Grid widget uses the following properties and some of them are defined as a function in a configuration file. For information on how the properties are used, see Grid widget properties usage. The following table describes the supported Grid properties:

Property	**Supported datatype**
Notify	Object *This is an object converted to a JSON string through JSON.stringify() method*.
Data	Object *This is an object converted to a JSON string through JSON.stringify() method*.
Previous Row Value	Object *This is an object converted to a JSON string through JSON.stringify() method*.
New Cell Value	String
Modified Rows	Object *This is an object converted to a JSON string through JSON.stringify() method*.
Selected Row	Object

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Supported datatype
(selectedRow)	*This is an object converted to a JSON string through JSON.stringify() method*.
Select Top Grid Item (selectTopGridItem	Object *This is an object converted to a JSON string through JSON.stringify() method*.

Limitations

- Tooltip is not supported in datetime and numeric datatypes.
- The widget's numeric text box has a precision limitation of up to 16 digits only. Numbers longer than 16 digits are converted to exponential numbers and lose their precision.

Grid Widget Properties Usage

The Grid widget is using the following properties to handle incoming or outgoing data and event.

Notify

This property is used for data validation inside the configuration file to pass a validation script in the **On Cell Change** event property of the widget in the Properties' **Scripts** tab. For example validation script, see Validate Data.

This is also used to provide additional functionality like in the **//notify operations** section of the code, where you can use the filter, export, reset, and clear filter operations through scripting.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//notify operations
cwidget.on("notify", function () {
    var grid = $(widgetObject.baseGrid.domElement).children("#kendogrid").data("kendoGrid");
    var notifyobj = JSON.parse(cwidget.notify);
    var functionName = "";
    var fileType = "";
    var field = "";
    var operator = "";
    var value = "";
    var validationResult = false;
    if (notifyobj != null && notifyobj.Functionname != null)
    {
        functionName = notifyobj.Functionname.toUpperCase();
    }
    switch (functionName) {
        case "EXPORT":
            if (notifyobj.Filetype != null)
            {
                fileType = notifyobj.Filetype.toUpperCase();
            }
            if (fileType == "PDF") {
                grid.saveAsPDF();
            }
            else if (fileType == "EXCEL") {
                grid.saveAsExcel();
            }
            break;
        case "RESETSORTORDER":
            grid.dataSource.sort({});
            break;
        case "CLEARFILTER":
            grid.dataSource.filter({});
            break;
        case "FILTER":
            field = notifyobj.field;
            operator = notifyobj.operator;
            value = notifyobj.value;
            grid.dataSource.filter({
                "field": field,
                "operator": operator,
                "value": value
            });
            grid.dataSource.read();
            break;
        case "VALIDATE":
            validationResult = notifyobj.result;
            if (validationResult == false){
                e = widgetObject.baseGrid.event;
                e.preventDefault();
            }
            break;
```

For example, you can use the following scripts when you want to export data to either PDF or Excel:

```
control.findByXmlNode("grid1").widgetProperties.notify =
'{"Functionname":"Export","Filetype":"Excel"}';
```

```
control.findByXmlNode("grid1").widgetProperties.notify =
'{"Functionname":"Export","Filetype":"Pdf"}';
```

Note: This operation only exports the data that is currently displayed on the grid, but not the entire list.

For information on other scripts, see the following topics:

- Clear Data Filters
- Reset Grid Data Sort Order Setting
- Filter Data through Scripting

Data

This property is used for incoming data. To bind the data to the Grid widget container, you can use the following example script in an **On Click** event.

```
try {
debugger;
var lookup = SFU.getLookupSchemaAndData("UOMLookup", null, true);
if (lookup.DisplayColumn === "" && lookup.ValueColumn === "" && lookup.Data.length === 0)
throw new Error("Lookup not found!");
control.findByXmlNode(“grid1”).widgetProperties.data = JSON.stringify(lookup.Data);
} catch (error) {
SFU.showError("Lookup Failed!");
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

In the configuration file, you need to define all the column headers under the **//Columns** and **//Field names with datatypes** sections**.** See the **BaseGrid_conf.js** file.

Previous Row Value and **New Cell Value**

These properties are also used for validation, where the **New Cell Value** property is used to get the edited cell value and **Previous Row Value** is for the old row value.

Modified Rows

This property is used to get the list of rows that are modified with their current values. This syncs with the cell changes. If no modifications are made, this property is empty.

Selected Row

This property is used to retrieve the corresponding value for each column in a selected row of a Grid widget.

Example

```
try{
var selectedrows =
JSON.parse(control.findByXmlNode("Widget1").widgetProperties.selectedRow);
console.log(selectedrows);
}
catch(e){
}
```

Select Top Grid Item

This property is used to select the first record in a Grid widget for every new session or when the data is loaded for the first time. You need to use this with the **data** property in order to work.

For example, you can use the following script in an **On Initialize** event.

```
control.findByXmlNode("Widget1").widgetProperties.selectTopGridItem =JSON.stringify(true);
var testData = [{
"uom_id": 1,
"description": "Pieces",
"oper_id": "Filling",
"canSchedJobs": true,
"seq_no": 0,
"state_desc": "NEW",
"color": 16776960,
"req_qty": 300000000,
"qty_prod": 100,
"last_edit_comment": "",
"req_finish_time": "2014-02-10T15:20:08Z"
},
{
"uom_id": 2,
"description": "Gram",
"oper_id": "Filling",
"canSchedJobs": true,
"seq_no": 0,
"state_desc": "READY",
"color": 65535,
"req_qty": 600,
"qty_prod": 300,
"last_edit_comment": "",
"req_finish_time": "2014-02-13T15:20:08Z"
}, {
"uom_id": 3,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"description": "Count",
"oper_id": "Filling",
"canSchedJobs": false,
"seq_no": 0,
"state_desc": "RUNNING",
"color": 65280,
"req_qty": 800,
"qty_prod": 500,
"last_edit_comment": "",
"req_finish_time": "2014-02-12T15:20:08Z"
}];
control.findByXmlNode("Widget1").widgetProperties.data=JSON.stringify(testData);
```

Grid Widget Event

The Grid widget supports the following events:

- On Cell Change
- On Initialize
- On Row Selection

On Cell Change

Use scripts for the **On Cell Change** event of a Grid widget to validate changes to a cell value.

This event gets the value for the **modifiedValue** and **oldValue** properties of the Grid widget. For more information on validating changes, see Validate Data.

Example

If you want to verify the modified value of a column, you can have a script like in the following example. This verifies that the modified value of the description column is "desc".

When you modify a cell value and remove focus from the cell, the **On Cell Change** event is triggered.

```
var validate = {};
var validate1 = function(validate)
{
var modified = (JSON.parse(event.modifiedvalue));
//To get a modified value of a column name (example: description)
var description = (modified).description;
// To check a column value (example: description column has a value of a sample string
value "desc")
if (description=="desc")
{
validate.result = true;
}
else
{
validate.result = false;
}
validate.Functionname = "Validate";
return validate;
};
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.findByXmlNode("Widget1").widgetProperties.notify=JSON.stringify(validate1(validate)
);
```

To get the previous row value of a Grid widget, you can use the following script:

```
var oldvalue = (JSON.parse(event.oldvalue));
// To get previous value of a column name: abbreviation
var oldabb = (oldvalue).abbreviation;
// To get previous value of a column name: description
var olddes = (oldvalue).description;
```

On Initialize

Use scripts for the **onInitialize** event of a Grid widget to accomplish a desired operation after the widget control is initialized, but before it is rendered to the UI.

Example

If by default you want to load the actual dataset from a database lookup once the widget is launched, add the script on the **On Initialize** event.

On Row Selection

Use scripts for the **onRowSelection** event of a Grid widget to return a dataset for a selected row in a Grid widget. You can use this with the **selectedRow** property.

Note: This event is ignored when filtering or sorting data, or resizing the Grid widget.

Example

```
try{
var selectedrows =
JSON.parse(control.findByXmlNode("Widget1").widgetProperties.selectedRow);
console.log(selectedrows);
}
catch(e){
}
```

Configure the Grid Widget

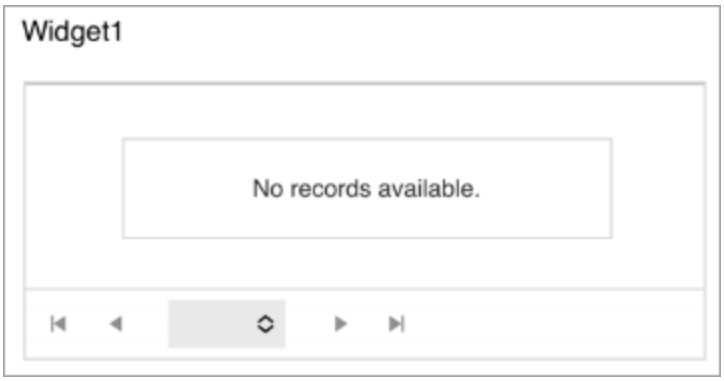
Modify specific sections in the configuration file to meet your required widget settings.

Pass Data to or from the Widget

You can pass data to or from the Grid widget either through a Web API or Database lookup. You need to define the data columns in the configuration file for them to appear in the widget.

If the schema section of the configuration file is missing, removed, or commented out, the widget is shown without columns and looks something like this:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



If the schema is available in the configuration file but no data is defined, the widget appears only with the schema columns and looks something like this:

Note: The screenshots are from an iOS device.

Define Data Columns

Define Data Columns

The data columns must be defined in the following sections of the configuration file to load your own dataset:

- **//Columns**
- **//section for cell datatypes and input field**

Some column headers are listed by default, just like in the following screenshot. You can replace them with the ones in your own dataset.

Note: Hiding column headers can only be done in the JavaScript configuration file and not through the lookup configuration.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// Columns

columns: [{
    field: "wo_id",
    //title: "Workorder",
    title: "Id",
    width: '180px',


},
{
    field: "ProductName",
    width: "200px",
    title: WW_localize("@@WidgetBaseGridProductName@@"),


},


{
    field: "CategoryID",
    width: "200px",
    title:  WW_localize("@@WidgetBaseGridCategory@@"),
    hidden: true,

    template: templateFunction,
    editable: function () {
        return false
    }
},
{
    field: "oper_id",
    title: "Operation",

}, {
    field: "seq_no",
    title: "Seq No",
```

Grid widget supports localization. You have the option to localize the column headers or Grid titles. In the screenshot, the **title** field is used to define the localization's key-value pair. For information on localization, see Localize Widgets.

Clear Data Filters

Clear Data Filters

If you want to clear the currently selected data filter, you can use the following example script with the **On Click** event of a Work Task control, together with the widget's **Notify** property .

```
control.findByXmlNode("grid1").widgetProperties.notify = '{"Functionname":"CLEARFILTER"}';
```

Reset Grid Data Sort Order Setting

Reset Grid Data Sort Order Setting

If you want to reset the sort order setting of the Grid data, you can use the following example script with the **On Click** event of a Work Tasks control.

```
control.findByXmlNode("grid1").widgetProperties.notify =
'{"Functionname":"RESETSORTORDER"}';
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Validate Data

Validate Data

The Grid widget allows you to perform validation at cell level. You can validate data when a user updates a value. Some of the validation rules are provided by default as part of the Grid widget's configuration file (**BaseGrid_SampleData_conf.js**), and include:

- Limiting the number of character string value
- Limiting the number of decimal places for numeric data fields
- Required fields

You can use the default validation or you can perform validation through scripting outside of the widget using other controls. If you create external validation scripts, the Grid widget handles them through a boolean result flag. The following are example validation scripts that the Grid widget recognizes, which you can add to the **On Cell Change** event property of the widget:

```
validate = {}; validate.result=true; validate.Functionname = 'Validate';
control.findByXmlNode("grid1").widgetProperties.notify=JSON.stringify(validate);
```

```
validate = {}; validate.result=false; validate.Functionname = 'Validate';
control.findByXmlNode("grid1").widgetProperties.notify=JSON.stringify(validate);
```

In the example scripts, if the result is true, the widget accepts the new value. If the result is false, the cell value returns to its previous value.

Old and modified values

You can use the following example scripts to get the previous row and new cell values of the Grid with the **On Cell Change** event property of the widget.

Old row value

```
var oldval = (JSON.parse(event.oldvalue));
```

For example, if you want to get the previous row value of a **uom_id** column cell:

```
var uom = JSON.parse(oldval).uom_id;
```

Modified cell value

```
var modified = (JSON.parse(event.modifiedvalue));
```

For example, if you want to get the new cell value of a **uom_id** column cell:

```
var uom = JSON.parse(modified).uom_id;
```

Disabling default validation

If you want to do the validation outside the widget, remove or comment out the validation section in the code.

The highlighted validation section in the following screenshot is for validating the required fields and character string limit respectively.

Note: Character string limit validation works on edit mode and not on initial data load.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
118             // section for cell datatypes and input field.
119             schema: {
120                 model: {
121                     id: "uom_id",
122                     fields: {
123                         uom_id: { type: "number", editable: false },
124                         description: {
125                             editable: true,
126                             defaultValue: "",
127                             validation: {
128                                 required: true,
129                                 maxlength:"10",
130                             }
131                         },
```

The validation for limiting the number of decimal places is added to one of the example fields. Search for the **weightEditor** line in the code, which serves as the placeholder code for limiting the number of decimal places.

```
222             field: "req_qty",
223             title: "Req Qty",
224             attributes:{class:"number"},
225             editor: weightEditor,
226              template: function (dataItem) {
227                 return widgetObject.baseGrid.formattedNumber(dataItem.req_qty,3);
```

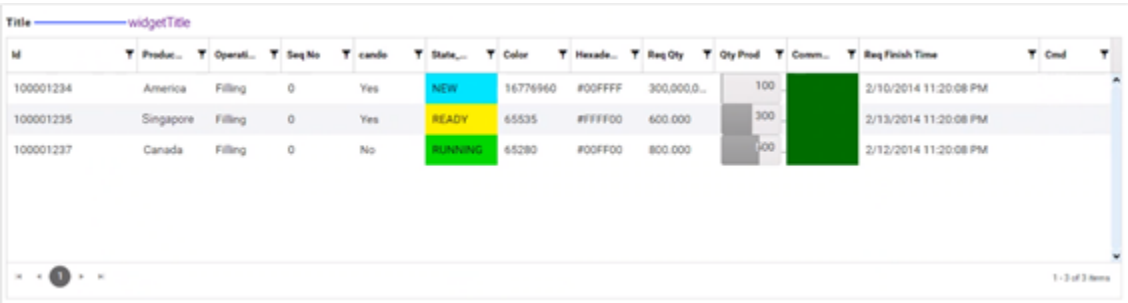
Grid Widget Configurable Parameters

Configurable parameters are provided in the configuration file to give you the option to change widget default settings.

Modify Grid Widget Title

Modify Grid Widget Title

The Grid widget's user interface label property that you can configure is marked in the following screenshot. Localizing this property is also supported. For localization information, see Localize Widgets.



To modify the title, under the code's **var conf** line, update the **widgetTitle** property accordingly. By default, this property is mapped to the value defined in the localization text file (**Forms_AWT2020U2_BaseGrid.txt**). You can access the text file from *<AVEVAWorkTasksInstallPath>***\Locales\en-US\Forms2\Standard**.

```
84          var conf = {
85              widgetTitle: WW_localize("@@WidgetBaseGridWidgetTitle@@"),
```

If you don't want to localize this property, simply enter the title that you want to appear in the UI. You can either remove the "**WW_localize...**" value or comment it out as highlighted in the following screenshot.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
var conf = {
    widgetTitle: "GridTitle", //WW_localize("@@WidgetBaseGridWidgetTitle@@")
```

Modify Column Width and Column Headers

Modify Column Width and Column Headers

You can modify column width and column headers using the **width** and **title** properties under the **//Columns** section of the configuration file. The default value for the width in each of the columns enables the widget to properly render and adjust data display. A scroll bar appears if there are more columns to display.

Note: If a **title** is not specified, the value for the **field** property is displayed.

```
// Columns

columns: [{
    field: "wo_id",
    //title: "Workorder",
    title: "Id",
    width: '180px',


},
{
    field: "ProductName",
    width: "200px",
    title: WW_localize("@@WidgetBaseGridProductName@@"),


},


{
    field: "CategoryID",
    width: "200px",
    title:  WW_localize("@@WidgetBaseGridCategory@@"),
    hidden: true,

    template: templateFunction,
    editable: function () {
        return false
    }
},
{
    field: "oper_id",
    title: "Operation",

}, {
    field: "seq_no",
    title: "Seq No",
```

Set Number of Records to Display in a Grid Page

Set Number of Records to Display in a Grid Page

You can set the number of records or rows to display on each grid page using the **pageSize** property under the **//Options** section of the configuration file. The Grid loads a maximum of 1000 records.

If in a widget control (or container), the **Consider Available Height** setting is set to *Yes*, the page size value that you set serves as the minimum number of rows to display. If there's an available white space, such as when resizing the browser window so that it increases the height of the widget, the Grid widget fills it with more rows or records.

If the **Consider Available Height** setting is set to *No* and you specify a fixed height, the Grid widget takes the page size value and fills the available white space accordingly based on the fixed height. When resizing the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

browser window, the size of the Grid widget remains the same; it still has the same height and the same displayed records, and only the browser window size changes.

```
// Options
noRecords: true,
filterable: {
    mode: "menu"
},
resizable: true,
sortable: {
    mode: "multiple",
    allowUnsort: true,
    showIndexes: true
},
editable: true,
selectable: true,
scrollable: true,
persistSelection: true,
pageable: {
    pageSize: 7,
    position: "bottom",
    responsive: true
},
```

Configure Cell Datatypes and Input Field

Configure Cell Datatypes and Input Field

You can configure the cell datatypes and input field under the **//section for cell datatypes and input field** section of the code within the configuration file. In this section, you can modify the listed fields, input field behavior, and their datatypes.

```
// section for cell datatypes and input field.
schema: {
    model: {
        id: "uom_id",
        fields: {
            uom_id: { type: "number", editable: false },
            description: {
                editable: true,
                defaultValue: "",
                validation: {
                    required: true,
                    maxlength:"10",
                }
            },

            period1: { type: "string", editable: true },
            period: { type: "string", editable: true },

            CategoryID: {
                editable: true,
                field: "CategoryID",

            },
            oper_id: { type: "string", editable: false },
            canSchedJobs: { type: "boolean", editable: true },
            seq_no: { type: "number", defaultValue: 0, editable: true, validation: { required: true, min: 0 } },
            state_desc: { editable: false },
            color: { type: "number" },
            req_qty: { type: "number", editable: true },
            qty_prod: { editable: true },
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure Column Background Color

Configure Column Background Color

You can configure the column background color under the **//column background color** section.

```
, {
   field: "last_edit_comment",
   title: "Comment",
   //column background color
   attributes: {
       style: "background-color: green",
   },
```

Add Column-value Tooltip

Add Column-value Tooltip

Enable tooltip for each column row value at the column level. To add a tooltip, add a **template** key right after the column header, as shown in the following screenshot.

```
   {
       field: "description",
       template: "<span title='#:kendo.toString(data.description, \"n4\")#'>#:description#</span>"
   },
```

Grid widget UI controls

The following are some of the UI controls that you can use within the Grid widget:

- Datetime picker
- Drop-down list
- Progress bar
- Textbox
- Number control

Most of the codes that enable or add these controls are already available in the configuration file (**BaseGrid_SampleData_conf.js**) as placeholder codes. If you don't want to use them, you can just remove or comment out the sections.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Add Progress Bar in Grid Column

Add Progress Bar in Grid Column

You can add a progress bar through the following code updates. There are two sections in the configuration file that you need to update to add a progress bar.

- In the **//Columns** section where all the column headers are defined, add a **template** key in the column header where you want to add a progress bar. The following screenshot is an example from the configuration file.

```
230         }, {
231             field: "qty_prod",
232             title: "Qty Prod",
233             //progress bar
234             template: "<div class='progress'  style='width: 100%;'></div>"
235         }, {
```

- In the **//adding progress bar** section, you can update the code similar to the following screenshot, depending on your requirement.

```
300             //adding progress bar
301             grid.tbody.find(".progress").each(function (e) {
302                 var row = $(this).closest("tr");
303                 var model = grid.dataItem(row);
304                 $(this).kendoProgressBar({
305                     max: model.req_qty,
306                     value: model.qty_prod,
307                     change: function (e) {
308                         e.progressWrapper.css({
309                             "background-color": "green",
310                             "border-color": "blue"
311                         });
```

Add Datetime Picker

Add Datetime Picker

Add a datetime picker for a particular field using the code under the **//adding Datetime picker** section of the configuration file. You can comment it out if you don't want to use it.

```
254 field: "req_finish_time", title: "Req Finish Time", format: "{0:dd MMM yyyy HH:mm:tt}", width: '250px',
255 //adding Datetime picker
256 editor: function (container, options) {
257         var input = $("<input/>");
258         input.attr("name",options.field);
259         input.appendTo(container);
260         input.kendoDateTimePicker({});
261 },
262 template: function (dataItem) {
263     return widgetObject.baseGrid.formattedDateTime(dataItem.req_finish_time);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Add Drop-down List

Add Drop-down List

You can use the following codes to have a drop-down list instead of a text input field. There are placeholder sections in the configuration file that you need to update to have a drop-down list in your own dataset.

- The **//adding drop down** section that defines the drop-down function. Update the code depending on which column you want to add the drop-down list.

  In the following code, the **"comment"** value is linked to the **var testDataCat** section of the code. If you use this code as is, that **testDataCat** section is where the values for the drop-down list are defined.

```
86      //adding drop down
87      widgetObject.baseGrid.categoryDropDownEditor = function (container, options) {
88
89          $('<input required name="' + options.field + '"/>')
90              .appendTo(container)
91              .kendoDropDownList({
92                  autoBind: false,
93                  dataTextField: "comment",
94                  dataValueField: "comment",
95                  dataSource: {
96                      //type: "odata",
97                      transport: {
98                          read: widgetObject.baseGrid.CatDataSourceRead
99                      }
100                 }
```

- For every column field to which you want to put a drop-down list, you need to have the following highlighted code, modified based on your dataset.

```
236                 field: "last_edit_comment",
237                 title: "Comment",
238                 //column background color
239                 attributes: {
240                     style: "background-color: green",
241                 },
242                 editor: widgetObject.baseGrid.categoryDropDownEditor,
243                 //dropdown
244                 template: function (dataItem) {
245
246                     if (typeof (dataItem.last_edit_comment) == 'undefined')
247                         dataItem.last_edit_comment = "";
248
249                     return dataItem.last_edit_comment;
250                 }
```

- The **var testDataCat** section. This is only for test values.

```
299     var testDataCat = [{
300         "comment_id": 1,
301         "comment": "Positive2",
302     }, {
303         "comment_id": 2,
304         "comment": "Neutral2",
305     }, {
306         "comment_id": 3,
307         "comment": "Negative2",
308     }];
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Sort, Filter, and Reorder

The Grid widget comes with sorting, filtering, and reordering features. Sorting data is based first on datatype and then sorted in alphabetical order for a single column at a time.

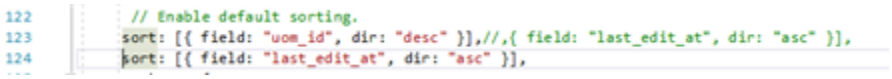
Filtering data can be applied to multiple columns. The grid persists the order of sorted and filtered data, and when you refresh the data through scripting, it keeps the sorted or filtered order too.

Note: The funnel icon is highlighted to indicate a filtered column.

Change Default Sorting Behavior

Change Default Sorting Behavior

You can change the default sorting behavior in the **//Enable default sorting** section of the configuration file.



Filter Data through Scripting

Filter Data through Scripting

You can use the following script with any of the Work Tasks controls to filter data.

```
control.findByXmlNode("grid1").widgetProperties.notify =
'{"Functionname":"FILTER","field":"description","operator":"startswith","value": "g"}';
```

Column Data Alignment

The column data is left aligned by default. If you want to align the data to the right, you are given the option to do that in each of the columns.

In the configuration file, add a line for **attributes** to set the class defined in the **Grid.style.css** file. You can find the css file in this location: <*AVEVAWorkTasksInstallPath*>**\Web\BPMUITemplates\Default\NextGenForms\custom\Widgets\BaseGrid\css**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Hierarchy Selector Widget

The Hierarchy Selector widget allows you to navigate to a multi-level dataset, and select a particular item class or category depending on the dataset you use it with.

This widget has a parent-child structure and requires a minimum of 2 levels. It accommodates the highest to the lowest level of data, where the parent node or highest level item can have another level of data.

With this widget, you can configure:

- The widget title, header, and the parent-child node section captions
- Color for selected parent and child nodes

Properties

The Hierarchy Selector widget uses the following properties and defines them as a function in a configuration file. For information on how the properties are used in the configuration file, see Hierarchy Selector Widget Properties Usage.

The following table lists the supported Hierarchy Selector properties and the type of data it accepts.

Property	Supported Datatype
Data	Object *This is an object converted to a JSON string through JSON.stringify() method*.
Selected Object	Object *This is an object converted to a JSON string through JSON.stringify() method*.
Selected Value	*This depends on the type of the returned property, which is defined in the* **nodesMapping** *or* **defaultNodesMapping** *section of the configuration file. The returned value is converted to a JSON string.*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Hierarchy Selector Widget Properties Usage

The following sections describe how the Hierarchy Selector widget properties are used in the configuration file.

Data

This property is defined as a function and this is called for incoming data.

```
//Subscribe to data property to handle new set of parent child node
cwidget.on('data', function () {
    if (typeof cwidget.data != 'undefined' && cwidget.data !== '') {
        try {
          // pass in data will be stringify
          widgetObject.hs.updateWidget(JSON.parse(cwidget.data));
        }
        catch(ex){
          alert(WW_localize("@@WidgetHSErrorSetDataConvertToObject@@")+ ex);
        }
    }
      else{
        alert(WW_localize("@@WidgetHSErrorSetDataConvertToObjectFromEmptyUndefined@@"));
    }
});
```

Incoming data is handled through the **defaultNodesMapping**, **nodesMapping,** and **nodesValue** sections in the configuration file.

The dataset you are using needs to follow the **nodesValue** structure. If there are changes to the **nodesValue** properties, you need to update the **nodesMapping** properties to match those changes. If there are no changes, then you can remove the **nodesMapping** section from your script. The configuration file will then use the **defaultNodesMapping** settings.

Under the **nodesValue** section in the following screenshot is the example data structure that the widget recognizes.

The entire tree structure needs to be passed to the widget and is not necessarily built as you select node.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
var testData =
{
  "nodesMapping" :
  {
    "groupName" : "GroupName",
    "groupNodes" : "UtilizationReasonGroups",
    "showGroupAsEnabled": "IsEnable",
    "childNodes" : "UtilizationReasons",
    "childName" : "Name",
    "childColor" : "EquipmentState.Color",
    "selectedChildNodeValue": "Name"
  },

  "nodesValue" :
  [{
    "GroupName": 'Changeover',
    "IsEnable" : false,
    "UtilizationReasonGroups": [],
    "UtilizationReasons":
    [{
      "Name": 'Changeover1',
      "EquipmentState":
      {
        "Name": 'DOWNTIME',
        "Color": '#DC0A0A',
      },
    },
    {
      "Name": 'Changeover2',
      "EquipmentState":
      {
        "Name": 'RUNNING',
        "Color": '#009848',
      },
    },
    {
      "Name": 'Planned Changeover',
      "EquipmentState":
      {
        "Name": 'PLANNED DT',
        "Color": '#F5A623',
      },
    },
```

Selected Object and **Selected Value**

These properties are defined in the configuration file for outgoing data with the **selectedChildNode** function. For example scripts see Pass Data to and from Widget.

Select Node

This property is used to take an object with any number of property value pairs for comparison. Using this property, you can set a default value or persist a node selection in a hierarchy.

Note: The **On Selection Change** event is triggered when the same node is selected through this property. This is because before selecting a child node, its parent node is selected first, where null is omitted. After that, the child node that you want is selected. So on each API call, if there's a matching node, there are two **On Selection Change** events received.

API parameter:

```
{
childNode:{
property1:value1,
property2:value2,
:
}
}
```

Example

```
control.findByXmlNode("Widget1").widgetProperties.selectNode = JSON.stringify(
{
childNode:{
reasonGroupID : 15,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
reasonID: 20,
}
});
```

Validation scenarios

There are scenarios where the Hierarchy Selector widget doesn't work as expected, which includes the following:

- If a node selection cannot be found, an error is thrown in the console log. No error is shown in the UI and no error code is returned.
- If the **childNode** parameter isn't defined correctly, an error is thrown in the console log.

  For example:

  ```
  control.findByXmlNode("Widget1").widgetProperties.selectNode = JSON.stringify(
  {
  childNode123:{
  reasonGroupID : 15,
  reasonID: 20,
  }
  });
  ```

- If the **childNode** value is null or empty, or isn't an object (such as numeric value), an error is thrown in the console log.

  For example:

  ```
  control.findByXmlNode("Widget1").widgetProperties.selectNode = JSON.stringify(
  {
  childNode:null
  });
  or
  control.findByXmlNode("Widget1").widgetProperties.selectNode = JSON.stringify(
  {
  childNode:{ }
  });
  or
  control.findByXmlNode("Widget1").widgetProperties.selectNode = JSON.stringify(
  {
  childNode:5
  });
  ```

- Error without calling JSON stringify (UI error)

  For example:

  ```
  control.findByXmlNode("Widget1").widgetProperties.selectNode =
  {
  childNode:{
  reasonGroupID : 15,
  reasonID: 20,
  }
  };
  ```

Hierarchy Selector Events

The Hierarchy Selector widget supports the following events:

- On Initialize

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- On Selection Change

On Initialize

Use scripts for the **On Initialize** event of a Hierarchy Selector widget to accomplish a desired operation after the widget control is initialized, but before it is rendered to the UI.

Example

If by default you want to load the actual dataset from a database lookup once the widget is launched, add the script on the **On Initialize** event.

On Selection Change

Use scripts for the **onSelectionChange** event of a Hierarchy Selector widget to perform a particular operation when transitioning from select to deselect state.

This event is triggered when a child node is selected or deselected. A child node is deselected when a parent or group node is selected, or a breadcrumb trail is selected. When the same child node is selected multiple times continuously or there is a constant deselect state, this event is triggered only once.

When a child node is selected or deselected, the value is fetched from the widget's **Selected Object** and **Selected Value** properties through scripting. A stringify null value is passed through the **Selected Object** (selectedObj) and **Selected Value** (selectedValue) properties on a deselect state.

Example

In the following example the **selectedObj** and **selectedValue** widget properties are used to passed the value of the selected reason object and the different properties within the reason object into the specified controls.

```
if(control.findByXmlNode("Widget1").widgetProperties.selectedObj != null &&
control.findByXmlNode("Widget1").widgetProperties.selectedObj != '')
{
var x = JSON.parse(control.findByXmlNode("Widget1").widgetProperties.selectedObj);
var y = JSON.parse(control.findByXmlNode("Widget1").widgetProperties.selectedValue);
var v = null;
if(x != null)
{
for(var prop in x)
{
if(x.hasOwnProperty(prop))
{
v = v+ prop + ": "+x[prop] + "\n";
}
}
control.findByXmlNode("ReasonAttribute1").value = x.reas_cd;
control.findByXmlNode("MESReasonselectedGroupID").value = x.reas_grp_id;
control.findByXmlNode("ReasonAttribute2").value = x.reas_desc;
control.findByXmlNode("ReasonAttribute3").value = x.color;
}
else
{
control.findByXmlNode("ReasonAttribute1").value = "";
control.findByXmlNode("MESReasonselectedGroupID").value = "";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.findByXmlNode("ReasonAttribute2").value = "";
control.findByXmlNode("ReasonAttribute3").value = "";
}
var z = 'selected Value :'+y + "\n";
z += 'selected Obj :'+v;
control.findByXmlNode("selectedValueWidget2").value = '' + y;
control.findByXmlNode("ReturnwholeObjectvaluesWidget2").value = z;
}
```

Configure Hierarchy Selector Widget

Modify specific sections in the configuration file to meet your required widget settings.

Pass Data to and from Widget

The Hierarchy Selector uses the widget's **Data** property for incoming data and the **Selected Object** and **Selected Value** properties for outgoing data. The data must be in a JSON string (*stringify*) format to be recognized and passed to and from the widget.

To see the supported data structure and how **Data, Selected Object,** and **Selected Value** are used in the configuration file, see Hierarchy Selector Widget Properties Usage.

The following are examples of how the **Selected Object** and **Selected Value** properties are used for scripting to return an object (which is the selected child node) with all its properties and to return a specific object property, respectively. In both cases, the script is added to a property in a Work Tasks control.

In the following examples, the scripts are added to the **Value** property of a textbox control.

To retrieve the value returned for the **selectedObj** *property*

```
var temp = control.findByXmlNode("Widget1").widgetProperties.selectedObj;
var v ="";
if(temp != undefined && temp != "")
{
var reason = JSON.parse(temp);
if(reason != null)
{
for(var prop in reason)
{
if(reason.hasOwnProperty(prop))
{
//console.log(prop + ": "+reason[prop]);
v = v+ prop + ": "+reason[prop] + "\n";
}
}
}
}
return v;
```

To retrieve the value returned for the **selectedValue** *property*

```
var temp = control.findByXmlNode("Widget1").widgetProperties.selectedValue;
if(temp != undefined && temp != "")
{
return JSON.parse(temp);
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
else
{
return "";
}
```

The selected child node can either return an object or a specific object property. If it returns an object, the returned data is passed to the **Selected Object** property. If it is a specific object property, then the data is passed to the **Selected Value** property. You can define the specific object property under the **defaultNodesMapping** section of the configuration file.

If you want to access any of the properties of the returned object, the following is an example script that you can use, where it returns the value for the specified property or column field.

To retrieve the value of any property from the returned object

```
var temp = control.findByXmlNode("Widget1").widgetProperties.selectedObj;
if(temp != undefined && temp != "")
{
var reason = JSON.parse(temp);
if(reason != null)
{
return reason.reas_cd;
}
else
{
return "";
}
}
else
{
return "";
}
```

Load Hierarchical Dataset

Load Hierarchical Dataset

The Hierarchy Selector widget supports a tree data structure. If your dataset does not match this structure, you need to create your own utility helper to convert your current structure to the supported tree structure. To see the supported tree structure, see the test data section in the **Hierarchy_SampleData_conf.js** file**.**

The Hierarchy Selector supports an **On Initialize** event property for scripting. For example, if by default, you want to load the actual dataset from a database lookup once the widget is launched, add your script in **On Initialize.** But if you are using the **Hierarchy_SampleData_conf.js** file, the widget loads the sample data first, and then triggers the script in **On Initialize.**

During runtime, if you switch to a different configuration file, the widget triggers the script in the **On Initialize** event. If you switch to a configuration with sample data, the same behavior described applies—the widget loads the sample data first—and then calls the script in the **On Initialize** event.

To load the parent-child hierarchical dataset

1. In the corresponding widget configuration file, define the dataset that you want to load under the **defaultNodesMapping** section. The section looks similar to the following:

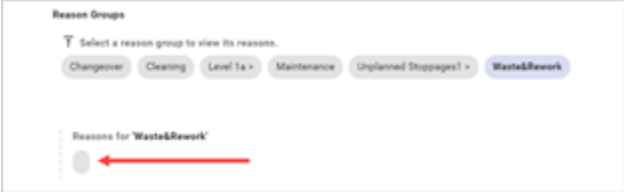
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// defaultNodesMapping is used to map property name use in incoming data.
// It will be used if nodesMapping is NOT supplied
defaultNodesMapping :
{
    "groupName": "",  // required
    "groupNodes": "", // required
    "showGroupAsEnabled": "", // optional, map to property name to show group
    "childNodes": "", // required
    "childName": "",  // required
    "childColor": "",  // optional, it can be null or empty string. If null or
    "selectedChildNodeValue": "" // optional, property name of return object.
},
```

2. Replace the current value for each of the properties listed with the ones that you have in your dataset. You can use the test data to see how this works. For info on using the test data, see Using the test data in Hierarchy Selector .

Things to take note of:

- If the **groupName** and **childName** properties under the **defaultNodesMapping** section are incorrectly mapped, their corresponding data doesn't get loaded, and you see an empty node image like in following screenshot. To resolve this issue, check your configuration file and fix the column mapping accordingly.



- The value that you specify for each of the listed properties is case sensitive. It must match the name from the dataset to load the data properly.
- The **showGroupAsEnabled** property allows you to map a property or a column name in a dataset to indicate if a group or parent node is selectable or not through an enable or disable visual cue.

  The value of the mapped column name or property is expected to be a boolean data type, where in, if the value is **False**, the parent or group node is shown as disabled and takes the color assigned to the **defaultParentDisabledColor** property in the configuration file. If the value is **True**, the node's color is kept as is, where it takes the color assigned to the **defaultParentNonSelectedColor** property.
- The **childColor** property supports a **null** value. If no color is specified, then the widget uses the default color of the selected child item**.** For information on the default color of the selected child node, see Set Color for Child Node.
- The following are the supported color codes. It also supports the actual color name. For example, **green**, **blue**, and so on.

	Color	Example
	Hex	#ff0000
	RGB	rgb(255, 0, 0)
	HSL	hsl(0, 100%, 50%)

If you provide an invalid color code or string, a color rendering issue is encountered. For example, **#Z34FF9**, **"#blue"**. To fix it, correct the invalid value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Override the Hierarchy Selector's Default Behavior for Selected Child Node

Override the Hierarchy Selector's Default Behavior for Selected Child Node

The Hierarchy Selector widget provides an option to override its default behavior on handling returned data (selected child node).

By default, selected child node is returned as a data object and passed through the widget's **Selected Object** property. The widget also provides an option to return a specific data object property through the function parameters defined in the configuration file. This specific object property is passed through the widget's **Selected Value** property.

To override the widget's default behavior

- Uncomment the **selectedChildNode** function section in the configuration file. If you are using the main configuration file (**Hierarchy_conf.js**), look for this section, and modify it accordingly based on the example code provided in the configuration file with sample data:

```
// uncomment selectedChildNode function to provide own customization. The function will be called when child node is selected/deselected.
// parameters:
//           selectedChildObject - selected child node object. Null when deselected.
//           selectedChildValue  - selected child node property value which property name is set in "selectedChildNodeValue" of defaultNodesMapp
// selectedChildNode : function(selectedChildObject,selectedChildValue)
// {
// }
```

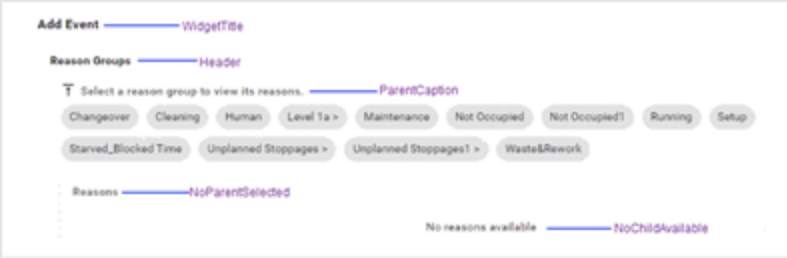
  The example code that overrides the widget's selected child node behavior is available in the **Hierarchy_SampleData_conf.js** file, as shown in the following screenshot.

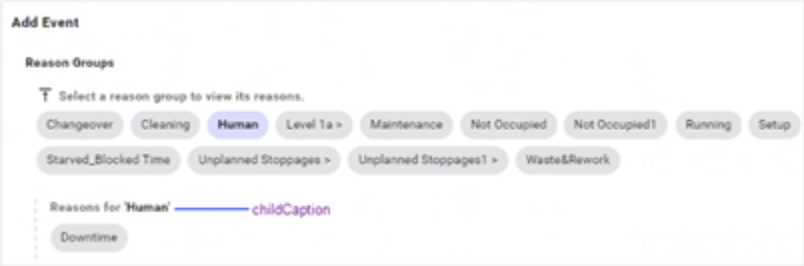
```
// uncomment selectedChildNode function to provide own customization. The function will be called when child node is select
// parameters:
//           selectedChildObject - selected child node object. Null when deselected.
//           selectedChildValue  - selected child node property value which property name is set in "selectedChildNodeValu
selectedChildNode : function(selectedChildObject,selectedChildValue)
{
    // Provide customization on routing extra selected property to value3 and value4
    // (value3 and value4 need to be added in widget.json)
    cwidget.selectedObj = JSON.stringify(selectedChildObject);
    cwidget.selectedValue = JSON.stringify(selectedChildValue);

    if(selectedChildObject != null && selectedChildObject.EquipmentState != null)
    {
      cwidget.value3 = JSON.stringify(selectedChildObject.EquipmentState.Name);
      cwidget.value4 = JSON.stringify(selectedChildObject.EquipmentState.Color);
    }
    else
    {
```

Modify Widget Title, Header, and Captions

The user interface label properties that you can configure in the Hierarchy Selector widget are marked in the following screenshots. Localizing these properties is also supported. For localization information, see Localize Widgets.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.





To modify the title, header, and captions in the Hierarchy Selector widget, go to the **//Title, header, and captions** section of the code in the configuration file and update the properties accordingly.

By default, the label properties are mapped to the values that you define in a text file (**Forms_AWT**<*VersionNumber*>**_HierarchySelector.txt)** used for localization as shown in the following screenshot. You can access the text file from <*AVEVAWorkTasksInstallPath*>**\Locales\en-US\Forms2\Standard**.

```
var conf={
    // Title, header, and captions
    widgetTitle : WW_localize("@@WidgetHSTitle@@"),
    header : WW_localize("@@WidgetHSheader@@"),
    parentCaption :WW_localize("@@WidgetHSParentCaption@@"),
    childCaption : WW_localize("@@WidgetHSChildCaption@@"),
    captionNoParentSelected : WW_localize("@@WidgetHSCaptionNoParentSelected@@"),
    widgetHSNoChildAvailable : WW_localize("@@WidgetHSNoChildAvailable@@"),
    noData : WW_localize("@@WidgetHSNoData@@"),
```

If you don't want to localize the label properties, simply enter the value that you want to appear in the UI for each of the label properties. You can either remove the "**WW_localize...**" value or comment it out as highlighted in the following screenshot.

```
40  var conf={
41      // Title, header, and captions
42      widgetTitle : "Title here", //WW_localize("@@WidgetHSTitle@@")
43      header : "Header",
44      parentCaption :WW_localize("@@WidgetHSParentCaption@@"),
45      childCaption : WW_localize("@@WidgetHSChildCaption@@"),
46      captionNoParentSelected : WW_localize("@@WidgetHSCaptionNoParentSelected@@"),
47      widgetHSNoChildAvailable : WW_localize("@@WidgetHSNoChildAvailable@@"),
48      noData : WW_localize("@@WidgetHSNoData@@"),
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Set Color for Parent Node

Keep or modify the assigned default color for the parent node when it is selected, initially loaded and a node is yet to be selected, or to show that it's disabled for selection.

```
// default disabled, select and non-select color for parent node
defaultParentDisabledColor: "#707070",
defaultParentNonSelectedColor : "#E0E0E0",
defaultParentSelectedColor : "#4D5EE03D",
```

Set Color for Child Node

Keep or modify the assigned default color for the child node when it is selected or on its initial display, where nothing is selected yet.

Note: The following default colors are used if the **childColor** property under the **nodesMapping** or **defaultNodesMapping** is not defined.

```
58          // default select and non-select color for child node if child color in nodesMapping or defaultNodesMapping is not defined
59          defaultChildNonSelectedColor : "#E0E0E0",
60          defaultChildSelectedColor : "#4D5EE03D",
```

Localize Widgets

Localize the widget to make it usable with a specific language, script, and culture. Localization for widget works the same way as it's done for AVEVA Work Tasks controls, where the key-value pairs are updated with **@@** prefix and suffix.

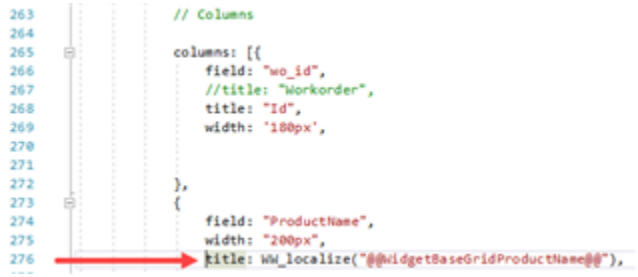
To localize the widget

1. In the corresponding configuration file, do the following:
   - (Hierarchy Selector) In the **//Title, header, and captions** section, keep the default and go to step 2. That section is already set to support localization by default. You only need to ensure that the value of the properties matches the ones in the localization text file.
   - (Grid) You are given the option to localize the Grid title and the different column headers in your dataset. By default, the Grid title is already set for localization, so you can just proceed to step 2. Ensure that the value of the property matches the one in the text file.

     For the column headers, use the **title** field to define the text to localize under the **//Columns** section of the code.

     For example:

     ```
     title: WW_localize("@@WidgetBaseGridProductName@@"),
     ```



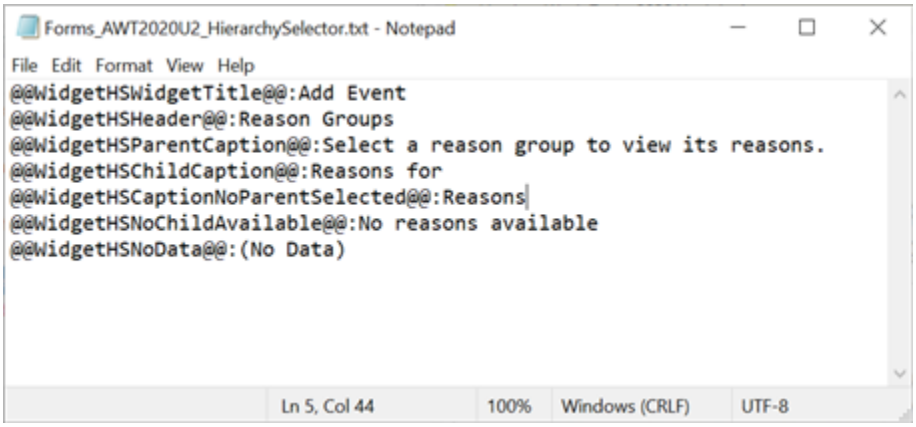
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

This must match the item in the localization text file. For example format of the text file, see Example localization text file.

2. Open each of the widgets' corresponding **wjson** file and check which properties defined in the said file you want to localize. For information on **wjson** file, see Dynamic Widget Properties and Events.
3. Open the corresponding widget text file for localization and define the values for each of the properties. You can access the widget text files from *<AVEVAWorkTasksInstallPath>***\Locales\en-US\Forms2\Standard.**
   - **Forms_AWT***<VersionNumber>***_BaseGrid.txt**
   - **Forms_AWT***<VersionNumber>***_HierarchySelector.txt**
4. To localize for cultures other than English (US), update the **web.config** file and create a separate localization text file for that language. For detailed steps on localizing for other cultures, see *Developer Guide* > *Localization* > *Steps to Localize AVEVA Work Tasks.*

Example localization text file

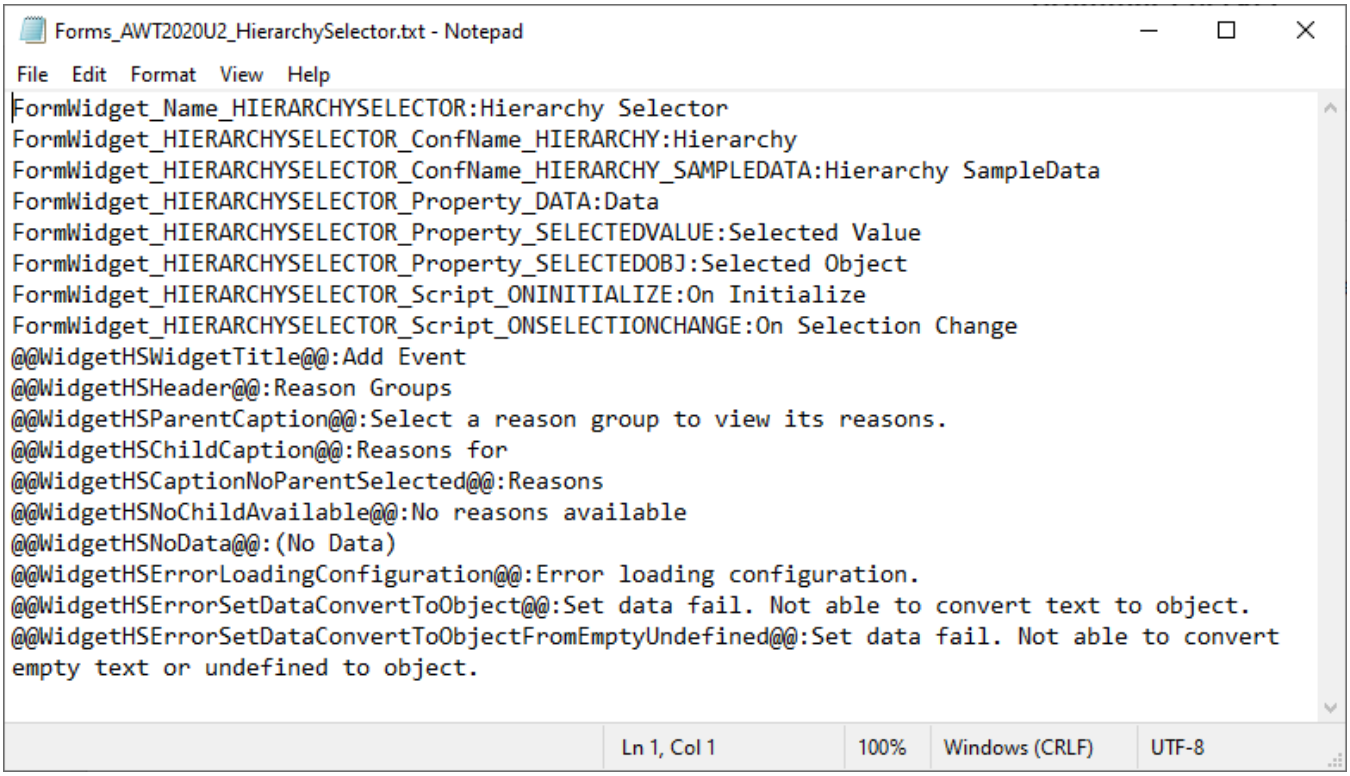
If you want to localize labels and column headers, you need to list them all in the text file like in the following example. The header or label enclosed with **@@** sign must also be defined in the configuration file as described in Localize Widgets.



The widgets are using a **wjson** file to define dynamic properties and events that appear in the AVEVA Work Tasks Properties page. The properties that you want to localize are then listed in the widget's text localization file. The following is an example of localization text file with properties from a **wjson** file.

Note: AVEVA Work Tasks requires some text conventions to follow when defining the key-value pairs in the localization text file. Pay attention to the text items in upper case format. If you're adding a new property or event in the **wjson** file, or a new configuration file, and you want it localized, follow the same format as in the following screenshot.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Configure Communication between Widget Control and Form

To configure communication between Widget control and a Form, you need to set the **Widget Type** as **Form** in the Widget control. Also, the **Form Details** property needs to be specified, which will be addressed as the Target Form in this topic. For information on specifying the Form Details property, see Widget.

Read more in below topics on how to establish the communication between the Widget control and the Target Form:

Initialize Communication

To initialize communication, you need to write the scripts in the Target Form. The below script can be used in the **On Form Load** script to initialize the communication context:

```
SFU.initializeWidgetFormCommunication(control);
```

To establish communicate between the Widget control and the Target Form, you need to define Form Parameters for the Target Form. So, when you configure the Widget control with **Widget Type** as **Form** and set the **Form Details** property to the Target Form, it would list the Form Parameters defined in the Target Form as the Widget properties. These Widget properties will be the communication points between the Widget control and the Target Form.

Note: Form Parameters are considered as **Widget Properties** for the Widget control irrespective of the data type. The **Widget Properties** are always represented as String data type.

Capture Values

To capture the values in the Target Form when the Widget control property is updated, you need to define the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

script as below (can be done for all the Widget properties, which needs to be captured):

```
ifc_msgHandler["FormParameterName"]= function(value){
// write scripts to update the UI with the value coming
// from the widget control to the form.
};
```

where, **FormParameterName** is the Form Parameter defined in the Target Form, which would be the Widget Property name in the Widget control.

So, the **On Form Load** script will look like as shown below, which includes the initialization part as well:

```
SFU.initializeWidgetFormCommunication(control);
ifc_msgHandler["FormParameterName"]= function(value){
// write scripts to update the UI with the value coming
// from the widget control to the form.
};
```

Send Values

To send values from the Target Form to the widget control, you need to define the script as below:

```
ifc_send("FormParameterName", "Value to be sent");
```

where, **FormParameterName** is the Form Parameter defined in the Target Form.

The above script can be written as required in the Target Form, like from an **On Click** script of a button control or an **On Data Change** script of a Textinput control.

Read URL Parameter Values

The Widget control has a property **Add Values as Url** and is available only when the **Widget Type** is set as **Form**. When **Add Values as Url** is set to **Yes**, then the Widget properties, which are the Form Parameters based properties, are passed as URL parameters to the iFrame control used internally to render the Form for the Widget control.

To read the URL Parameter values, use the below script in the Target Form:

```
var urlParameterValue = ifc_getFormParam("FormParameterName");
```

where, **FormParameterName** is the Form Parameter defined in the Target Form.

Custom Controls

Custom controls are controls designed with specific or custom properties. Custom controls make use of the advanced forms features and support jQuery and Knockout based JavaScript controls.

Custom controls can be added either to a custom category or to the **Controls** category in the **Forms Designer** toolbox.

A custom control is created and integrated with the forms as follows:

- Create a dynamic-link library (DLL) or an assembly where all the prerequisites for creating and registering a custom forms control is followed.
- Add the assembly to the global assembly cache (GAC) or place the assembly in a dedicated folder.
- If the assembly is placed in a dedicated folder, then register the location of the folder with the **SOA Folder List**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Register the assembly as a forms custom control through the **Forms Controls.**
- Use the new custom control in your forms from the toolbox in the **Forms Designer**.

Note:

- When you add a custom control as part of the Form, the reference to the path of the custom control is present in the Form. Hence, if you import the Form containing custom control in another environment, the Form expects that the custom control is present in the same path.

- For support in the AVEVA Work Tasks Pro app,
- update the CSS files in the path **../<BPMUITemplates>/Default/NextGenForms/custom/css**.
- update the js files in the path **../<BPMUITemplates>/Default/NextGenForms/custom/js**.
- For AVEVA Work Tasks Pro, mapping of themes is not considered.

Create Custom Controls

Create a custom control to design a control with specific or custom properties.

To create a custom control

1. Create a DLL project (**Visual C# > Windows > Class Library**) using Visual Studio.NET.
2. Create or add two .cs files – **SFControls.cs** (control initialization file) and **MyCustom.cs** (public control class file) to your project.
3. Add a reference to the **Skelta.Forms.Core.dll** (located at **<AVEVA Work Tasks Install Path>\Bin** folder) in your project.
4. Add the following namespaces to your **MyCustom.cs** file.

   ```
   using Skelta.Forms.Core;
   using Skelta.Forms.Core.External;
   using Skelta.Forms.Core.CommonObjects;
   using Skelta.Forms.Core.Controls;
   using Skelta.Forms.Core.Interfaces;
   ```

5. Ensure the control class (**MyCustom.cs**) inherits from the **BaseDataExternalControl** class.
6. Write the control definition code with the following in mind:
   a. Ensure the **Core Designer** is set to **Next Generation Designer** and the **Designer Category** is set to **Custom** to show up the control in the toolbox of the **Forms Designer**.

      ```
      [DesignerControl(992200, CoreDesigner.NextGenDesigner, DesignerCategory.Custom)]
      public class MyControl : BaseDataExternalControl
      ```

   b. Suppress properties, if required, using the attribute **SuppressedPropertiesNextGen**.

      ```
      // Suppress properites PropertyName1, PropertyName2, and PropertyName3.
      [SuppressedPropertiesNextGen("PropertyName1;PropertyName2;PropertyName3;")]
      ```

   c. Override the **GetNewInstanceForClone()** function, create an instance of the control, and assign the basic properties of the control.
   d. Override the **GetControlHtml()** function and define the HTML code. Use the **TopLevelForm.InDesignMode** boolean variable to segregate the designer and run-time views, if required. Ensure all view model properties are specific to the instance defined in the view.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

e. Override the **GetViewModelDefinition()** function (if required) to define the **ViewModel** definition of the Knockout binding based on the custom control Binding Attribute (For more information, see *Binding Attribute* in Attributing Properties). Ensure all view model properties are specific to the instance defined in the view model. For more information, see Autocomplete.cs.

f. Override the **GetViewModelScript()** function (if required) to define the **ViewModel** script based on the predefined named script like **Value**, **Validate**, **Mandatory** and so on.

g. Ensure you add the JS file reference to **this.TopLevelForm.RuntimeJS** variable to add custom JS files referred by the custom control. The variable is a key value pair collection where the key is the file name and the value is the file path (absolute or relative). Also ensure the JS files are present in the **BPMUITemplates/Default/NextGenForms/custom/js** folder. Otherwise, it will not be rendered.

Note: For the earlier versions of the product, ensure the JS files are present in the **BPMUITemplates/Default/NextGenForms/js** folder.

h. Ensure you add the CSS file reference to **this.TopLevelForm.RuntimeCss** variable to add custom CSS files referred by the custom control. The variable is a key value pair collection where the key is the file name and the value is the file path (absolute or relative). Also ensure the CSS files are present in the **BPMUITemplates/Default/NextGenForms/custom/css** folder. Otherwise, it will not be rendered.

Note: For the earlier versions of the product, ensure the CSS files are present in the **BPMUITemplates/Default/NextGenForms/css** folder.

i. Define rules for a custom control to validate the data using the client-side JavaScript, if required. For more information, see Rules and Defining Rules.

j. Create class properties as control properties.

```
//Designer Property
[DesignerProperty(773000, Categories.Validation, CoreDesigner.NextGenDesigner)]
//Binding Property
[BindingAttribute(882995, "Skelta.Forms.Core.CommonObjects.IntegerBindingProperty")]
```

7. Create **SFControls.cs** class under **Skelta.Forms.ExternalControls** namespace.
8. Add the following namespaces to your **SFControls.cs** file.

```
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.External;
using MyCustomControls; // Control namespace
```

9. Implement the **IExternalControls** in the **SFControls** class.
   a. In the **IExternalControls.GetControl(string controlFullName)** method, return the new control instance based on the control namespace.
   b. In the **GetControls** method, return a new object array which contains all the control instances.
10. Build your project, create a **SOA** folder on your hard drive (for example, **C:\SOA**), place your custom control DLL file in the folder, and then register the location of the folder with the **Skelta SOA Folder List**.
11. Register the custom control DLL file through the **Form Controls** to display the custom control in the controls or custom category. For more information, see Custom Categories.

Note:
- Control initialization file must always be named as **SFControls.cs** and the class name must always be named as **SFControls**.
- Ensure reference to the **Skelta.Forms.Core.dll** assembly is the same as the assembly on the AVEVA Work Tasks server.
- For information about SOA, see *Object Access* in the *Workflow Management User Guide*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To localize the text for the custom control

- See Localization.

To add an icon to a custom control

1. Add a custom icon in the **<AVEVA Work Tasks Install Path>/Web/BPMUITemplates/Default/NextGenForms/Themes/<ThemeName>/images** folder.

   Note:
   - If the image name is tagged with **FormCustomPath:**, the location is **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/images/MyImage.png**

   - If the image name is tagged with or **FormCustomThemePath:**, the location is **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/images/MyImage.png**

   - If the image name is not tagged with **FormCustomPath:** or **FormCustomThemePath:**, the location of image of earlier version will be considered i.e. **<Installpath>/Web/BPMUITemplates/Default/Themes/<ThemeName>/ NextGenForms/images** folder.

2. Open the **SkeltaCustomControlDesigner.css** file in the **<AVEVA Work Tasks Install Path>/Web/BPMUITemplates/Default/NextGenForms/Themes/<ThemeName>/custom\css** folder.

   Note: If the stylesheet name is not tagged with **FormCustomPath:** or **FormCustomThemePath:**, the location of stylesheet of earlier version will be considered.

1. Add the following style to the **SkeltaCustomControlDesigner.css** file.

```
/* Replace <custom_icon> with the file name for the normal icon. */
/* Replace <namespace.classname> with the custom control namespace and class name. */
.sktbci[data-skttp="<namespace.classname>"]
{
background-image: url(../../images/<custom_icon>.png);
}
/* Replace <custom_icon_hover> with the file name for the hover icon. */
.sktbc:hover > .sktbci[data-skttp="<namespace.classname>"]
{
background-image: url(../../images/<custom_icon_hover>.png);
}
```

2. Save the **SkeltaCustomControlDesigner.css** file.
3. Reload the **Forms Designer** page to see the custom control with the custom icon in the toolbox.

Extend Existing Controls

Create a custom control by inheriting and extending from an existing forms control to leverage the features of the existing control.

When a custom control class inherits from one of the existing forms control class and the **ICustomControl** interface, the custom control has access to all the features of the existing forms control. You can extend the forms control by overriding the existing properties and functionalities of the forms control class.

Use **GetViewModelDefinition()** function to make use of the existing forms control properties by assigning values

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

to the existing forms control property using the **base** keyword. Override **GetControlTypeName** property and return the base type of the control to inherit all features of the existing control.

Example: Inheriting all features of Text Control

Custom control inheriting all features from the Text control class:

```
public override string GetControlTypeName
{
get
{
return typeof(TextInput).Name;
}
}
```

Example: Inheriting Text Control

Custom control inheriting from the Text control class:

```
[DesignerControl(992200, CoreDesigner.NextGenDesigner, DesignerCategory.Custom)]
public class FirstName: TextInput, ICustomControl
```

Example: Extending ViewModelDefinition() and ViewModelScript() functions

```
public override string GetViewModelDefinition()
{
 base.MaxLength = 10;
 base.MinLength = 4;
 return base.GetViewModelDefinition();
}
public override string GetViewModelScript()
{
 string code = string.Empty;
 var valueScript = this.PropertyExpressions.Where(e => e.Name == "Value").ToList();
 if (valueScript.Count == 0)
 {
 string retVal = "Test Value";
 code = "if(currentValue==\"" + string.Empty + "\"" + "){ return \"" + retVal + "\"" + ";}
else{ return currentValue; }";
 this.PropertyExpressions.Add(new Property("Value", code));
 }
 var validateScript = this.PropertyExpressions.Where(e => e.Name == "Validate").ToList();
 if (validateScript.Count == 0)
 {
 string comma = ",";
 string validationmsg = "Comma (,) is not a valid character";
 code = "if(currentValue.indexOf(\"" + comma + "\"" + ") > -1){ return new
ValidationOptions(false, \"" + validationmsg + "\"" + ");} else { return new
ValidationOptions(true, \"" + string.Empty + "\"" + "); }";
 this.PropertyExpressions.Add(new Property("Validate", code));
 }
 return base.GetViewModelScript();
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Custom Properties

Custom properties are properties designed with specific or custom attributes.

Attributing Properties

Attributing properties is defining a custom property of a custom control to render the property in the **Property** pages and the forms **ViewModel**.

Attributing properties includes defining:

Designer Property

Designer property determines the location of a custom property in the **Property** page of a custom control.

```
[DesignerProperty(int index, Categories categoryType, CoreDesigner designerType)]
```

where **index** is the property index, **categoryType** is the property tab, and **designerType** is the forms type.

Example

```
// Setting the location of the custom property
// in the Behaviour tab of the Property page.
[DesignerProperty(773000, Categories.Behaviour, CoreDesigner.NextGenDesigner)]
```

OR

```
[DesignerProperty(int index, Categories categoryType, Type controlProperty, string
propertyDetails)]
```

where **index** is the property index, **categoryType** is the property tab, **controlProperty** is the property type for the IControlProperty, and **string** is the property details. The **string** is an associated property name for the form XML definition. To generate the string, create a form with the required controls configured with the required properties and then export the form.

Note: We do not recommend using lookups and data grid controls as the forms will be dependent on the repository.

Example

```
// Setting the location of the custom property
// in the Behaviour tab of the Property page.
public string DataFormXml
{
get
{
// Code to return the form XML definition
}
}
[DesignerProperty(773000, Categories.Behaviour, typeof(GridProperty), "DataFormXml")]
```

Binding Attribute

Binding attribute determines how a custom property is rendered in the client-side JavaScript **ViewModel**.

```
[BindingAttribute(PropertyIndex, "<Namespace>.<ClassName>", BindingPropertyName =
"<ViewModelPropertyName>, IsObservable = <TrueFalse>")]
```

Example

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// Custom binding is defined under MyIntegerBindingProperty class
// located at MyNamespace.MyIntegerBindingProperty class path.
[BindingAttribute(882995, true, "MyNamespace.MyIntegerBindingProperty")]
```

Note:
- New index should be defined for each custom property.
- Index value for a custom property should be higher than the defined property index values.
- Properties for custom controls are cached in IIS. Therefore, we recommend to reset the IIS to clear the IIS cache when new properties are added or existing properties are changed.

Property Index

Property index determines the sequence in which a property is rendered on the **Forms Designer Property** page and on the forms at run-time.

Common Properties

Property	Index
ID	881000
Description	881100
Name	881200
Visible	881300
Retain Space	881400
Enable	881500
Read-Only	881600
Mandatory	881700
Lost Focus	881800
Height	881900
Width	882000
Tooltip	882100
Value	882200
Default Value	882300
Control Binding	882400

List Properties

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Index
Default Value	882500
Options	882600
Options Text	882700
Options Value	882800
Control Binding	882900

Button Properties

Property	Index
Button Text	882500
Button Type	882600
Value	882700

Event Lookup Properties

Property	Index
Control Binding	882500

Radio Button Properties

Property	Index
Default Value	882500
Control Binding	882600
Options	882700
Options Text	882800
Options Value	882900
Options Per Row	883000

Text Properties

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Index
Persist Password	882500
Encryption Mode	882600
Pattern	882700

Drop-Down Properties

Property	Index
Default Value	882500
Options	882600
Default Text	882700
Options Text	882800
Options Value	882900

Base Form Properties

Property	Index
ID	881010
Control Binding	882500
Behavior	88260

Create Custom Properties

Create a custom property to define a new property.

To create a custom property

1. Create a class property with a getter and setter section along with the user defined logic.
2. Decorate the class property with the **DesignerProperty** property attribution.
   (For more information, see *Designer Property* in Attributing Properties.)

```
// Setting the location of the custom property
// in the Behaviour tab of the Property page.
[DesignerProperty(773000, Categories.Behaviour, CoreDesigner.NextGenDesigner)]
OR
[DesignerProperty(773000, Categories.Behaviour, typeof(GridProperty), "DataFormXml")]
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. If you want the custom property to be a part of the View Model, then you must decorate the class property with the **BindingAttribute** property attribution.

   ```
   / Custom binding is defined under IntegerBindingProperty class
   // located at Skelta.Forms.Core.CommonObjects.IntegerBindingProperty class path.
   [BindingAttribute(882995, "Skelta.Forms.Core.CommonObjects.IntegerBindingProperty")]
   ```

4. Provide a **default value** for the custom property, if required, by decorating the class property with the DefaultValue property attribution.

Example

```
private string _step = "0";
/// <summary>
/// Get/set the step value for the slider.
/// </summary>
[DefaultValue("1")]
[DesignerProperty(775000, Categories.Behaviour, CoreDesigner.NextGenDesigner)]
[BindingAttribute(882998, EFormMode.NextGenMode,
"Skelta.Forms.Core.CommonObjects.IntegerBindingProperty")]
public string Step
{
 get { return _step; }
 set
 {
 _step = value;
 }
}
```

IControlProperty

IControlProperty is the interface for the control property.

Definition of IControlProperty

```
public interface IControlProperty
{
BaseControl GetPropertyControl(BaseControl control, PropertyInfo propertyInformation);
void ReadProperty(PropertyInfo propertyInformation, IXPathNavigable xmlDocument, object
propertyValue);
void WriteProperty(IXPathNavigable xmlDocument, BaseDataControl propertyControl,
BaseControl control);
}
```

GetPropertyControl()

This method renders the property for the custom control.

Syntax

```
BaseControl GetPropertyControl(BaseControl control, PropertyInfo propertyInformation);
```

Parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
control	Control for which the property is being rendered.
propertyInformation	Information about the property.

Return Value

This method returns the control.

ReadProperty()

This method reads the property value and updates the instance XML for the custom control.

Syntax

```
void ReadProperty(PropertyInfo propertyInformation, IXPathNavigable xmlDocument, object propertyValue);
```

Parameters

Parameter	Description
propertyInformation	Information about the property.
xmlDocument	Form instance XML.
propertyValue	Value of the property.

Return Value

This method returns the control.

WriteProperty()

This method writes the property value for the custom control.

Syntax

```
void WriteProperty(IXPathNavigable xmlDocument, BaseDataControl propertyControl, BaseControl control);
```

Parameters

Parameter	Description
xmlDocument	Form instance XML.
propertyControl	Property of the control.
control	Control for which the property is being rendered.

Return Value

This method returns the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Custom Bindings

Custom bindings are specific knockout bindings for the **ViewModel** at run-time. Custom bindings override the Knockout binding based on the custom control Binding Attribute (For more information, see *Binding Attribute* in Attributing Properties).

To create custom bindings

1. Create a custom binding class by inheriting the IBindingProperty.
2. Add the following namespaces to your custom binding property class file.

   ```
   using Skelta.Forms.Core;
   using Skelta.Forms.Core.Classes;
   using Skelta.Forms.Core.CommonObjects;
   using Skelta.Forms.Core.Controls;
   using Skelta.Forms.Core.Interfaces;
   ```

3. Implement RenderBinding function of the IBindingProperty interface.

   ```
   // For String, Number, Boolean
   <ContainerName>.<ControlID>.<KnockoutPropertyName> = ko.observable();
   <ContainerName>.<ControlID>.<KnockoutPropertyName> = ko.observable(<PropertyValue>);

   // For Array
   <ContainerName>.<ControlID>.<KnockoutPropertyName> = ko.observableArray();
   <ContainerName>.<ControlID>.<KnockoutPropertyName> =
   ko.observableArray(<ArrayPropertyVariable>);
   // Container Name
   bindingHelperObject.ViewModelContainerName
   // Control ID
   bindingHelperObject.Control.Id
   ```

4. Return the Knockout binding code snippet as a string.
5. Add the following function in the custom control class.

   ```
   public override IBindingProperty GetBindingClassObject(string fullyQualifiedClassPath)
   {
   try
   {
   return
   (IBindingProperty)Activator.CreateInstance(Type.GetType(fullyQualifiedClassPath));
   }
   catch (Exception ex)
   {
   return null;
   }
   }
   ```

Example

Custom binding for items, series, and colors properties of the Knockout Kendo Bar Chart control. The items, series, and colors are required arrays added with a third party JavaScript file.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
class BarchartBindingProperty : IBindingProperty
{
/// <summary>
/// The Value Property Implementation.
/// </summary>
/// <param name="bindingHelper">binding Helper object</param>
/// <returns>The implementation of Value Property</returns>
public string RenderBinding(object bindingHelper)
{
StringBuilder stringBuilder = new StringBuilder();
BindingHelper bindingHelperObject = (BindingHelper)bindingHelper;
var chartControl = bindingHelperObject.ViewModelContainerName + "." +
bindingHelperObject.Control.Id;
//// KO binding where items and series variables will get the array data from the external
source (3rd Party JS file).
 stringBuilder.AppendLine(bindingHelperObject.ViewModelContainerName + "." +
bindingHelperObject.Control.Id + "._items = ko.observableArray();");
stringBuilder.AppendLine(bindingHelperObject.ViewModelContainerName + "." +
bindingHelperObject.Control.Id + "._series = ko.observableArray();");
stringBuilder.AppendLine(bindingHelperObject.ViewModelContainerName + "." +
bindingHelperObject.Control.Id + "._colors = ko.observableArray();");
//// Items
stringBuilder.AppendLine("if(" + chartControl + "._itemsURL !== \"\") {");
stringBuilder.AppendLine(chartControl + "._items(getItems(" + chartControl +
"._itemsURL));");
stringBuilder.AppendLine("}");
stringBuilder.AppendLine("else {");
stringBuilder.AppendLine(chartControl + "._items(eval(" + chartControl + "._itemsData));");
stringBuilder.AppendLine("}");
//// Series
stringBuilder.AppendLine("if(" + chartControl + "._seriesURL !== \"\") {");
stringBuilder.AppendLine(chartControl + "._series(getSeries(" + chartControl +
"._seriesURL));");
stringBuilder.AppendLine("}");
stringBuilder.AppendLine("else {");
 stringBuilder.AppendLine(chartControl + "._series(eval(" + chartControl +
"._seriesData));");
stringBuilder.AppendLine("}");
//// Colors
stringBuilder.AppendLine("if(" + chartControl + "._colorsURL !== \"\") {");
stringBuilder.AppendLine(chartControl + "._colors(getColors(" + chartControl +
"._colorsURL));");
stringBuilder.AppendLine("}");
stringBuilder.AppendLine("else {");
stringBuilder.AppendLine(chartControl + "._colors(eval(" + chartControl +
"._colorsData));");
stringBuilder.AppendLine("}");
return stringBuilder.ToString();
}
}
```

Note:
Assignment of KO observable variables must be a function.

```
control.<property(value)>;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For example:

```
chartControl.items(<JSONCollection>);
```

IBindingProperty

IBindingProperty is the interface for the custom control.

Definition of IBindingProperty

```
public interface IBindingProperty
{
string RenderBinding(object bindingHelper);
}
```

Rules

Rules validate the value of a control.

Rule Structure

Object	Description
RuleFor	Associated property for the rule.
RuleName	JavaScript function name for the rule. This name is used in the client-side JavaScript and this JavaScript function is called when the rule is executed.
RuleDefinition	Error message for the rule.
ControlType	Associated control for the rule.
RenderMode	Render the control in HTML5 or HTML.

Lost Focus Rules

Lost focus rules trigger when a control looses focus or when a form is submitted.

Example

```
rules.Add(new Rule { RuleFor = "lostFocus", RuleName = "RuleName", RuleDefinition = "Error
Message", ControlType = this.GetType().Name, RenderMode = RenderingMode.Html5.ToString()
});
```

Value Rules

Value rules are triggered when the value of a control changes or when a form is submitted.

Example

```
rules.Add(new Rule { RuleFor = "Value", RuleName = "RuleName", RuleDefinition = "Error
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
message", ControlType = this.GetType().Name, RenderMode = RenderingMode.Html5.ToString()
});
```

Note:
- **RuleName** defined in the control class file is used as is in the client-side JavaScript function.
- When a rule in the **Rules** collection fails, further rules are not executed.

Define Rules

Define rules for a custom control to validate the data using the client-side JavaScript.

To define custom control rules

1. Override the **Rules** collection property in the custom control class.
2. Define the rule name, rule definition, control type, and render mode, if required.
3. At the client-side custom control JavaScript file, implement a JavaScript function with the following in mind:
4. Use the same name for the JavaScript function as that of the rule name defined in the custom control class.
5. Use the **Control**, **ControlValue** and **ruleObject** for parameters.
6. Verify the **ControlValue** parameter before taking any action.
7. Get the rule definition message using **ruleObject.ErrorMessage** property.

Example: Custom Control Class

```
/// <summary>
/// Rules for the control
/// </summary>
public override Collection<Rule> Rules
{
get
{
Collection<Rule> rules = base.Rules;
string ruleName = "MinMaxNumberValidation";
string errorMessage = "$$Name$$ value is out of range.";
rules.Insert(0, new Rule { RuleFor = "Value", RuleName = ruleName, RuleDefinition =
errorMessage,
ControlType = this.GetType().Name, RenderMode = this.RenderAs.ToString() });
return rules;
}
}
```

Example: Custom Control Client-side JavaScript

```
function minMaxNumberValidation(controlValue, control, ruleObject)
{
if (control.renderMode === "NonHtml5")
{
var errorMessage = "";
var errorBoolValue = false;
var errorCode = errorConstants.none;
var value = parseFloat(controlValue);
if (!isNaN(value))
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
if ((value >= control.min && value <= control.max) == false)
{
errorBoolValue = true;
errorCode = errorConstants.valueValidationRules;
control.validationError(errorBoolValue);
control.validationErrorMessage(Skelta.localize.getString(ruleObject.errorMessage).replace("
$$Name$$", control.xmlNodeBoundTo.toString()));
control.validationErrorCode(errorCode);
}
}}
}
```

Reference External CSS and JS Files

Reference external CSS and JS files dynamically by adding those files in the **RuntimeCSS** and **RuntimeJS** collections while returning the **GetControlHTML()** function to render custom controls in the **Forms Designer** page and at run-time of the forms.

Example

```
private void addJsCssFiles()
 {
//// <link rel="stylesheet" href="css/jquery-ui.css" />
 //// <script src="js/jquery-ui.js" type="text/javascript"></script>
 //// <script src="js/knockout-jquery-ui-widget.js" type="text/javascript"></script>
 //// <script src="js/autocomplete-datasource.js" type="text/javascript"></script>
 if (string.IsNullOrEmpty (this.TopLevelForm.RuntimeCss["css1"]))
 {
 this.TopLevelForm.RuntimeCss.add("css1", "css/jquery-ui.css");
 }
 if (string.IsNullOrEmpty (this.TopLevelForm.RuntimeJS["js1"]))
 {
 this.TopLevelForm.RuntimeJS.add("js1", "js/jquery-ui.js");
 }
 if (string.IsNullOrEmpty (this.TopLevelForm.RuntimeJS["js2"]))
 {
 this.TopLevelForm.RuntimeJS.add("js2", "js/knockout-jquery-ui-widget.js");
 }
 if (string.IsNullOrEmpty (this.TopLevelForm.RuntimeJS["js3"]))
 {
 this.TopLevelForm.RuntimeJS.add("js3", "js/autocomplete-datasource.js");
 }
 }
```

Note:
- CSS file should be placed in the **<AVEVA Work Tasks Install Path>/Web/BPMUITemplates/Default/NextGenForms/custom/css/** folder.
- JS file should be placed in the **<AVEVA Work Tasks Install Path>/Web/BPMUITemplates/Default/NextGenForms/custom/js** folder.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Localization

Localize a custom control name, property, and description to make it usable with a specific language, script, and culture.

To localize a custom control name in Forms Designer Toolbox

- Add a new key-value pair in a custom text file and place the file in the **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2** folder in the following format: **FormNGFControl<namespace.classname>:localized Name**.

  For example, to localize the **Masked Input Control** control name, code as follows:

  ```
  FormNGFControlFormCustomControl.MaskedInputControl:Masked Input
  ```

Note: Replace **namespace.classname** with the custom control namespace and class name.

To localize a custom property name and description

- Add a new key-value pair in a custom text file and place the file in the **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2** folder in the following format: **Form<property name>Property:localized Name** and **Form<property name>PropertyDescription:localized Description**.

  For example, to localize the name and description of the **Retain Selected Value** property, code as:

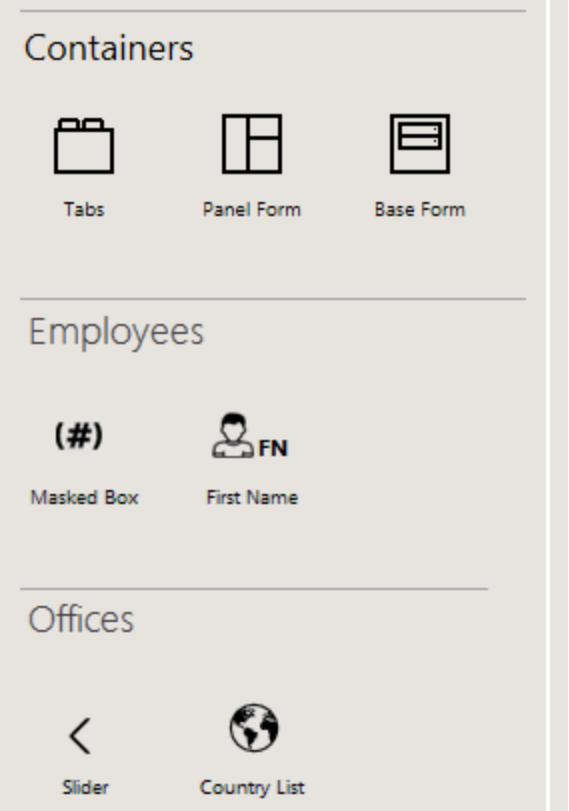
  ```
  FormRetainSelectedValueProperty:Retain Selected Value
  ```

  ```
  FormRetainSelectedValuePropertyDescription:Keep selected value regardless of the current parameter values.
  ```

Custom Categories

Custom categories organize custom controls in the **Forms Designer** toolbox.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Custom categories are displayed in the **Forms Designer** toolbox after the default **Controls** and **Containers** category.

Create Custom Categories

Create a custom category to organize custom controls in the **Forms Designer** toolbox.

To create a custom category in the Forms Designer toolbox

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Form Controls**. The **Form Controls** page appears.
2. On the action bar, click **New**, and then click **Folder**. The **New Folder** wizard appears.
3. In the **Title** box, type a name for the folder, and in the **Description** box, type a description for the folder.
4. Click **Save & Close**. The folder is created, and an information message appears.
5. Click **OK** (or click the **Close** button). The information message and the **New Folder** wizard are closed.

Note:
The name of the folder created in the **Form Controls** page is used for the name of the category in the **Forms Designer** toolbox.
- Custom category is displayed in the **Forms Designer** toolbox only if a custom control is added to the folder.
- Custom controls can be added either to one of the custom categories or to the **Controls** category in the **Forms Designer** toolbox.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Add Custom Controls in Custom Categories

Add a custom control in a custom category of the **Forms Designer** toolbox to organize custom controls in the **Forms Designer** toolbox.

To add a custom control in a custom category of the Forms Designer toolbox

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Form Controls**. The **Form Controls** page appears.
2. Right-click on a folder, and then click **Show Child Item** (or double-click the folder). The child items page for the folder appears.
3. On the action bar, click **New**, and then click **Form Control**. The **New Form Control** wizard appears.
4. In the **Name of the control** box, type a name for the control, and in the **Description** box, type a description for the control.
5. Click **Save & Continue**. The control is created, and the **Form Custom Control** wizard appears.
6. Do one of the following:

Task	Action
• Select the custom control assembly created in the global assembly cache (GAC).	a. Click **GAC**. A list of assembly files appear. b. Click the assembly file for the custom control, and then click **Save**. The custom control is added to the custom category of the **Forms Designer** toolbox.
• Select the custom control assembly created in the dedicated folder.	a. Click **Folder**. A list of assembly files appear. b. Click the assembly file for the custom control, and then click **Save**. The custom control is added to the custom category of the **Forms Designer** toolbox.

Note: If a Custom Form Control is consumed inside a form, you cannot edit or delete the Custom Form Control. To modify or delete the Custom Form Control, ensure it is not used in any form.

Add Custom Controls in the Controls Category

Add a custom control in the Controls category of the Forms Designer toolbox to make it easily available in the toolbox.

To add a custom control in the Controls category of the Forms Designer toolbox

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Form Controls**. The **Form Controls** page appears.
2. On the action bar, click **New**, and then click **Form Control**. The **New Form Control** wizard appears.
3. In the **Name of the control** box, type a name for the control, and in the **Description** box, type a description for the control.
4. Click **Save & Continue**. The control is created, and the **Form Custom Control** wizard appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Do one of the following:

Task	Action
• Select the custom control assembly created in the global assembly cache (GAC).	a. Click **GAC**. A list of assembly files appear. b. Click the assembly file for the custom control, and then click **Save**. The custom control is added to the Controls category of the **Forms Designer** toolbox.
• Select the custom control assembly created in the dedicated folder.	a. Click **Folder**. A list of assembly files appear. b. Click the assembly file for the custom control, and then click **Save**. The custom control is added to the Controls category of the **Forms Designer** toolbox.

Note: Custom control added to the **Controls** category is displayed after the last control in the **Controls** category.

Organize Form Controls

Organize a form control by performing one of the following:

- Moving the form control to a folder.
- Making the form control a child item of another form control.

To organize a form control by moving the form control to a folder

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Form Controls**. The **Form Controls** window appears.
2. On the action bar, click **Organize**. The **Organize Form Controls** dialog box appears.
3. Drag a form control to a folder. The form control is moved to the folder.
4. Click **Close**. The **Organize Form Controls** dialog box is closed.

To organize a form control by making the form control a child item of another form control

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Form Controls**. The **Form Controls** window appears.
2. On the action bar, click **Organize**. The **Organize Form Controls** dialog box appears.
3. Drag a form control to another form control. The form control is now a child item of another form control.
4. Click **Close**. The **Organize Form Controls** dialog box is closed.

Example: Masked Input Control

A custom Masked Input control with control initialization, class, and JavaScript files.

- SFControl.cs (control initialization file)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- MaskedInput.cs (control class file)
- skelta.maskedinput.js (JavaScript file)
- Adding Masked Input Property for Check Syntax Validation

SFControl.cs

```
using CustomControl;
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.External;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Skelta.Forms.ExternalControls
{
class SFControls : IExternalControls
{
#region IExternalControls Members
BaseControl IExternalControls.GetControl(string controlFullName)
{
switch (controlFullName)
{
case "CustomControl.MyCustom":
return new MyCustom();
default:
return null;
}
}
object[] IExternalControls.GetControls()
{
return new object[] { new MyCustom() };
}
#endregion
}
}
```

MaskedInput.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Skelta.Forms.Core;
using Skelta.Forms.Core.External;
using Skelta.Forms.Core.CommonObjects;
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.Interfaces;
using System.ComponentModel;
using System.Collections.ObjectModel;
namespace CustomControl
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// My Custom control class.
// This custom control will act as a Masked Input control.
// Add attributes to make it available in the Forms Designer toolbox.
[DesignerControl(992200, CoreDesigner.NextGenDesigner, DesignerCategory.Custom)]
public class MyCustom : BaseDataExternalControl
{
public override BaseControl GetNewInstanceForClone()
{
var custom = new MyCustom();
custom.Mask = this.Mask;
return custom;
}
// Mask field.
private string mask = "###-##-###";
// Mask property.
// Add Default Value for the property.
 [DefaultValue("###-##-###")]

 // Add the Designer attribute to render this property in the Properties page, under the
Validation tab.
[DesignerProperty(774000, Categories.Validation, CoreDesigner.NextGenDesigner)]
public string Mask
{
get { return mask; }
set { mask = value; }
}
/// <summary>
/// Render the control - Look and feel of the control in the Forms Designer and Preview.
/// </summary>
 /// <returns>Controls View</returns>
public override string GetControlHtml()
{
var stringBuilder = new StringBuilder();
StringBuilder dataBindAttribute = new StringBuilder();

 // Create binding for the controls with the "DataBindAttribute" keyword. By default all
the below properties are rendered.
 // You can also add other Knockout binding like "attr" here.
// For a separate view for the Design mode using the "this.TopLevelForm.InDesignMode"
keyword.
if (this.TopLevelForm.InDesignMode == false)
{
dataBindAttribute.Append(" " + DataBindAttribute + "=\"value: " + this.Id + "._value");
dataBindAttribute.Append(", enable: " + this.Id + "._enable");
dataBindAttribute.Append(", readOnly: " + this.Id + "._readOnly");
dataBindAttribute.Append(", mandatory: " + this.Id + "._isMandatory");
dataBindAttribute.Append(", hasfocus:" + this.Id + "._lostFocus");
dataBindAttribute.Append("\"");
}
// Render the control.
stringBuilder.Append("<input type='text' " + this.GetControlIdAttribute +
dataBindAttribute.ToString() + " onkeypress='return
Skelta.forms.maskedInput.keyPress(this);'");
if (this.Mask.Trim().Length > 0)
{
stringBuilder.Append(" placeholder=\"" + this.Mask + "\"");
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
// End the Input tag.
stringBuilder.Append(" />");
// Function to add the custom JS files.
AddJsFiles();
// Function to add the custom CSS files.
AddCssFiles();
return stringBuilder.ToString();
}
/// <summary>
/// Add CSS files associated with the custom control.
/// </summary>
private void AddCssFiles()
{
/* As this custom control has no specific CSS, no custom CSS file is added. */
////if (string.IsNullOrEmpty (this.TopLevelForm.RuntimeCss["MyCSSFile"]))
////{
////  // "this.TopLevelForm.RuntimeCss" is the collection which holds your custom CSS
files.
////  // It is a key-value pair collection where the key is the file name and the value is
the file path(absolute or relative).
////  // Ensure the CSS files are present in the "BPMUITemplates/Default/NextGenForms/css"
folder. Otherwise, it will not be rendered.
////  this.TopLevelForm.RuntimeCss.Add("MyCSSFile", "css/maskedinput.css");
////}
}
/// <summary>
/// Add custom JS files associated with the custom control.
/// </summary>
private void AddJsFiles()
{
if (string.IsNullOrEmpty (this.TopLevelForm.RuntimeJS["MyJsFile"]))
{
// "this.TopLevelForm.RuntimeJS" is the collection which holds your custom JS files.
// It is a key-value pair collection where the key is the file name and the value is the
file path(absolute or relative).
// Ensure the JS files are present in the "BPMUITemplates\Default\NextGenForms\js" folder.
Otherwise, it will not be rendered.
 this.TopLevelForm.RuntimeJS.Add("MyJsFile", "js/Skelta.maskedinput.js");
}
}
/// <summary>
/// Rules for the control - Add custom validations for the control when value of the
control changes.
/// </summary>
public override Collection<Rule> Rules
{
get
{
// By default a few rules like Mandatory are added.
 // We recommend to insert or add rules instead of creating a new rules collection.
Collection<Rule> rules = base.Rules;
// RuleFor -  Associated property for the rule.
// RuleName - JavaScript function name for the rule.
// RuleDefnition - Error message for the rule.
// ControlType - Associated control for the rule.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 // RenderMode - Render the control in HTML5 or HTML.
rules.Insert(0, new Rule { RuleFor = "Value", RuleName =
"Skelta.forms.maskedInput.validation", RuleDefinition = "Enter valid data (mask text: " +
this.Mask + ")", ControlType = this.GetType().Name });
return rules;
}
}
}
}
```

skelta.maskedinput.js

```
(function ()
{
if (skelta.utilities.isUndefined(skelta.forms))
{
// Create the Skelta.Forms namespace
skelta.createNamespace("skelta.forms");
}
skelta.forms.maskedInput = (function ()
{
function maskOperation(elem)
{
var cpf = elem;
//getting key code of pressed key
var key = window.event;
var keycode = (key.which) ? key.which : key.keyCode;
//comparing pressed keycodes
if (keycode > 31 && (keycode < 48 || keycode > 57))
{
//(" You can enter only characters 0 to 9 ");
return false;
}
if ((cpf.value.length == cpf.placeholder.indexOf("-", cpf.value.length)))
{
cpf.value += "-";
return true;
}
if (cpf.value.length == cpf.placeholder.length || cpf.value.length >
cpf.placeholder.length)
{
return false;
}
return true;
}
function maskValidation(controlvalue, control, ruleObject)
{
if (!controlvalue) return;
var cpf = control.domElement;
var errMessage = ruleObject.errorMessage;
if (cpf.placeholder.length != controlvalue.length)
{
errorBoolValue = true;
control.validationError(true);
control.validationErrorMessage(errMessage);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.validationErrorCode(errorConstants.valueValidationRules);
}
}
return {
keyPress: maskOperation,
validation: maskValidation
}
}());
})();
```

Add Masked Input Property for Check Syntax Validation

The following code represents how to add the "Mask" property along with the other common properties for check syntax validation.

```
(function ()
{
var standardControlMembers = skelta.forms.constants.controlsCollection;
var genericControlFunctions =
skelta.forms.constants.controlFunctions;//genericControlFunctions has properties -
nonContainerFunctions and containerFunctions which is a collection
of function names [{ name: "functionName"}]
var maskedInputMembers = {
functions: genericControlFunctions.nonContainerFunctions,//.concat([{ name: "functionName"
}
isContainerControl: false,
controlType: "MaskedInput",
properties: [{ propertyName: "xmlNodeBoundTo", propertyType: "2", isWritable: false },
{ propertyName: "tagName", propertyType: "0", isWritable: true },
{ propertyName: "id", propertyType: "2", isWritable: false },
{ propertyName: "value", propertyType: "0", isWritable: true },
{ propertyName: "isMandatory", propertyType: "0", isWritable: true },
{ propertyName: "readOnly", propertyType: "0", isWritable: true },
{ propertyName: "enable", propertyType: "0", isWritable: true },
{ propertyName: "mask", propertyType: "2", isWritable: false },
{ propertyName: "visible", propertyType: "0", isWritable: true }]
}
standardControlMembers.push(maskedInputMembers);
})();
```

In the example, the "Mask" property is a nonobservable variable and is a readonly property, indicated by **propertyType: "2"** and **isWritable: false** in the above code.

For more information on the usage of the above code, see Adding Custom Controls for Check Syntax Validation.

Custom Controls with Kendo UI

You can create a custom control with Kendo UI widget to design a control with specific or custom properties of Kendo UI widgets.

- Create Custom Controls with Kendo UI
- Create Knockout Bindings for Kendo UI
- Knockout Bindings for Kendo UI

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create Custom Controls with Kendo UI

Create a custom control with Kendo UI widget to design a control with specific or custom properties of Kendo UI widgets.

To create a custom control with a Kendo UI Widget

1. Create a DLL project (**Visual C# > Windows > Class Library**) using Visual Studio.NET.
2. Create or add two .cs files – **SFControls.cs** (control initialization file) and **MyCustom.cs** (public control class file) to your project.
3. Add a reference to the **Skelta.Forms.Core.dll** (located at **<AVEVA Work Tasks Install Path>\Bin** folder) in your project.
4. Add the following namespaces to your **MyCustom.cs** file.

   ```
   using Skelta.Forms.Core;
   using Skelta.Forms.Core.External;
   using Skelta.Forms.Core.CommonObjects;
   using Skelta.Forms.Core.Controls;
   using Skelta.Forms.Core.Interfaces;
   ```

5. Ensure the control class (**MyCustom.cs**) inherits from the **BaseDataExternalControl** class.
6. Write the control definition code with the following in mind:
   a. Ensure the **Core Designer** is set to **Next Generation Designer** and the **Designer Category** is set to **Custom** to show up the control in the toolbox of the **Forms Designer**.

      ```
      [DesignerControl(992200, CoreDesigner.NextGenDesigner, DesignerCategory.Custom)]
      public class MyControl : BaseDataExternalControl
      ```

   b. Suppress properties, if required, using the attribute **SuppressedPropertiesNextGen**.

      ```
      // Suppress properites PropertyName1, PropertyName2, and PropertyName3.
      [SuppressedPropertiesNextGen("PropertyName1;PropertyName2;PropertyName3;")]
      ```

   c. Override the **GetNewInstanceForClone()** function, create an instance of the control, and assign the basic properties of the control.
   d. Override the **GetControlHtml()** function and define the HTML code. Use the **TopLevelForm.InDesignMode** boolean variable to segregate the designer and run-time views, if required. Ensure all view model properties are specific to the instance defined in the view. For examples, refer to Knockout-Kendo examples.
   e. Override the **GetViewModelDefinition()** function (if required) to define the **ViewModel** definition of the Knockout binding based on the custom control Binding Attribute (For more information, see *Binding Attribute* in Attributing Properties). Ensure all view model properties are specific to the instance defined in the view model. For more information, see Autocomplete.cs.
   f. Override the **GetViewModelScript()** function (if required) to define the **ViewModel** script based on the predefined named script like **Value**, **Validate**, **Mandatory** and so on.
   g. Build the following JS files to include the specific **Kendo UI** widget and the **Knockout** bindings for the specific **Kendo UI** widget.

      – kendo.custom.min.js (for the specific **Kendo UI** widget). For more information, see Example: RichTextBox Control.

      – knockout-kendo.custom.min.js (for the **Knockout** bindings). For more information, see Create Knockout Bindings for Kendo UI.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

h. Ensure you add the JS file reference – for the specific **Kendo UI** widget followed by the **Knockout** bindings for the specific **Kendo UI** widget – to **this.TopLevelForm.RuntimeJS** variable to add custom JS files referred by the custom control in the **GetControlHtml()** function. The variable is a key value pair collection where the key is the file name and the value is the file path (absolute or relative). Also ensure the JS files are present in the **BPMUITemplates\Default\NextGenForms\js** folder. Otherwise, it will not be rendered.

i. Ensure you add the CSS file reference to **this.TopLevelForm.RuntimeCss** variable to add custom CSS files referred by the custom control in the **GetControlHtml()** function. The variable is a key value pair collection where the key is the file name and the value is the file path (absolute or relative). Also ensure the CSS files are present in the **BPMUITemplates\Default\NextGenForms\css** folder. Otherwise, it will not be rendered.

j. Define rules for a custom control to validate the data using the client-side JavaScript, if required. For more information, see Rules and Defining Rules.

k. Create class properties as control properties.

```
//Designer Property
[DesignerProperty(773000, Categories.Validation, CoreDesigner.NextGenDesigner)]
//Binding Property
[BindingAttribute(882995, "Skelta.Forms.Core.CommonObjects.IntegerBindingProperty")]
```

7. Create **SFControls.cs** class under **Skelta.Forms.ExternalControls** namespace.
8. Add the following namespaces to your **SFControls.cs** file.

```
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.External;
using MyCustomControls; // Control namespace
```

9. Implement the **IExternalControls** in the **SFControls** class.
   a. In the **IExternalControls.GetControl(string controlFullName)** method, return the new control instance based on the control namespace.
   b. In the **GetControls** method, return a new object array which contains all the control instances.
10. Build your project, create a **SOA** folder on your hard drive (for example, **C:\SOA**), place your custom control DLL file in the folder, and then register the location of the folder with the **Skelta SOA Folder List**.
11. Register the custom control DLL file through the **Form Controls** to display the custom control in the controls or custom category. For more information, see Custom Categories.

Note:
- Control initialization file must always be named as **SFControls.cs** and the class name must always be named as **SFControls**.
- Ensure reference to the **Skelta.Forms.Core.dll** assembly is the same as the assembly on the AVEVA Work Tasks server.
- For information about SOA, see *Object Access* in the *Workflow Management User Guide*.

To localize the text for the custom control

- See Localization.

To add an icon to a custom control

1. Add a custom icon in the **<AVEVA Work Tasks Install Path>\Web\BPMUITemplates\Default\Themes\<ThemeName>\NextGenForms\images** folder.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Open the **SkeltaCustomControlDesigner.css** file in the **<AVEVA Work Tasks Install Path>\Web\BPMUITemplates\Default\Themes\<ThemeName>\NextGenForms\css\custom** folder.

1. Add the following style to the **SkeltaCustomControlDesigner.css** file.

```
/* Replace <custom_icon> with the file name for the normal icon. */
/* Replace <namespace.classname> with the custom control namespace and class name. */
.sktbci[data-skttp="<namespace.classname>"]
{
background-image: url(../../images/<custom_icon>.png);
}
/* Replace <custom_icon_hover> with the file name for the hover icon. */
.sktbc:hover > .sktbci[data-skttp="<namespace.classname>"]
{
background-image: url(../../images/<custom_icon_hover>.png);
}
```

2. Save the **SkeltaCustomControlDesigner.css** file.
3. Reload the **Forms Designer** page to see the custom control with the custom icon in the toolbox.

Note: The Kendo UI JavaScript files - **kendo.all.full.js** and **knockout-kendo.full.js** - are available at **BPMUITemplates/Default/NextGenForms/js/** folder for reference.

Create Knockout Bindings for Kendo UI

Create **Knockout** bindings for specific **Kendo UI** widgets.

To create the Knockout bindings for the specific Kendo UI Widget

1. Create and add the following code to your **knockout-kendo.custom.min.js** file.

```
(function ()
{
var createBinding = ko.kendo.bindingFactory.createBinding.bind(ko.kendo.bindingFactory);
//use constants to ensure consistency and to help reduce minified file size
var CLICK = "click",
CLICKED = "clicked",
CLOSE = "close",
COLLAPSE = "collapse",
CONTENT = "content",
DATA = "data",
ENABLE = "enable",
EXPAND = "expand",
ENABLED = "enabled",
EXPANDED = "expanded",
ERROR = "error",
FILTER = "filter",
HIDE = "hide",
INFO = "info",
ISOPEN = "isOpen",
MAX = "max",
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
MIN = "min",
OPEN = "open",
PALETTE = "palette",
READONLY = "readonly",
RESIZE = "resize",
SEARCH = "search",
SELECT = "select",
SELECTED = "selected",
SHOW = "show",
SUCCESS = "success",
SIZE = "size",
TARGET = "target",
TITLE = "title",
VALUE = "value",
VALUES = "values",
WARNING = "warning";
}());
```

2. Create binding for the specific **Kendo UI** widget to the function.

   For information on **Knockout** bindings for **Kendo UI** widgets, see Knockout Bindings for Kendo UI (In this file, the **Kendo UI** widgets that are currently not used are commented. Uncomment the specific **Kendo UI** widget to create the bindings to the function).

```
// For example, to create Knockout binding for Autocomplete Kendo UI widget, use the
following code
createBinding({
name: "kendoAutoComplete",
events: {
change: VALUE,
open: {
writeTo: ISOPEN,
value: true
},
close: {
writeTo: ISOPEN,
value: false
}
},
watch: {
enabled: ENABLE,
search: [SEARCH, CLOSE],
data: function (value)
{
ko.kendo.setDataSource(this, value);
},
value: VALUE
}
});
```

Knockout Bindings for Kendo UI

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
/*
* knockout-kendo 0.8.1
* Copyright © 2013 Ryan Niemeyer & Telerik
*
* Licensed under the Apache License, Version 2.0 (the "License");
* you may not use this file except in compliance with the License.
* You may obtain a copy of the License at
* http://www.apache.org/licenses/LICENSE-2.0
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/
(function (factory)
{
// CommonJS
if (typeof require === 'function' && typeof exports === 'object' && typeof module ===
'object')
{
factory(require('knockout'), require('jquery'), require('kendo'));
// AMD
} else if (typeof define === 'function' && define.amd)
{
define(['knockout', 'jquery', 'kendo'], factory);
// Normal script tag
} else
{
factory(window.ko, window.jQuery, window.kendo);
}
}(function (ko, $, kendo, undefined)
{
//handle require.js scenarios where kendo is not actually returned
kendo = kendo || window.kendo;
ko.kendo = ko.kendo || {};
ko.kendo.BindingFactory = function ()
{
var self = this;
this.createBinding = function (widgetConfig)
{
//only support widgets that are available when this script runs
if (!$()[widgetConfig.parent || widgetConfig.name])
{
return;
}
var binding = {};
//the binding handler's init function
binding.init = function (element, valueAccessor, all, vm, context)
{
//step 1: build appropriate options for the widget from values passed in and global options
var options = self.buildOptions(widgetConfig, valueAccessor);
//apply async, so inner templates can finish content needed during widget initialization
if (options.async === true || (widgetConfig.async === true && options.async !== false))
{
setTimeout(function ()
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
binding.setup(element, options, context);
}, 0);
return;
}
binding.setup(element, options, context);
if (options && options.useKOTemplates)
{
return { controlsDescendantBindings: true };
}
};
//build the core logic for the init function
binding.setup = function (element, options, context)
{
var widget, $element = $(element);
//step 2: setup templates
self.setupTemplates(widgetConfig.templates, options, element, context);
////
//// Note:
//// Below fix is for the Data Grid firing dataBound Event Multiple Times on Load
//// Instead of adding watch, using data for setting the grid dataSource
//// Before the kendo grid widget is initialized, options.dataSource is set to options.data
////
//// ---- Code Added ----
if (options.useKOTemplates === true && options.data !== undefined)
{
options.dataSource = options.data;
}
//// ---- /Code Added ----
//step 3: initialize widget
widget = self.getWidget(widgetConfig, options, $element);
//step 4: add handlers for events that we need to react to for updating the model
self.handleEvents(options, widgetConfig, element, widget, context);
//step 5: set up computed observables to update the widget when observable model values
change
self.watchValues(widget, options, widgetConfig, element);
//step 6: handle disposal, if there is a destroy method on the widget
if (widget.destroy)
{
ko.utils.domNodeDisposal.addDisposeCallback(element, function ()
{
widget.destroy();
});
}
};
binding.options = {}; //global options
binding.widgetConfig = widgetConfig; //expose the options to use in generating tests
ko.bindingHandlers[widgetConfig.bindingName || widgetConfig.name] = binding;
};
//combine options passed in binding with global options
this.buildOptions = function (widgetConfig, valueAccessor)
{
var defaultOption = widgetConfig.defaultOption,
options = ko.utils.extend({}, ko.bindingHandlers[widgetConfig.name].options),
valueOrOptions = ko.utils.unwrapObservable(valueAccessor());
if (valueOrOptions instanceof kendo.data.DataSource || typeof valueOrOptions !== "object"
|| valueOrOptions === null || (defaultOption && !(defaultOption in valueOrOptions)))
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
options[defaultOption] = valueAccessor();
} else
{
ko.utils.extend(options, valueOrOptions);
}
return options;
};
var templateRenderer = function (id, context)
{
return function (data)
{
return ko.renderTemplate(id, context.createChildContext((data._raw && data._raw()) ||
data));
};
};
//prepare templates, if the widget uses them
this.setupTemplates = function (templateConfig, options, element, context)
{
var i, j, option, existingHandler;
if (templateConfig && options && options.useKOTemplates)
{
//create a function to render each configured template
for (i = 0, j = templateConfig.length; i < j; i++)
{
option = templateConfig[i];
if (options[option])
{
options[option] = templateRenderer(options[option], context);
}
}
//initialize bindings in dataBound event
existingHandler = options.dataBound;
options.dataBound = function ()
{
ko.memoization.unmemoizeDomNodeAndDescendants(element);
if (existingHandler)
{
existingHandler.apply(this, arguments);
}
};
}
};
//unless the object is a kendo datasource, get a clean object with one level unwrapped
this.unwrapOneLevel = function (object)
{
var prop,
result = {};
if (object)
{
if (object instanceof kendo.data.DataSource)
{
result = object;
}
else if (typeof object === "object")
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
for (prop in object)
{
//include things on prototype
result[prop] = ko.utils.unwrapObservable(object[prop]);
}
}
}
return result;
};
//return the actual widget
this.getWidget = function (widgetConfig, options, $element)
{
var widget;
if (widgetConfig.parent)
{
//locate the actual widget
var parent = $element.closest("[data-bind*='" + widgetConfig.parent + ":']");
widget = parent.length ? parent.data(widgetConfig.parent) : null;
} else
{
widget = $element[widgetConfig.name](this.unwrapOneLevel(options)).data(widgetConfig.name);
}
//if the widget option was specified, then fill it with our widget
if (ko.isObservable(options.widget))
{
options.widget(widget);
}
return widget;
};
//respond to changes in the view model
this.watchValues = function (widget, options, widgetConfig, element)
{
var watchProp, watchValues = widgetConfig.watch;
if (watchValues)
{
for (watchProp in watchValues)
{
if (watchValues.hasOwnProperty(watchProp))
{
self.watchOneValue(watchProp, widget, options, widgetConfig, element);
}
}
}
};
this.watchOneValue = function (prop, widget, options, widgetConfig, element)
{
var computed = ko.computed({
read: function ()
{
var existing, custom,
action = widgetConfig.watch[prop],
value = ko.utils.unwrapObservable(options[prop]),
params = widgetConfig.parent ? [element] : []; //child bindings pass element first to APIs
//support passing multiple events like ["open", "close"]
if ($.isArray(action))
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
action = widget[value ? action[0] : action[1]];
} else if (typeof action === "string")
{
action = widget[action];
} else
{
custom = true; //running a custom function
}
if (action && options[prop] !== undefined)
{
if (!custom)
{
existing = action.apply(widget, params);
if (widgetConfig.name === "kendoColorPicker" && prop === "value" && !value)
{
value = "#1b1b1b";
}
params.push(value);
} else
{
params.push(value, options);
}
//try to avoid unnecessary updates when the new value matches the current value
if (custom || existing !== value)
{
action.apply(widget, params);
}
}
},
disposeWhenNodeIsRemoved: element
}).extend({ throttle: (options.throttle || options.throttle === 0) ? options.throttle : 1
});
//if option is not observable, then dispose up front after executing the logic once
if (!ko.isObservable(options[prop]))
{
computed.dispose();
}
};
//write changes to the widgets back to the model
this.handleEvents = function (options, widgetConfig, element, widget, context)
{
var prop, eventConfig, events = widgetConfig.events;
if (events)
{
for (prop in events)
{
if (events.hasOwnProperty(prop))
{
eventConfig = events[prop];
if (typeof eventConfig === "string")
{
eventConfig = { value: eventConfig, writeTo: eventConfig };
}
self.handleOneEvent(prop, eventConfig, options, element, widget, widgetConfig.childProp,
context);
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
}
};
//bind to a single event
this.handleOneEvent = function (eventName, eventConfig, options, element, widget,
childProp, context)
{
var handler = typeof eventConfig === "function" ? eventConfig : options[eventConfig.call];
//call a function defined directly in the binding definition, supply options that were
passed to the binding
if (typeof eventConfig === "function")
{
handler = handler.bind(context.$data, options);
}
//use function passed in binding options as handler with normal KO args
else if (eventConfig.call && typeof options[eventConfig.call] === "function")
{
handler = options[eventConfig.call].bind(context.$data, context.$data);
}
//option is observable, determine what to write to it
else if (eventConfig.writeTo && ko.isWriteableObservable(options[eventConfig.writeTo]))
{
handler = function (e)
{
var propOrValue, value;
if (!childProp || !e[childProp] || e[childProp] === element)
{
propOrValue = eventConfig.value;
value = (typeof propOrValue === "string" && this[propOrValue]) ?
this[propOrValue](childProp && element) : propOrValue;
options[eventConfig.writeTo](value);
}
};
}
if (handler)
{
widget.bind(eventName, handler);
}
};
};
ko.kendo.bindingFactory = new ko.kendo.BindingFactory();
//utility to set the dataSource with a clean copy of data. Could be overridden at run-time.
ko.kendo.setDataSource = function (widget, data, options)
{
var isMapped, cleanData;
if (data instanceof kendo.data.DataSource)
{
widget.setDataSource(data);
return;
}
if (!options || !options.useKOTemplates)
{
isMapped = ko.mapping && data && data.__ko_mapping__;
cleanData = data && isMapped ? ko.mapping.toJS(data) : ko.toJS(data);
}
widget.dataSource.data(cleanData || data);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
};
//attach the raw data after Kendo wraps our items
(function ()
{
var existing = kendo.data.ObservableArray.fn.wrap;
kendo.data.ObservableArray.fn.wrap = function (object)
{
var result = existing.apply(this, arguments);
result._raw = function ()
{
return object;
};
return result;
};
})();
//private utility function generator for gauges
var extendAndRedraw = function (prop)
{
return function (value)
{
if (value)
{
ko.utils.extend(this.options[prop], value);
this.redraw();
this.value(0.001 + this.value());
}
};
};
//library is in a closure, use this private variable to reduce size of minified file
var createBinding = ko.kendo.bindingFactory.createBinding.bind(ko.kendo.bindingFactory);
//use constants to ensure consistency and to help reduce minified file size
var CLICK = "click",
CLICKED = "clicked",
CLOSE = "close",
COLLAPSE = "collapse",
CONTENT = "content",
DATA = "data",
ENABLE = "enable",
EXPAND = "expand",
ENABLED = "enabled",
EXPANDED = "expanded",
ERROR = "error",
FILTER = "filter",
HIDE = "hide",
INFO = "info",
ISOPEN = "isOpen",
MAX = "max",
MIN = "min",
OPEN = "open",
PALETTE = "palette",
READONLY = "readonly",
RESIZE = "resize",
SEARCH = "search",
SELECT = "select",
SELECTED = "selected",
SHOW = "show",
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
SUCCESS = "success",
SIZE = "size",
TARGET = "target",
TITLE = "title",
VALUE = "value",
VALUES = "values",
WARNING = "warning";
//createBinding({
// name: "kendoAutoComplete",
// events: {
// change: VALUE,
// open: {
// writeTo: ISOPEN,
// value: true
// },
// close: {
// writeTo: ISOPEN,
// value: false
// }
// },
// watch: {
// enabled: ENABLE,
// search: [SEARCH, CLOSE],
// data: function (value)
// {
// ko.kendo.setDataSource(this, value);
// },
// value: VALUE
// }
//});
//createBinding({
// name: "kendoButton",
// defaultOption: CLICKED,
// events: {
// click: {
// call: CLICKED
// }
// },
// watch: {
// enabled: ENABLE
// }
//});
createBinding({
name: "kendoCalendar",
defaultOption: VALUE,
events: {
change: VALUE
},
watch: {
max: MAX,
min: MIN,
value: VALUE
}
});
createBinding({
name: "kendoColorPicker",
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
events: {
change: VALUE,
open: {
writeTo: ISOPEN,
value: true
},
close: {
writeTo: ISOPEN,
value: false
}
},
watch: {
enabled: ENABLE,
value: VALUE,
color: VALUE,
palette: PALETTE
}
});
//createBinding({
// name: "kendoComboBox",
// events: {
// change: VALUE,
// open: {
// writeTo: ISOPEN,
// value: true
// },
// close: {
// writeTo: ISOPEN,
// value: false
// }
// },
// watch: {
// enabled: ENABLE,
// isOpen: [OPEN, CLOSE],
// data: function (value)
// {
// ko.kendo.setDataSource(this, value);
// },
// value: VALUE
// }
//});
createBinding({
name: "kendoDatePicker",
defaultOption: VALUE,
events: {
change: VALUE,
open:
{
writeTo: ISOPEN,
value: true
},
close: {
writeTo: ISOPEN,
value: false
}
},
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
watch: {
readOnly: READONLY,
enabled: ENABLE,
max: MAX,
min: MIN,
value: VALUE,
isOpen: [OPEN, CLOSE]
}
});
createBinding({
name: "kendoDateTimePicker",
defaultOption: VALUE,
events: {
change: VALUE,
open:
{
writeTo: ISOPEN,
value: true
},
close: {
writeTo: ISOPEN,
value: false
}
},
watch: {
readOnly: READONLY,
enabled: ENABLE,
max: MAX,
min: MIN,
value: VALUE,
isOpen: [OPEN, CLOSE]
}
});
//createBinding({
// name: "kendoDropDownList",
// events: {
// change: VALUE,
// open: {
// writeTo: ISOPEN,
// value: true
// },
// close: {
// writeTo: ISOPEN,
// value: false
// }
// },
// watch: {
// enabled: ENABLE,
// isOpen: [OPEN, CLOSE],
// data: function (value)
// {
// ko.kendo.setDataSource(this, value);
// //if nothing is selected and there is an optionLabel, select it
// if (value.length && this.options.optionLabel && this.select() < 0)
// {
// this.select(0);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// }
// },
// value: VALUE
// }
//});
//createBinding({
// name: "kendoEditor",
// defaultOption: VALUE,
// events: {
// change: VALUE
// },
// watch: {
// enabled: ENABLE,
// value: VALUE
// }
//});
////
//// Note: { Prafulla.Jain }
//// Below fix is for the Data Grid firing dataBound Event Multiple Times on Load
//// Instead of adding watch, using data for setting the grid dataSource
//// Before the kendo grid widget is initialized, options.dataSource is set to options.data
////
//// ---- Original Code ----
////createBinding({
//// name: "kendoGrid",
//// defaultOption: DATA,
//// watch: {
//// data: function (value, options)
//// {
//// ko.kendo.setDataSource(this, value, options);
//// }
//// },
//// templates: ["rowTemplate", "altRowTemplate"]
////});
//// ---- Modified Code ----
createBinding({
name: "kendoGrid",
defaultOption: DATA,
watch: "data",
templates: ["rowTemplate", "altRowTemplate"]
});
//createBinding({
// name: "kendoListView",
// defaultOption: DATA,
// watch: {
// data: function (value, options)
// {
// ko.kendo.setDataSource(this, value, options);
// }
// },
// templates: ["template"]
//});
//createBinding({
// name: "kendoMaskedTextBox",
// defaultOption: VALUE,
// events: {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// change: VALUE
// },
// watch: {
// enabled: ENABLE,
// isReadOnly: READONLY,
// value: VALUE
// }
//});
//createBinding({
// name: "kendoMenu",
// async: true
//});
//createBinding({
// name: "kendoMenuItem",
// parent: "kendoMenu",
// watch: {
// enabled: ENABLE,
// isOpen: [OPEN, CLOSE]
// },
// async: true
//});
//createBinding({
// name: "kendoMultiSelect",
// events: {
// change: VALUE,
// open: {
// writeTo: ISOPEN,
// value: true
// },
// close: {
// writeTo: ISOPEN,
// value: false
// }
// },
// watch: {
// enabled: ENABLE,
// search: [SEARCH, CLOSE],
// data: function (value)
// {
// ko.kendo.setDataSource(this, value);
// },
// value: VALUE
// }
//});
//var notificationHandler = function (type, value)
//{
// if (value || value === 0)
// {
// this.show(value, type);
// }
// else
// {
// this.hide();
// }
//};
//createBinding({
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// name: "kendoNotification",
// watch: {
// error: function (value)
// {
// notificationHandler.call(this, ERROR, value);
// },
// info: function (value)
// {
// notificationHandler.call(this, INFO, value);
// },
// success: function (value)
// {
// notificationHandler.call(this, SUCCESS, value);
// },
// warning: function (value)
// {
// notificationHandler.call(this, WARNING, value);
// }
// }
//});
createBinding({
name: "kendoNumericTextBox",
defaultOption: VALUE,
events: {
change: VALUE
},
watch: {
readOnly: READONLY,
enabled: ENABLE,
value: VALUE,
max: function (newMax)
{
this.options.max = newMax;
//make sure current value is still valid
////var value = this.value();
////if ((value || value === 0) && value > newMax)
////{
//// this.value(newMax);
////}
},
min: function (newMin)
{
this.options.min = newMin;
//make sure that current value is still valid
////var value = this.value();
////if ((value || value === 0) && value < newMin)
////{
//// this.value(newMin);
////}
}
}
});
//createBinding({
// name: "kendoPanelBar",
// async: true
//});
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//createBinding({
// name: "kendoPanelItem",
// parent: "kendoPanelBar",
// watch: {
// enabled: ENABLE,
// expanded: [EXPAND, COLLAPSE],
// selected: [SELECT]
// },
// childProp: "item",
// events: {
// expand: {
// writeTo: EXPANDED,
// value: true
// },
// collapse: {
// writeTo: EXPANDED,
// value: false
// },
// select: {
// writeTo: SELECTED,
// value: VALUE
// }
// },
// async: true
//});
//createBinding({
// name: "kendoProgressBar",
// defaultOption: VALUE,
// events: {
// change: VALUE
// },
// watch: {
// enabled: ENABLE,
// value: VALUE
// }
//});
//createBinding({
// name: "kendoRangeSlider",
// defaultOption: VALUES,
// events: {
// change: VALUES
// },
// watch: {
// values: VALUES,
// enabled: ENABLE
// }
//});
//createBinding({
// async: true,
// name: "kendoScheduler",
// watch: {
// data: function (value, options)
// {
// ko.kendo.setDataSource(this, value, options);
// }
// }
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//});
//createBinding({
// name: "kendoSlider",
// defaultOption: VALUE,
// events: {
// change: VALUE
// },
// watch: {
// value: VALUE,
// enabled: ENABLE
// }
//});
//createBinding({
// name: "kendoSortable",
// defaultOption: DATA,
// events: {
// end: function (options, e)
// {
// var dataKey = "__ko_kendo_sortable_data__",
// data = e.action !== "receive" ? ko.dataFor(e.item[0]) : e.draggableEvent[dataKey],
// items = options.data,
// underlyingArray = options.data;
// //remove item from its original position
// if (e.action === "sort" || e.action === "remove")
// {
// underlyingArray.splice(e.oldIndex, 1);
// //keep track of the item between remove and receive
// if (e.action === "remove")
// {
// e.draggableEvent[dataKey] = data;
// }
// }
// //add the item to its new position
// if (e.action === "sort" || e.action === "receive")
// {
// underlyingArray.splice(e.newIndex, 0, data);
// //clear the data we passed
// delete e.draggableEvent[dataKey];
// //we are moving the item ourselves via the observableArray, cancel the draggable and
hide the animation
// $(e.draggableEvent.target).hide();
// e.preventDefault();
// }
// //signal that the observableArray has changed now that we are done changing the array
// items.valueHasMutated();
// }
// }
//});
//createBinding({
// name: "kendoSplitter",
// async: true
//});
//createBinding({
// name: "kendoSplitterPane",
// parent: "kendoSplitter",
// watch: {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// max: MAX,
// min: MIN,
// size: SIZE,
// expanded: [EXPAND, COLLAPSE]
// },
// childProp: "pane",
// events: {
// collapse: {
// writeTo: EXPANDED,
// value: false
// },
// expand: {
// writeTo: EXPANDED,
// value: true
// },
// resize: SIZE
// },
// async: true
//});
//createBinding({
// name: "kendoTabStrip",
// async: true
//});
//createBinding({
// name: "kendoTab",
// parent: "kendoTabStrip",
// watch: {
// enabled: ENABLE
// },
// childProp: "item",
// async: true
//});
//createBinding({
// name: "kendoTooltip",
// events: {},
// watch: {
// content: CONTENT,
// filter: FILTER
// }
//});
createBinding({
name: "kendoTimePicker",
defaultOption: VALUE,
events: {
change: VALUE
},
watch: {
readOnly: READONLY,
max: MAX,
min: MIN,
value: VALUE,
enabled: ENABLE,
isOpen: [OPEN, CLOSE]
}
});
//createBinding({
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// name: "kendoTreeView",
// async: true
//});
//createBinding({
// name: "kendoTreeItem",
// parent: "kendoTreeView",
// watch: {
// enabled: ENABLE,
// expanded: [EXPAND, COLLAPSE],
// selected: function (element, value)
// {
// if (value)
// {
// this.select(element);
// } else if (this.select()[0] == element)
// {
// this.select(null);
// }
// }
// },
// childProp: "node",
// events: {
// collapse: {
// writeTo: EXPANDED,
// value: false
// },
// expand: {
// writeTo: EXPANDED,
// value: true
// },
// select: {
// writeTo: SELECTED,
// value: true
// }
// },
// async: true
//});
//createBinding({
// name: "kendoUpload",
// watch: {
// enabled: ENABLE
// }
//});
//createBinding({
// async: true,
// name: "kendoWindow",
// events: {
// open: {
// writeTo: ISOPEN,
// value: true
// },
// close: {
// writeTo: ISOPEN,
// value: false
// }
// },
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// watch: {
// content: CONTENT,
// title: TITLE,
// isOpen: [OPEN, CLOSE]
// }
//});
//createBinding({
// name: "kendoChart",
// watch: {
// data: function (value)
// {
// ko.kendo.setDataSource(this, value);
// }
// }
//});
//createBinding({
// name: "kendoLinearGauge",
// defaultOption: VALUE,
// watch: {
// value: VALUE,
// gaugeArea: extendAndRedraw("gaugeArea"),
// pointer: extendAndRedraw("pointer"),
// scale: extendAndRedraw("scale")
// }
//});
//createBinding({
// name: "kendoRadialGauge",
// defaultOption: VALUE,
// watch: {
// value: VALUE,
// gaugeArea: extendAndRedraw("gaugeArea"),
// pointer: extendAndRedraw("pointer"),
// scale: extendAndRedraw("scale")
// }
//});
}));
```

Example: Barchart Control

A custom **Barchart** control using Kendo UI widget with control initialization, class, JavaScript, and CSS files.

- SFControl.cs (control initialization file)
- BarChart.cs (control class file)
- BarChartBindingProperty.cs (Bindings for the control)
- knockout-kendo.kendochart.js (JavaScript file for Knockout binding of the specific Kendo UI widget)
- corp.customcontrol.barchart.js (JavaScript initialization file)
- Adding Barchart Properties for Check Syntax Validation

SFControl.cs

```
using Corp.CustomControls.BarChartCustom;
using Skelta.Forms.Core.Controls;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using Skelta.Forms.Core.External;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Skelta.Forms.ExternalControls
{
public class SFControls : IExternalControls
{
BaseControl IExternalControls.GetControl(string controlFullName)
{
switch (controlFullName)
{
case "Corp.CustomControls.BarChartCustom.BarChart":
return new BarChart();
default:
return null;
}
}
object[] IExternalControls.GetControls()
{
return new object[] { new BarChart() };
}
}
}
```

BarChart.cs

```
using System;
using System.ComponentModel;
using System.Text;
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.Interfaces;
namespace Corp.CustomControls.BarChartCustom
{
//Adding the attribute to make it available in the forms designer
[DesignerControl(992200, CoreDesigner.NextGenDesigner, DesignerCategory.Custom)]
public class BarChart : BaseDataExternalControl
{
public override BaseControl GetNewInstanceForClone()
{
var custom = new BarChart();
custom.ChartType = this.ChartType;
return custom;
}
public override string TypeAssociation
{
get
{
return "Custom";
}
}
private string chartType = "Column";
private string _items = string.Empty;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
private string _series = string.Empty;
private string _colors = string.Empty;
private string _itemsURL = string.Empty;
private string _seriesURL = string.Empty;
private string _colorsURL = string.Empty;
[DefaultValue("Column")]
[DesignerProperty(774000, Categories.Advanced, CoreDesigner.NextGenDesigner)]
public string ChartType
{
get { return chartType; }
set { chartType = value; }
}
[DesignerProperty(774100, Categories.Advanced, CoreDesigner.NextGenDesigner)]
[BindingAttribute(889900, "Skelta.Forms.Core.CommonObjects.CommonStringBindingProperty",
BindingPropertyName = "itemsData", IsObservable = false)]
public string Items
{
get { return _items; }
set { _items = value; }
}
[DesignerProperty(774150, Categories.Advanced, CoreDesigner.NextGenDesigner)]
[BindingAttribute(889950, "Skelta.Forms.Core.CommonObjects.CommonStringBindingProperty",
BindingPropertyName = "seriesData", IsObservable = false)]
public string Series
{
get { return _series; }
set { _series = value; }
}
[DesignerProperty(774200, Categories.Advanced, CoreDesigner.NextGenDesigner)]
[BindingAttribute(8810000, "Skelta.Forms.Core.CommonObjects.CommonStringBindingProperty",
BindingPropertyName = "colorsData", IsObservable = false)]
public string Colors
{
get { return _colors; }
set { _colors = value; }
}
[DesignerProperty(774250, Categories.Advanced, CoreDesigner.NextGenDesigner)]
[BindingAttribute(8810100, "Skelta.Forms.Core.CommonObjects.CommonStringBindingProperty",
BindingPropertyName = "itemsURL", IsObservable = false)]
public string ItemsURL
{
get { return _itemsURL; }
set { _itemsURL = value; }
}
[DesignerProperty(774300, Categories.Advanced, CoreDesigner.NextGenDesigner)]
[BindingAttribute(8810200, "Skelta.Forms.Core.CommonObjects.CommonStringBindingProperty",
BindingPropertyName = "seriesURL", IsObservable = false)]
public string SeriesURL
{
get { return _seriesURL; }
set { _seriesURL = value; }
}
[DesignerProperty(774350, Categories.Advanced, CoreDesigner.NextGenDesigner)]
[BindingAttribute(8810300, "Skelta.Forms.Core.CommonObjects.CommonStringBindingProperty",
BindingPropertyName = "colorsURL", IsObservable = false)]
public string ColorsURL
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
get { return _colorsURL; }
set { _colorsURL = value; }
}
[BindingAttribute(8810400, true, "Corp.CustomControls.BarChart.BarChartBindingProperty")]
public override string ControlBindings
{
get;
set;
}
public override string GetView()
{
AddJsFiles();
return base.GetView();
}
public override string GetControlHtml()
{
var stringBuilder = new StringBuilder();
StringBuilder dataBindAttribute = new StringBuilder();
//Creating the binding for the controls, by default all these properties are rendered, If
any other Knockout binding like "attr" etc, can be added here.
//"DataBindAttribute" is the keyword, which is mandatory data: items, series: [{ name:
'sample', field: 'value' }]
//We can have separate views for design mode using the below keyword
"this.TopLevelForm.InDesignMode", which will be true while rendering in design mode.
if (this.TopLevelForm.InDesignMode == false)
{
dataBindAttribute.Append(" " + DataBindAttribute + "=\"kendoChart: { data: " + this.Id +
"._items, title: { text: 'Graph Sample' }, series: " + this.Id + "._series, seriesColors: "
+ this.Id + "._colors}");
dataBindAttribute.Append("\"");
}
stringBuilder.Append("<div " + this.GetControlIdAttribute + dataBindAttribute.ToString() +
"></div>");
return stringBuilder.ToString();
}
private void AddJsFiles()
{
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["kendo.dataviz.chart"]))
{
this.TopLevelForm.RuntimeJS.Add("kendo.dataviz.chart", "js/kendo.dataviz.chart.min.js");
}
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["knockout-kendo.kendochart"]))
{
this.TopLevelForm.RuntimeJS.Add("knockout-kendo.kendochart", "js/knockout-
kendo.kendochart.js");
}
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["corp.customcontrol.barchart"]))
{
this.TopLevelForm.RuntimeJS.Add("corp.customcontrol.barchart", "js/
corp.customcontrol.barchart.js");
}
}
public override string GetViewModelScript()
{
StringBuilder viewModelScript = new StringBuilder();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
viewModelScript.Append(base.GetViewModelScript());
string controlString = "_" + this.ParentForm.Id + "." + this.Id;
viewModelScript.AppendLine("corp.customcontrol.barChart.initialize(" + controlString +
");");
return viewModelScript.ToString();
}
public override IBindingProperty GetBindingClassObject(string fullyQualifiedClassPath)
{
try
{
return (IBindingProperty)Activator.CreateInstance(Type.GetType(fullyQualifiedClassPath));
}
catch (Exception ex)
{
return null;
}
}
}
}
```

BarChartBindingProperty.cs

```
using System.Text;
using Skelta.Forms.Core.CommonObjects;
using Skelta.Forms.Core.Interfaces;
namespace Corp.CustomControls.BarChart
{
public class BarChartBindingProperty : IBindingProperty
{
public string RenderBinding(object bindingHelper)
{
StringBuilder stringBuilder = new StringBuilder();
BindingHelper bindingHelperObject = (BindingHelper)bindingHelper;
string chartControl = bindingHelperObject.ViewModelContainerName + "." +
bindingHelperObject.Control.Id;
stringBuilder.AppendLine(bindingHelperObject.ViewModelContainerName + "." +
bindingHelperObject.Control.Id + "._items = ko.observableArray([]);");
stringBuilder.AppendLine(bindingHelperObject.ViewModelContainerName + "." +
bindingHelperObject.Control.Id + "._series = ko.observableArray([]);");
stringBuilder.AppendLine(bindingHelperObject.ViewModelContainerName + "." +
bindingHelperObject.Control.Id + "._colors = ko.observableArray([]);");
return stringBuilder.ToString();
}
}
}
```

knockout-kendo.kendochart.js

```
(function ()
{
//library is in a closure, use this private variable to reduce size of minified file
var createBinding = ko.kendo.bindingFactory.createBinding.bind(ko.kendo.bindingFactory);
//use constants to ensure consistency and to help reduce minified file size
var CLICK = "click",
CENTER = "center",
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
CHECK = "check",
CHECKED = "checked",
CLICKED = "clicked",
CLOSE = "close",
COLLAPSE = "collapse",
CONTENT = "content",
DATA = "data",
DATE = "date",
DISABLE = "disable",
ENABLE = "enable",
EXPAND = "expand",
ENABLED = "enabled",
EXPANDED = "expanded",
ERROR = "error",
FILTER = "filter",
HIDE = "hide",
INFO = "info",
ISOPEN = "isOpen",
ITEMS = "items",
MAX = "max",
MIN = "min",
OPEN = "open",
PALETTE = "palette",
READONLY = "readonly",
RESIZE = "resize",
SCROLLTO = "scrollTo",
SEARCH = "search",
SELECT = "select",
SELECTED = "selected",
SELECTEDINDEX = "selectedIndex",
SHOW = "show",
SIZE = "size",
SUCCESS = "success",
TARGET = "target",
TITLE = "title",
VALUE = "value",
VALUES = "values",
WARNING = "warning",
ZOOM = "zoom",
DISPLAYTOOLBAR = "displaytoolbar";
createBinding({
name: "kendoChart",
watch: {
data: function (value)
{
ko.kendo.setDataSource(this, value);
}
}
});
}());
```

corp.customcontrol.barchart.js

```
(function ()
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//// Defining Namespaces
window.corp = window.corp || {};
corp.customcontrol = corp.customcontrol || {};
corp.customcontrol.barChart = corp.customcontrol.barChart || {};
corp.customcontrol.barChart = (function ()
{
//// Initializes Bar Chart Control
function initializeBarChart(barChartControl)
{
barChartControl._items(getItems(barChartControl));
barChartControl._series(getSeries(barChartControl));
barChartControl._colors(getColors(barChartControl));
}
function getItems(barChartControl)
{
if (barChartControl._itemsURL === "")
{
return eval(barChartControl.itemsData);
}
var items;
$.ajax({
type: "GET",
async: false,
contentType: "application/json",
url: barChartControl._itemsURL,
dataType: "json",
success: function (data)
{
if (data)
{
items = data;
}
},
error: function (data)
{
alert(data.status + ' ' + data.statusText + ' ' + data.responseText);
}
});
return items;
}
function getSeries(barChartControl)
{
if (barChartControl._seriesURL === "")
{
return eval(barChartControl.seriesData);
}
var series;
$.ajax({
type: "GET",
async: false,
contentType: "application/json",
url: barChartControl._seriesURL,
dataType: "json",
success: function (data)
{
if (data)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
series = data;
}
},
error: function (data)
{
alert(data.status + ' ' + data.statusText + ' ' + data.responseText);
}
});
return series;
}
function getColors(barChartControl)
{
if (barChartControl._colorsURL === "")
{
return eval(barChartControl.colorsData);
}
var colors;
$.ajax({
type: "GET",
async: false,
contentType: "application/json",
url: barChartControl._colorsURL,
dataType: "json",
success: function (data)
{
if (data)
{
colors = data;
}
},
error: function (data)
{
alert(data.status + ' ' + data.statusText + ' ' + data.responseText);
}
});
return colors;
}
//// Public methods
return {
initialize: initializeBarChart
};
})();
})();
```

Add Barchart Properties for Check Syntax Validation

The following code represents how to add the "Barchart" properties along with the other common properties for check syntax validation.

```
(function ()
{
var standardControlMembers = Skelta.forms.constants.controlsCollection;
var genericControlFunctions =
Skelta.forms.constants.controlFunctions;//genericControlFunctions has properties -
nonContainerFunctions and containerFunctions which is a collection
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
of function names [{ name: "functionName"}]
var barchartMembers = {
functions: genericControlFunctions.nonContainerFunctions,//.concat([{ name: "functionName"
}
isContainerControl: false,
controlType: "Barchart",
properties: [{ propertyName: "xmlNodeBoundTo", propertyType: "2", isWritable: false },
{ propertyName: "tagName", propertyType: "0", isWritable: true },
{ propertyName: "id", propertyType: "2", isWritable: false },
{ propertyName: "value", propertyType: "0", isWritable: true },
{ propertyName: "isMandatory", propertyType: "0", isWritable: true },
{ propertyName: "readOnly", propertyType: "0", isWritable: true },
{ propertyName: "enable", propertyType: "0", isWritable: true },
{ propertyName: "itemsData", propertyType: "2", isWritable: true },
{ propertyName: "seriesData", propertyType: "2", isWritable: true },
{ propertyName: "colorsData", propertyType: "2", isWritable: true }]
}
standardControlMembers.push(barchartMembers);
})();
```

In the example, the "itemsData", "seriesData", and "colorsData" properties are nonobservable variables, indicated by **propertyType: "2"** and **isWritable: true** respectively in the above code.

For more information on the usage of the above code, see Adding Custom Controls for Check Syntax Validation.

Example: RichTextBox Control

A custom **RichTextBox** control using Kendo UI widget with control initialization, class, and JavaScript files.

- SFControl.cs (control initialization file)
- RichTextBox.cs (control class file)
- knockout-kendo.editor.js (JavaScript file for Knockout binding of the specific Kendo UI widget)
- richtextbox.js (JavaScript file for custom Knockout binding to support height, width, and read-only behavior)

Selecting the JavaScript files for the specific Kendo UI widget

The included **kendo.all.min.js** is limited to widgets – **Grid**, **DateTimePicker**, **DatePicker**,**TimePicker**, **NumericTextBox**, **ColorPicker**, **Editor**, **Treeview**, **Gauge** and PDF export features – **pdf** and **drawing** (which is built using the **gulp** command) for performance benefits.

To support additional widgets and features, you must refer only to the JavaScript files for the specific Kendo UI widget other than the JavaScript files for the included widgets. For this, you must be aware of script dependencies and requirements for the Kendo UI widgets. Refer to Include Individual Widget Scripts for Kendo for the complete list of JavaScript files required for each Kendo UI widget.

Ensure you do not build the required Kendo UI widget JavaScript files using the **gulp** command as the script files created and registered will contain duplicate code parts.

SFControl.cs

```
using Corp.CustomControls.BarChartCustom;
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.External;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Skelta.Forms.ExternalControls
{
public class SFControls : IExternalControls
{
BaseControl IExternalControls.GetControl(string controlFullName)
{
switch (controlFullName)
{
case "Corp.CustomControls.BarChartCustom.BarChart":
return new BarChart();
default:
return null;
}
}
object[] IExternalControls.GetControls()
{
return new object[] { new BarChart() };
}
}
}
```

RichTextBox.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Skelta.Forms.Core;
using Skelta.Forms.Core.External;
using Skelta.Forms.Core.CommonObjects;
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.Interfaces;
using System.Globalization;
using System.Web;
namespace Kendo.Custom.Controls
{
[DesignerControl(992201, CoreDesigner.NextGenDesigner, DesignerCategory.Custom)]
[SuppressedPropertiesNextGen("DefaultValue;ReadOnly;EnableResponsiveView;ToolTip;OnDataChan
gedScript;IncludeInSummaryView;ExpValidate;ExpValue;MessageWhenEmpty;StyleSheet;ExpDefaultV
alue;ExpVisible;ExpMandatory;")]
public class RichTextBox : BaseDataExternalControl
{
public override BaseControl GetNewInstanceForClone()
{
RichTextBox richTextBox = new RichTextBox();
richTextBox.Enabled = this.Enabled;
richTextBox.Visible = this.Visible;
richTextBox.ToolTip = this.ToolTip;
richTextBox.Height = this.Height;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
richTextBox.Width = this.Width;
return richTextBox;
}
/// <summary>
/// Gets the Display Name
/// </summary>
public static string TypeDisplayName
{
get
{
return "RichTextBox";
}
}
public override string TypeAssociation
{
get
{
return "Custom";
}
}
bool toolbarVisible = true;
/// <summary>
/// Gets/Sets the Rows
/// </summary>
[DesignerProperty(PropertyOrder.ImagePath, Categories.Appearance,
CoreDesigner.NextGenDesigner)]
[BindingAttribute(882940, "Skelta.Forms.Core.CommonObjects.CommonBoolBindingProperty")]
public bool ToolbarVisible
{
get { return toolbarVisible; }
set { toolbarVisible = value; }
}
/// <summary>
/// Adding JS & CSS and setting view model value
/// </summary>
/// <returns></returns>
public override string GetView()
{
AddJsFiles();
return base.GetView();
}
/// <summary>
/// Method to render the control, How it has to look in designer as well as in the preview.
/// </summary>
/// <returns>Controls view </returns>
public override string GetControlHtml()
{
var stringBuilder = new StringBuilder();
stringBuilder.Append("<textarea " + GetControlIdAttribute + " data-bind=\"kendoEditor: {
value: " + this.Id + "._value, ");
stringBuilder.Append(GetTools());
stringBuilder.Append("}, ");
stringBuilder.Append(" richTextBoxReadOnly: " + this.Id + "._enable, richTextBoxHeight: " +
this.Id + "._height, richTextBoxWidth: " + this.Id + "._width \"");
if (this.TopLevelForm.InDesignMode)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
int defaultHeight = 200;
int defaultWidth = 400;
if (this.Height > 0)
{
defaultHeight = this.Height;
}
if (this.Width > 0)
{
defaultWidth = this.Width;
}
string style = " style='height: " + defaultHeight.ToString() + "px; width: " +
defaultWidth.ToString() + "px;'";
stringBuilder.Append(style);
}
stringBuilder.Append("> </textarea>");
return stringBuilder.ToString();
}
private string GetTools()
{
StringBuilder readOnlyBinding = new StringBuilder();
readOnlyBinding.Append("tools: [");
if (this.toolbarVisible)
{
readOnlyBinding.Append("'bold',");
readOnlyBinding.Append("'italic',");
readOnlyBinding.Append("'underline',");
readOnlyBinding.Append("'strikethrough',");
readOnlyBinding.Append("'justifyLeft',");
readOnlyBinding.Append("'justifyCenter',");
readOnlyBinding.Append("'justifyRight',");
readOnlyBinding.Append("'justifyFull',");
readOnlyBinding.Append("'insertUnorderedList',");
readOnlyBinding.Append("'insertOrderedList',");
readOnlyBinding.Append("'indent',");
readOnlyBinding.Append("'outdent',");
readOnlyBinding.Append("'createLink',");
readOnlyBinding.Append("'unlink',");
readOnlyBinding.Append("'insertImage',");
readOnlyBinding.Append("'subscript',");
readOnlyBinding.Append("'superscript',");
readOnlyBinding.Append("'createTable',");
readOnlyBinding.Append("'addRowAbove',");
readOnlyBinding.Append("'addRowBelow',");
readOnlyBinding.Append("'addColumnLeft',");
readOnlyBinding.Append("'addColumnRight',");
readOnlyBinding.Append("'deleteRow',");
readOnlyBinding.Append("'deleteColumn',");
readOnlyBinding.Append("'viewHtml',");
readOnlyBinding.Append("'formatting',");
readOnlyBinding.Append("'fontName',");
readOnlyBinding.Append("'fontSize',");
readOnlyBinding.Append("'foreColor',");
readOnlyBinding.Append("'backColor',");
readOnlyBinding.Append("'print'");
}
readOnlyBinding.Append("]");
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
return readOnlyBinding.ToString();
}
/// <summary>
/// Method to add the custom Js files associated to the custom control
/// </summary>
private void AddJsFiles()
{
// 6. kendo.combobox.min.js
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["kendo.combobox.min.js"]))
{
// "this.TopLevelForm.RuntimeCss" is the collection which holds your custom CSS files.
// It is a key-value pair collection where the key is the file name and the value is the
file path(absolute or relative).
//Important point is the JS file should be present inside "BPMUITemplate/Default/
NextGenForms/js" folder or else it won't get rendered.
this.TopLevelForm.RuntimeJS.Add("kendo.combobox.min.js", "js/kendo.combobox.min.js");
}
//// 4. kendo.editor.min.js
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["kendo.editor.min.js"]))
{
// "this.TopLevelForm.RuntimeCss" is the collection which holds your custom CSS files.
// It is a key-value pair collection where the key is the file name and the value is the
file path(absolute or relative).
//Important point is the JS file should be present inside "BPMUITemplate/Default/
NextGenForms/js" folder or else it won't get rendered.
this.TopLevelForm.RuntimeJS.Add("kendo.editor.min.js", "js/kendo.editor.min.js");
}
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["knockout.kendo.editor.js"]))
{
// "this.TopLevelForm.RuntimeCss" is the collection which holds your custom CSS files.
// It is a key-value pair collection where the key is the file name and the value is the
file path(absolute or relative).
//Important point is the JS file should be present inside "BPMUITemplate/Default/
NextGenForms/js" folder or else it won't get rendered.
this.TopLevelForm.RuntimeJS.Add("knockout.kendo.editor.js", "js/knockout.kendo.editor.js");
}
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["richtextbox.js"]))
{
// "this.TopLevelForm.RuntimeCss" is the collection which holds your custom CSS files.
// It is a key-value pair collection where the key is the file name and the value is the
file path(absolute or relative).
//Important point is the JS file should be present inside "BPMUITemplate/Default/
NextGenForms/js" folder or else it won't get rendered.
this.TopLevelForm.RuntimeJS.Add("richtextbox.js", "js/richtextbox.js");
}
}
}
}
```

knockout-kendo.editor.js

```
// Knockout binding for Kendo editor
(function ()
{
var createBinding = ko.kendo.bindingFactory.createBinding.bind(ko.kendo.bindingFactory);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//use constants to ensure consistency and to help reduce minified file size
var ENABLE = "enable"
VALUE = "value";
// Knockout Kendo Editor
createBinding({
name: "kendoEditor",
defaultOption: VALUE,
events: {
change: VALUE
},
watch: {
enabled: ENABLE,
value: VALUE
}
});
}());
```

richtextbox.js

```
// Knockout custom binding for Kendo editor
// Supports Height, Width and Read-Only behaviour.
ko.bindingHandlers.richTextBoxReadOnly = {
update: function (element, valueAccessor)
{
var value = ko.utils.unwrapObservable(valueAccessor());
var editorBody = $(element).parent().find("iframe")[0].contentDocument.body;
$(editorBody).attr("contenteditable", value);
if (value === true)
{
$(element).parent().parent().parent().find(".k-editor-toolbar").show();
}
else
{
$(element).parent().parent().parent().find(".k-editor-toolbar").hide();
}
}
};
ko.bindingHandlers.richTextBoxHeight = {
update: function (element, valueAccessor)
{
var value = ko.utils.unwrapObservable(valueAccessor());
if (value > 0)
{
$(element).parent().parent().parent().find(".k-content").css("height", value + "px");
$(element).parent().parent().parent().parent().parent().find(".k-editor").css("height",
value + "px");
}
}
};
ko.bindingHandlers.richTextBoxWidth = {
update: function (element, valueAccessor)
{
var value = ko.utils.unwrapObservable(valueAccessor());
if (value > 0)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
$(element).parent().parent().parent().parent().parent().find(".k-editor").css("width",
value + "px");
}
}
};
```

Custom Controls with Angular 12

You can create a custom control with Angular 12 to design a control with specific or custom properties of Angular 12.

Create Custom Controls with Angular 12

Create a custom control with Angular 12 to design a control with specific or custom properties of Angular 12.

To create a custom control with Angular 12

1. Create a DLL project (**Visual C# > Windows > Class Library**) using Visual Studio.NET.
2. Create or add two .cs files – **SFControls.cs** (control initialization file) and **MyCustom.cs** (public control class file) to your project.
3. Add a reference to the **Skelta.Forms.Core.dll** (located at **<Skelta BPM Install Path>\Bin** folder) in your project.
4. Add the following namespaces to your **MyCustom.cs** file.

   ```
   using Skelta.Forms.Core;
   using Skelta.Forms.Core.External;
   using Skelta.Forms.Core.CommonObjects;
   using Skelta.Forms.Core.Controls;
   using Skelta.Forms.Core.Interfaces;
   ```
5. Ensure the control class (**MyCustom.cs**) inherits from the **BaseDataExternalControl** class.
6. Write the control definition code with the following in mind:
   a. Ensure the **Core Designer** is set to **Next Generation Designer** and the **Designer Category** is set to **Custom** to show up the control in the toolbox of the **Forms Designer**.

      ```
      [DesignerControl(992200, CoreDesigner.NextGenDesigner, DesignerCategory.Custom)]
      public class MyControl : BaseDataExternalControl
      ```
   b. Suppress properties, if required, using the attribute **SuppressedPropertiesNextGen**.

      ```
      // Suppress properites PropertyName1, PropertyName2, and PropertyName3.
      [SuppressedPropertiesNextGen("PropertyName1;PropertyName2;PropertyName3;")]
      ```
   c. Override the **GetNewInstanceForClone()** function, create an instance of the control, and assign the basic properties of the control.
   d. Override the **GetControlHtml()** function and define the HTML code. Use the **TopLevelForm.InDesignMode** boolean variable to segregate the designer and run-time views, if required. Ensure all view model properties are specific to the instance defined in the view.

      For example:

      ```
      StringBuilder stringBuilder = new StringBuilder();

      if (this.TopLevelForm.InDesignMode)
      ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
stringBuilder.Append("<div class=\"dropDownDesignClass\"></div>");
}
else
{
stringBuilder.Append("<drop-down" + this.UniqueControlId() + " controlid= \"" +
this.Id + "\" data-bind=\"" + CustomBindings.AngularInstanceId + ":
skelta.forms.utilities.getGuId()\"></drop-down" + this.UniqueControlId() + ">");
}
return stringBuilder.ToString();
```

Where **"drop-down" + this.UniqueControlId()** selector and **controlid** are the custom attributes. And dropDownDesignClass is a css class defined to show the UI/UX at design time.

e. Override the **GetViewModelDefinition()** function (if required) to define the **ViewModel**. Define a function and subscribe to **isFormLoaded** event of **topLevelForm**.

For example, the **ViewModel** can be built as:

```
var viewModelDefinitionBuilder = new StringBuilder();
var ctr = "_" + this.ParentForm.Id + "." + this.Id;
var javascriptFileInclusions = "['/inline.bundle.js', '/polyfills.bundle.js',
'/styles.bundle.js', '/vendor.bundle.js', '/main.bundle.js']";
viewModelDefinitionBuilder.Append(base.GetViewModelDefinition());
viewModelDefinitionBuilder.Append(ctr + ".initialize" + " = function()");
viewModelDefinitionBuilder.Append("{");
viewModelDefinitionBuilder.Append("window.controlInstance = " + ctr + ";");
viewModelDefinitionBuilder.Append(@"skelta.forms.utilities.includeCustomJsFiles(" +
javascriptFileInclusions + ", true, true);");
viewModelDefinitionBuilder.Append("};");
if (!string.IsNullOrEmpty(this.OnInitializeScript))
{
viewModelDefinitionBuilder.AppendLine(ctr + ".onInit = function(control) {");
viewModelDefinitionBuilder.AppendLine(this.OnInitializeScript);
viewModelDefinitionBuilder.AppendLine("}");
}
return viewModelDefinitionBuilder.ToString();
```

Note: Ensure the included JS files are present in the **<Installpath>\Web\BPMUITemplates\Default\NextGenForms\js** folder.

a. Define rules for a custom control to validate the data using the client-side JavaScript, if required. For more information, see Rules and Defining Rules.

Create class properties as control properties.

```
//Designer Property
[DesignerProperty(773000, Categories.Validation, CoreDesigner.NextGenDesigner)]
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//Binding Property
[BindingAttribute(882995, "Skelta.Forms.Core.CommonObjects.IntegerBindingProperty")]
```

7. Create **SFControls.cs** class under **Skelta.Forms.ExternalControls** namespace.
8. Add the following namespaces to your **SFControls.cs** file.

```
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.External;
using MyCustomControls; // Control namespace
```

9. Implement the **IExternalControls** in the **SFControls** class.
   a. In the **IExternalControls.GetControl(string controlFullName)** method, return the new control instance based on the control namespace.
   b. In the **GetControls** method, return a new object array which contains all the control instances.
10. Build your project, create a **SOA** folder on your hard drive (for example, **C:\SOA**), place your custom control DLL file in the folder, and then register the location of the folder with the **Skelta SOA Folder List**.
11. Register the custom control DLL file through the **Form Controls** to display the custom control in the controls or custom category. For more information, see Custom Categories.

Note:
- Control initialization file must always be named as **SFControls.cs** and the class name must always be named as **SFControls**.
- Ensure reference to the **Skelta.Forms.Core.dll** assembly is the same as the assembly on the AVEVA Work Tasks server.
- For information about SOA, see Object Access in the *User Guide*.
-Ensure all Bundled JS files are present in the **<Installpath>\Web\BPMUITemplates\Default\NextGenForms\js** folder.

To localize the text for the custom control

- See Localization.

To add an icon to a custom control

1. Add a custom icon in the **<Install Path>\Web\BPMUITemplates\Default\Themes\<ThemeName>\NextGenForms\images** folder.
2. Open the **SkeltaCustomControlDesigner.css** file in the **<Install Path>\Web\BPMUITemplates\Default\Themes\<ThemeName>\NextGenForms\css\custom** folder.
3. Add the following style to the **SkeltaCustomControlDesigner.css** file.

```
/* Replace <custom_icon> with the file name for the normal icon. */
/* Replace <namespace.classname> with the custom control namespace and class name. */
.sktbci[data-skttp="<namespace.classname>"]
{
background-image: url(../../images/<custom_icon>.png);
}
/* Replace <custom_icon_hover> with the file name for the hover icon. */
.sktbc:hover > .sktbci[data-skttp="<namespace.classname>"]
{
background-image: url(../../images/<custom_icon_hover>.png);
}
```

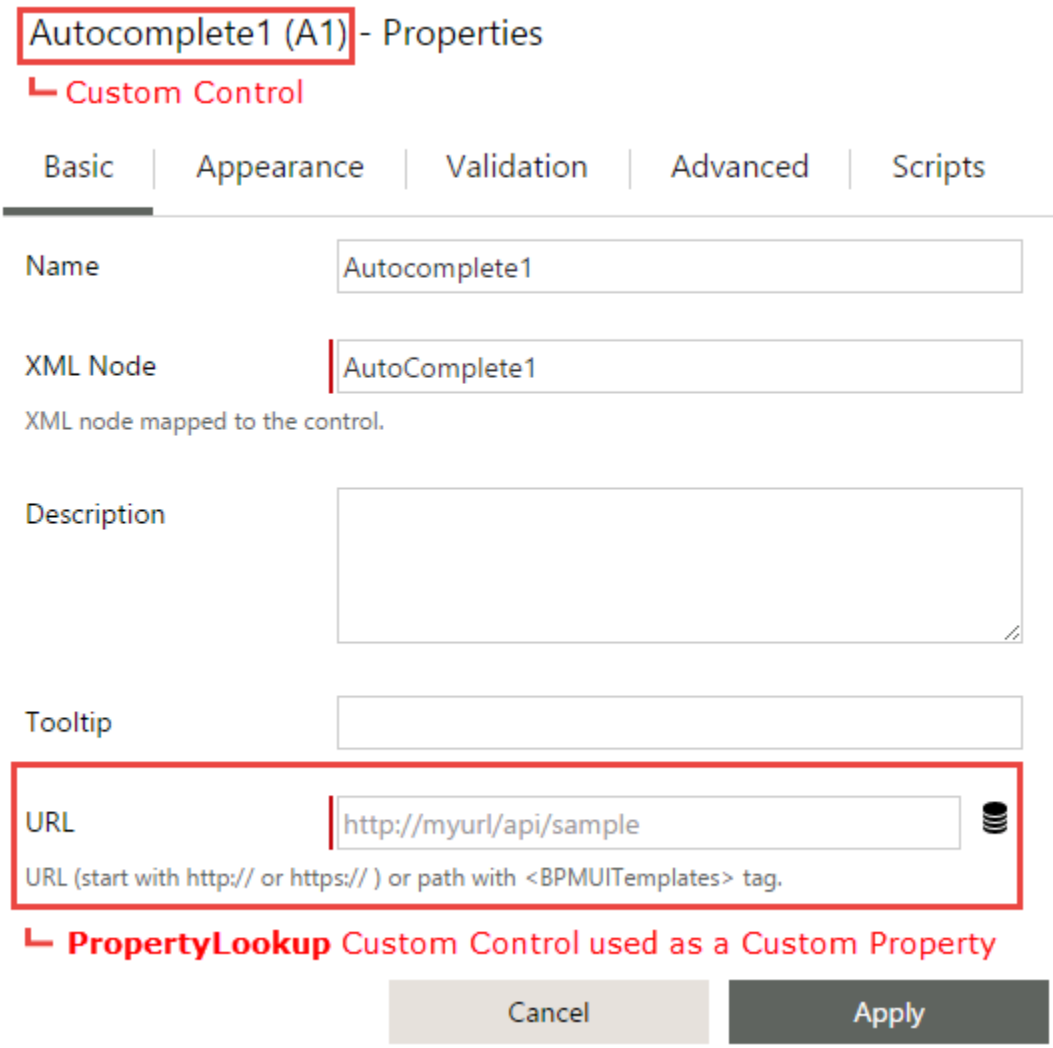
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Save the **SkeltaCustomControlDesigner.css** file.
5. Reload the **Forms Designer** page to see the custom control with the custom icon in the toolbox.

Custom Controls with Custom Properties

You can create a custom control with a custom property (which is actually a custom control).

Here the **Autocomplete** custom control makes use of the **PropertyLookup** custom control as its custom property (**URL**).



1. Create a custom control (**PropertyLookup**) to take inputs from the secondary page.

   The **PropertyLookup** control is an internal control, which is a control whose **Designer Category** is not set and therefore does not appear in the **Toolbox** of the **Forms Designer**.

   The **PropertyLookup** control is a read-only text box with an icon on the right and two properties:

   – **ImageClass** for showing an icon on the right

   – **ClickFunction** for handling the icon click event

   The **PropertyLookup.css** CSS file implements the **ImageClass** property and the **PropertyLookup.js** JavaScript

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

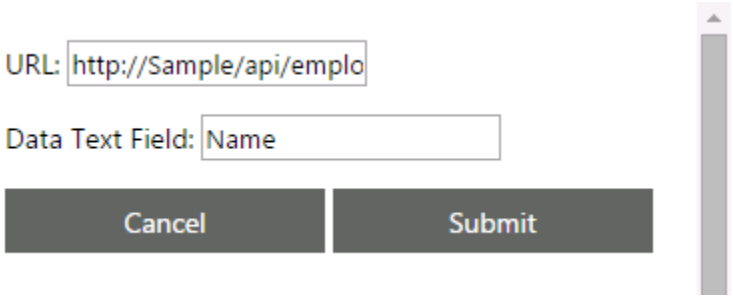
file implements the **ClickFunction** property.

2. Create another custom control (**Autocomplete**) and use the first custom control (**PropertyLookup**) as a custom property (**URL**).
   a. To use the **PropertyLookup** control as a property (**URL**) for the **Autocomplete** control, code in the **Autocomplete.cs** control initialization file as follows:

   ```
   [DesignerProperty(376567, Categories.Behaviour, typeof(PopupProperty), "")]
   [Binding(883100, "Skelta.Forms.Core.CommonObjects.CommonStringBindingProperty")]
   public string URL
   {
   get;
   set;
   }
   ```

   Here the **PopupProperty** refers to the **PopupProperty.cs** control initialization file, which renders the property (**URL**) for the **Autocomplete** control using the **IControlProperty** interface.
   b. To use an inline frame (iframe) in the property page, code in the **PropertyLookup.js** JavaScript file as follows:

   ```
   // Here PropertyLookup.htm is placed in the NextGen folder.
   parent.document.getElementById("propertyLookupFrame").src = "PropertyLookup.htm";
   parent.document.getElementById("propertyLookupFrame").attributes["data-skfsw"].value
   = 1;
   ```

   Here the **propertyLookupFrame** renders the inline frame using the **PropertyLookup.htm** HTML file. The `attributes["data-skfsw"].value = 1` ensures the inline frame is shown on top of the forms control property page.
   c. To access the **editPage View Model**, code in the **PropertyLookup.htm** HTML file as follows:

   ```
   var editPageVm = window.parent.frames["editframe"].viewModelObject ?
   window.parent.frames["editframe"].viewModelObject :
   window.parent.frames["editframe"].contentWindow.viewModelObject;
   ```

   d. To close the inline frame on click event, code in the **PropertyLookup.htm** HTML file as follows:

   ```
   parent.document.getElementById("propertyLookupFrame").attributes["data-skfsw"].value
   = 0;
   parent.document.getElementById("propertyLookupFrame").attributes["src"].value = "";
   ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example: PropertyLookup Control

A custom **PropertyLookup** control with control initialization, class, JavaScript, CSS, and HTM files.

- SFControl.cs (control initialization file)
- PropertyLookup.cs (control class file)
- PropertyLookup.js (JavaScript file)
- PropertyLookup.css (CSS file)

SFControl.cs

```
using Custom.Forms.Controls;
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.External;
namespace Skelta.Forms.ExternalControls
{
class SFControls : IExternalControls
{
#region IExternalControls Members
BaseControl IExternalControls.GetControl(string controlFullName)
{
switch (controlFullName)
{
case "Custom.Forms.Controls.PropertyLookup":
return new PropertyLookup();
default:
return null;
}
}
object[] IExternalControls.GetControls()
{
return new object[] { new PropertyLookup() };
}
#endregion
}
}
```

PropertyLookup.cs

```
using Skelta.Forms.Core;
using Skelta.Forms.Core.Controls;
namespace Custom.Forms.Controls
{
using System.Collections.Generic;
using System.Text;
/// <summary>
/// Lookup control for property window
/// </summary>
////[DesignerControl(992202, CoreDesigner.NextGenDesigner, DesignerCategory.Custom)]
[SuppressedPropertiesNextGen("DefaultValue;EnableResponsiveView")]
public class PropertyLookup : BaseDataExternalControl
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
public override BaseControl GetNewInstanceForClone()
{
var propertyLookup = new PropertyLookup
{
ImageClass = this.ImageClass,
ClickFunction = this.ClickFunction,
EnableDisplayValueBinding = this.EnableDisplayValueBinding
};
return propertyLookup;
}
/// <summary>
/// Get the Display name
/// </summary>
public static string TypeDisplayName
{
get
{
return "PropertyLookup";
}
}
public override string TypeAssociation
{
get
{
return "Custom";
}
}
/// <summary>
/// Get or set the Image name
/// </summary>
public string ImageClass
{
get;
set;
}
/// <summary>
/// Get or set the Click Function name
/// </summary>
public string ClickFunction
{
get;
set;
}
/// <summary>
/// Get or set the Property Lookup binding event
/// </summary>
public string PropertyLookupEvents
{
get;
set;
}
/// <summary>
/// Get or set a value indicating whether the control requires a Display Value binding
property
/// </summary>
public bool EnableDisplayValueBinding
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
get;
set;
}
/// <summary>
/// Get a value indicating whether the control requires Display Value binding
/// </summary>
public override bool RequiresDisplayValueBinding
{
get
{
return this.EnableDisplayValueBinding;
}
}
/// <summary>
/// Set the Display Value handler
/// </summary>
public override string SetDisplayValueHandlerMethod
{
get
{
return Common.ViewUtilitiesVarName + ".updateDisplayValue;";
}
}
public override string GetViewModelDefinition()
{
var control = "_" + this.ParentForm.Id + "." + this.Id;
var stringBuilder = new StringBuilder();
stringBuilder.Append(base.GetViewModelDefinition());
stringBuilder.AppendLine(control + ".clickFunction=function (d, e) {propertyLookup." +
this.ClickFunction + "(d, e," + control + ");};");
return stringBuilder.ToString();
}
/// <summary>
/// Get the control structure or the view
/// </summary>
/// <returns>The control view</returns>
public override string GetControlHtml()
{
AddCssFiles();
AddJsFiles();
var stringBuilder = new StringBuilder();
stringBuilder.AppendLine("<div class=\"" + CssClasses.EventLookup + "\">");
stringBuilder.AppendLine("<div class=\"" + CssClasses.FlexContent + "\">");
stringBuilder.Append("<input type='text' id='" + this.Id + "' readonly ='readonly'");
stringBuilder.Append(this.DataBindProperties());
stringBuilder.Append("class='" + CssClasses.FlexContent + " " + CssClasses.Inputs + "' >");
stringBuilder.AppendLine("</div>");
stringBuilder.AppendLine("<div>");
stringBuilder.AppendLine("<div class='" + this.ImageClass + "' " +
BaseDataControl.DataBindAttribute + "='click:" + this.Id + ".clickFunction,
clickBubble:false' ></div>");
stringBuilder.AppendLine("</div>");
stringBuilder.AppendLine("</div>");
return stringBuilder.ToString();
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
private void AddJsFiles()
{
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["PropertyLookup.js"]))
{
// "this.TopLevelForm.RuntimeJS" is the collection which holds your custom JS files.
// It is a key-value pair collection where the key is the file name and the value is the
file path(absolute or relative).
// Ensure the JS files are present in the "BPMUITemplates\Default\NextGenForms\js" folder.
Otherwise, it will not be rendered.
this.TopLevelForm.RuntimeJS.Add("PropertyLookup.js", "js/PropertyLookup.js");
}
}
/// <summary>
/// Method to add the CSS files associated with the custom control
/// </summary>
private void AddCssFiles()
{
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeCss["PropertyLookup.css"]))
{
this.TopLevelForm.RuntimeCss.Add("PropertyLookup.css", "css/PropertyLookup.css");
}
}
/// <summary>
/// Data bind properties of property lookup input field
/// </summary>
/// <returns>Data bind attribute value</returns>
private string DataBindProperties()
{
var dataBindProperties = new List<string>
{
"value:" + this.Id + (this.RequiresDisplayValueBinding ? "._displayValue" : "._value"),
"mandatory:" + this.Id + "._isMandatory",
"hasfocus:" + this.Id + "._lostFocus"
};
return Common.GetDataBindAttributeString(dataBindProperties.ToArray());
}
}
}
```

PropertyLookup.js

```
var PropertyLookup = (function ()
{
var propertyLookupInstance;
var createPropertylookup = function ()
{
var propertyClick = function (d, e, control)
{
parent.document.getElementById("propertyLookupFrame").src = "PropertyLookup.htm";
parent.document.getElementById("propertyLookupFrame").attributes["data-skfsw"].value = 1;
};
return {
onIconClick: propertyClick
};
};
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
return {
getInstance: function ()
{
if (!propertyLookupInstance)
{
propertyLookupInstance = createPropertylookup();
}
return propertyLookupInstance;
}
};
})();
var propertyLookup = PropertyLookup.getInstance();
```

PropertyLookup.css

```
.sknpl
{
margin-left: 6px;
width: 25px;
height: 25px;
background-image: url(../images/NGF-Sprites-DesignTime-img.png);
background-repeat: no-repeat;
background-position: -210px -1365px;
}
.sknpl:hover
{
background-image: url(../images/NGF-Sprites-DesignTime-img.png);
background-position: -210px -1408px;
background-color: #000000;
}
```

Example: Autocomplete Control

A custom **Autocomplete** control with control initialization, class, JavaScript, XML and HTM files.

- SFControl.cs (control initialization file)
- Autocomplete.cs (control class file)
- GridProperty.cs (control class file)
- PopupProperty.cs (control class file)
- Autocomplete.js (JavaScript file)
- knockout-kendo.autocompletebinding.js (JavaScript file for Knockout binding of the specific Kendo UI widget)
- BaseFormWithValidationScript.xml (XML file)
- PropertyLookup.htm (HTM file for the inline frame)

SFControl.cs

```
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.External;
using System;
using System.Collections.Generic;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Kendo.Custom.Controls;
namespace Skelta.Forms.ExternalControls
{
class SFControls : IExternalControls
{
#region IExternalControls Members
BaseControl IExternalControls.GetControl(string controlFullName)
{
switch (controlFullName)
{
case "Kendo.Custom.Controls.AutoComplete":
return new AutoComplete();
case "Kendo.Custom.Controls.CustomPanel":
return new CustomPanel();
case "Kendo.Custom.Controls.CustomGrid":
return new CustomGrid();
case "Kendo.Custom.Controls.CustomAttachment":
return new CustomAttachment();
default:
return null;
}
}
object[] IExternalControls.GetControls()
{
return new object[] { new AutoComplete(), new CustomPanel(), new CustomGrid(), new
CustomAttachment() };
}
#endregion
}
}
```

Autocomplete.cs

```
namespace Kendo.Custom.Controls
{
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Reflection;
using System.Text;
using System.Web;
using System.Xml;
using Kendo.Custom.ControlProperty;
using Skelta.Forms.Core;
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.Interfaces;
[DesignerControl(992201, CoreDesigner.NextGenDesigner, DesignerCategory.Custom)]
[SuppressedPropertiesNextGen("DefaultValue;ReadOnly;EnableResponsiveView")]
public class AutoComplete : BaseDataExternalControl
{
public enum FilterConditionsEnum
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
startswith = 0,
endswith,
contains
}
public override BaseControl GetNewInstanceForClone()
{
var auto = new AutoComplete
{
Enabled = this.Enabled,
Visible = this.Visible,
ToolTip = this.ToolTip,
AutoCompleteLength = this.AutoCompleteLength,
Filter = this.Filter,
HeaderTemplate = this.HeaderTemplate,
DataTextField = this.DataTextField,
MessageWhenEmpty = this.MessageWhenEmpty,
Data = this.Data,
URL = this.URL
};
return auto;
}
/// <summary>
/// Get the Display name
/// </summary>
public static string TypeDisplayName
{
get
{
return "AutoComplete";
}
}
public override string TypeAssociation
{
get
{
return "Custom";
}
}
/// <summary>
/// Set the MinLength for TextInput
/// </summary>
[DesignerProperty(PropertyOrder.MinimumLength, Categories.Validation,
CoreDesigner.NextGenDesigner)]
[Binding(882930, "Skelta.Forms.Core.CommonObjects.IntegerBindingProperty")]
public int AutoCompleteLength
{
get;
set;
}
/// <summary>
/// Get or set the FilterConditionsEnum position
/// </summary>
[Scriptable(550600)]
[DesignerProperty(553000, Categories.Appearance, CoreDesigner.FormDesigner)]
[DesignerProperty(PropertyOrder.LabelPosition + 1, Categories.Appearance,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
CoreDesigner.NextGenDesigner)]
[Binding(883100, "Skelta.Forms.Core.CommonObjects.CommonEnumBindingProperty")]
public FilterConditionsEnum Filter
{
get;
set;
}
/// <summary>
/// Get or set the FilterConditionsEnum position
/// </summary>
[Scriptable(550600)]
[DesignerProperty(376567, Categories.Behaviour, typeof(PopupProperty), "")]
[Binding(883100, "Skelta.Forms.Core.CommonObjects.CommonStringBindingProperty")]
public string URL
{
get;
set;
}
/// <summary>
/// Set the MinLength for TextInput
/// </summary>
[DesignerProperty(PropertyOrder.MinimumLength + 1, Categories.Appearance,
CoreDesigner.NextGenDesigner)]
[Binding(882930, "Skelta.Forms.Core.CommonObjects.CommonStringBindingProperty")]
public string HeaderTemplate
{
get;
set;
}
string _dataTextField = "name";
/// <summary>
/// Set the MinLength for TextInput
/// </summary>
[DesignerProperty(PropertyOrder.MinimumLength + 2, Categories.Appearance,
CoreDesigner.NextGenDesigner)]
[Binding(882940, "Skelta.Forms.Core.CommonObjects.CommonStringBindingProperty")]
public string DataTextField
{
get;
set;
}
public string DataFormXml
{
get
{
// Returns the resource form definition XML present in the assembly
return Helper.GetResourceTextFile("DataXml.xml");
}
}
[DesignerProperty(PropertyOrder.MinimumLength + 2, Categories.Behaviour,
typeof(GridProperty), "DataFormXml")]
public Dictionary<string, string> Data
{
get;
set;
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
public override string SetDisplayValueHandlerMethod
{
get
{
return Common.ViewUtilitiesVarName + ".updateDisplayValue;";
}
}
public override string GetViewModelDefinition()
{
var stringBuilder = new StringBuilder();
stringBuilder.Append(base.GetViewModelDefinition());
if (string.IsNullOrEmpty(this.URL) && this.Data != null && this.Data.Count > 0)
{
var jArray = this.Data.Aggregate("[", (current, eachItem) => current + ("{Name:'" +
eachItem.Value + "',SurName:'" + eachItem.Key + "'},"));
jArray = jArray.TrimEnd(',');
jArray += "]";
stringBuilder.AppendLine("_" + this.ParentForm.Id + "." + this.Id +
".items=ko.observableArray(" + jArray + ");");
}
else
{
stringBuilder.AppendLine("_" + this.ParentForm.Id + "." + this.Id +
".items=autoComplete.getData(_" + this.ParentForm.Id + "." + this.Id + ");");
}
return stringBuilder.ToString();
}
/// <summary>
/// Method to render the control, How it has to look in designer as well as in the preview
/// </summary>
/// <returns>Controls view </returns>
public override string GetControlHtml()
{
var stringBuilder = new StringBuilder();
var dataBindAttribute = new StringBuilder();
var headerTemplate = string.IsNullOrEmpty(this.HeaderTemplate) ? string.Empty : "
headerTemplate:'" + this.HeaderTemplate + "', ";
var dataTextField = string.IsNullOrEmpty(this.DataTextField) ? string.Empty : "
dataTextField:'" + this.DataTextField + "', ";
var placeHolder = string.IsNullOrEmpty(this.MessageWhenEmpty) ? string.Empty : "
placeholder:'" + this.MessageWhenEmpty + "', ";
dataBindAttribute.Append(" " + DataBindAttribute + "=" + "\"kendoAutoComplete : {" +
placeHolder + headerTemplate + dataTextField + "data: " + this.Id + ".items, enabled: " +
this.Id + "._enable, filter:'" + this.Filter + "', readOnly: " + this.Id + "._readOnly,
value: " + this.Id + "._value, minLength:" + this.AutoCompleteLength + "}\"");
stringBuilder.Append("<input " + DataAttributes.Role + "=\"skAutoComplete\" " +
this.GetControlIdAttribute + dataBindAttribute);
stringBuilder.Append(GetCommonViewForControls(this));
stringBuilder.Append("/>");
return stringBuilder.ToString();
}
/// <summary>
/// Method to add the custom JS files associated with the custom control
/// </summary>
private void AddJsFiles()
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["kendo.autocomplete.min.js"]))
{
// "this.TopLevelForm.RuntimeCss" is the collection which holds your custom CSS files.
// It is a key-value pair collection where the key is the file name and the value is the
file path(absolute or relative).
// Ensure the JS files are present in the "BPMUITemplates\Default\NextGenForms\js" folder.
Otherwise, it will not be rendered.
this.TopLevelForm.RuntimeJS.Add("kendo.autocomplete.min.js", "js/
kendo.autocomplete.min.js");
}
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["knockout-
kendo.autocompletebinding.min.js"]))
{
// "this.TopLevelForm.RuntimeCss" is the collection which holds your custom CSS files.
// It is a key-value pair collection where the key is the file name and the value is the
file path(absolute or relative).
// Ensure the JS files are present in the "BPMUITemplates\Default\NextGenForms\js" folder.
Otherwise, it will not be rendered.
this.TopLevelForm.RuntimeJS.Add("knockout-kendo.autocompletebinding.min.js", "js/knockout-
kendo.autocompletebinding.min.js");
}
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["AutoComplete.js"]))
{
this.TopLevelForm.RuntimeJS.Add("AutoComplete.js", "js/AutoComplete.js");
}
}
/// <summary>
/// Adding JS and CSS files and setting the view model value
/// </summary>
/// <returns></returns>
public override string GetView()
{
AddJsFiles();
return base.GetView();
}
public override IBindingProperty GetBindingClassObject(string fullyQualifiedClassPath)
{
try
{
return (IBindingProperty)Activator.CreateInstance(Type.GetType(fullyQualifiedClassPath));
}
catch (Exception ex)
{
return null;
}
}
internal static string GetCommonViewForControls(object control)
{
var stringBuilder = new StringBuilder();
var cultureInfo = CultureInfo.InvariantCulture;
var baseControl = control as BaseControl;
if (baseControl.ToolTip.Length > 0)
{
stringBuilder.Append(" title = \"" + HttpUtility.HtmlEncode(baseControl.ToolTip) + "\" ");
}
if (baseControl.Height > 0 && baseControl.Width > 0)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
stringBuilder.Append(" style=\"height:" + baseControl.Height.ToString(cultureInfo) +
"px;width:" + baseControl.Width.ToString(cultureInfo) + "px;max-width:" +
baseControl.Width.ToString(cultureInfo) + "px;\" ");
}
else
{
if (baseControl.Height > 0)
{
stringBuilder.Append(" style=\"height:" + baseControl.Height.ToString(cultureInfo) + "px;\"
");
}
else if (baseControl.Width > 0)
{
stringBuilder.Append(" style=\"width:" + baseControl.Width.ToString(cultureInfo) + "px;max-
width:" + baseControl.Width.ToString(cultureInfo) + "px;\" ");
}
}
return stringBuilder.ToString();
}
}
}
```

GridProperty.cs

```
using System.Linq;
using Kendo.Custom.Controls;
using Skelta.Forms2.Web;
namespace Kendo.Custom.ControlProperty
{
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Reflection;
using System.Xml;
using System.Xml.XPath;
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.Interfaces;
/// <summary>
/// Class for showing the control property as Grid
/// </summary>
public class GridProperty : IControlProperty
{
/// <summary>
/// Get the grid form control for rendering the property
/// </summary>
/// <param name="mainControl">Main control</param>
/// <param name="propertyInformation">property information</param>
/// <returns>Grid form with 2 text inputs</returns>
public BaseControl GetPropertyControl(BaseControl mainControl, PropertyInfo
propertyInformation)
{
if (propertyInformation == null || mainControl == null)
{
return null;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
var propertyDetails = string.Empty;
var designerProperties = propertyInformation.GetCustomAttributes(typeof(DesignerProperty),
false);
foreach (var designerProperty in
designerProperties.Cast<DesignerProperty>().Where(designerProperty =>
designerProperty.DesignerType.Equals(CoreDesigner.NextGenDesigner)))
{
propertyDetails = designerProperty.ControlPropertyDetails;
break;
}
if (string.IsNullOrEmpty(propertyDetails))
{
propertyDetails = Helper.GetResourceTextFile("BaseFormWithValidationScript.xml");
}
var localized = CommonFunctions.ProcessLocalization(propertyDetails,
CommonFunctions.GetCurrentECCulture);
var baseF = BaseForm.LoadDefinitionXml(localized);
return baseF.Controls.FirstOrDefault();
}
/// <summary>
/// Read the property information from XML document
/// </summary>
/// <param name="propertyInformation">Property information</param>
/// <param name="xmlDocument">Instance xml</param>
/// <param name="propertyValue">Value of the property</param>
public void ReadProperty(PropertyInfo propertyInformation, IXPathNavigable xmlDocument,
object propertyValue)
{
var xmlDoc = xmlDocument as XmlDocument;
var values = propertyValue as Dictionary<string, string>;
if (xmlDoc == null || values == null || propertyInformation == null)
{
return;
}
foreach (var value in values)
{
var xmlElement = xmlDoc.CreateElement(propertyInformation.Name);
var propText = xmlElement.AppendChild(xmlDoc.CreateElement(propertyInformation.Name +
"Text"));
var propValue = xmlElement.AppendChild(xmlDoc.CreateElement(propertyInformation.Name +
"Value"));
propText.InnerText = value.Key;
propValue.InnerText = value.Value;
xmlDoc.FirstChild.AppendChild(xmlElement);
}
}
/// <summary>
/// Set the value to property of the control
/// </summary>
/// <param name="xmlDocument">Instance XML</param>
/// <param name="propertyControl">Control which is rendered for the property</param>
/// <param name="mainControl">Main control</param>
public void WriteProperty(IXPathNavigable xmlDocument, BaseDataControl propertyControl,
BaseControl mainControl)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
var paramterName = string.Empty;
var paramterControl = string.Empty;
var propertyValues = new Dictionary<string, string>();
var grid = propertyControl as BaseForm;
var xmlDoc = xmlDocument as XmlDocument;
if (xmlDoc == null || grid == null || mainControl == null)
{
return;
}
var xmlNodeList = xmlDoc.SelectNodes("//" + propertyControl.XmlNodeBoundTo);
var prop = mainControl.GetType().GetProperty(propertyControl.XmlNodeBoundTo);
if (xmlNodeList != null)
{
for (var i = 0; i < xmlNodeList.Count; i++)
{
foreach (var gridControl in grid.Controls)
{
foreach (XmlNode node in xmlNodeList[i].ChildNodes)
{
var baseDataControl = gridControl as BaseDataControl;
if (IsParameterPartAvailable(baseDataControl, node, prop.Name + "Text"))
{
paramterName = node.InnerText;
break;
}
if (IsParameterPartAvailable(baseDataControl, node, prop.Name + "Value"))
{
paramterControl = node.InnerText;
break;
}
}
}
if (!string.IsNullOrEmpty(paramterName) && !string.IsNullOrEmpty(paramterControl))
{
propertyValues.Add(paramterName, paramterControl);
}
}
}
prop.SetValue(mainControl, Convert.ChangeType(propertyValues, prop.PropertyType,
CultureInfo.InvariantCulture), null);
}
/// <summary>
/// Get parameter value based on the XMLNodeBoundTo
/// </summary>
/// <param name="baseDataControl">Base data control</param>
/// <param name="node">Iterative node</param>
/// <param name="parameterPart">Value for XMLNodeBoundTo</param>
/// <returns>Whether the parameter item is found</returns>
private static bool IsParameterPartAvailable(BaseDataControl baseDataControl, XmlNode node,
string parameterPart)
{
return baseDataControl != null && node.Name == baseDataControl.XmlNodeBoundTo && node.Name
== parameterPart;
}
}
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

PopupProperty.cs

```
using System.Linq;
using Custom.Forms.Controls;
using Skelta.Forms2.Web;
namespace Kendo.Custom.ControlProperty
{
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Reflection;
using System.Xml;
using System.Xml.XPath;
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.Interfaces;
/// <summary>
/// Class for showing the control property as Popup
/// </summary>
public class PopupProperty : IControlProperty
{
/// <summary>
/// Gets the Popup form control for rendering property
/// </summary>
/// <param name="mainControl">Main control</param>
/// <param name="propertyInformation">Property information</param>
/// <returns>Grid form with 2 text inputs</returns>
public BaseControl GetPropertyControl(BaseControl mainControl, PropertyInfo
propertyInformation)
{
return new PropertyLookup
{
Id = "E_SetDataSource",
Name = propertyInformation.Name,
Description = propertyInformation.Name,
IsMandatory = true,
XmlNodeBoundTo = propertyInformation.Name,
ImageClass = "sknpl",
ClickFunction = "onIconClick"
};
}
/// <summary>
/// Read the property information from the XML document
/// </summary>
/// <param name="propertyInformation">Property information</param>
/// <param name="xmlDocument">Instance XML</param>
/// <param name="propertyValue">Value of the property</param>
public void ReadProperty(PropertyInfo propertyInformation, IXPathNavigable xmlDocument,
object propertyValue)
{
var xmlDoc = xmlDocument as XmlDocument;
if (xmlDoc == null || propertyInformation == null || propertyValue == null)
{
return;
}
if (string.IsNullOrEmpty(propertyValue.ToString())) return;
var propElement = xmlDoc.CreateElement(propertyInformation.Name);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
propElement.InnerText = propertyValue.ToString();
xmlDoc.FirstChild.AppendChild(propElement);
}
/// <summary>
/// Set the value to property of the control
/// </summary>
/// <param name="xmlDocument">Instance XML</param>
/// <param name="propertyControl">Control which is rendered for the property</param>
/// <param name="mainControl">Main control</param>
public void WriteProperty(IXPathNavigable xmlDocument, BaseDataControl propertyControl,
BaseControl mainControl)
{
var xmlDoc = xmlDocument as XmlNode;
if (xmlDoc == null || propertyControl == null || mainControl == null)
{
return;
}
var value = string.Empty;
foreach (var item in xmlDoc.FirstChild.ChildNodes.Cast<XmlNode>().Where(item => item.Name
== propertyControl.XmlNodeBoundTo))
{
value = item.InnerText;
}
var prop = mainControl.GetType().GetProperty(propertyControl.XmlNodeBoundTo);
prop.SetValue(mainControl, Convert.ChangeType(value, prop.PropertyType,
CultureInfo.InvariantCulture), null);
}
}
}
```

Autocomplete.js

```
var AutoComplete = (function ()
{
var autoCompleteInstance;
var createAutoComplete = function ()
{
var getData = function (control)
{
var items = ko.observableArray([]);
$.ajax({
type: "GET",
async: false,
url: control.uRL,
success: function (data)
{
items = ko.observableArray(data);
},
error: function (request, status, error)
{
alert(error);
}
});
return items;
};
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
return {
getData: getData,
};
};
return {
getInstance: function ()
{
if (!autoCompleteInstance)
{
autoCompleteInstance = createAutoComplete();
}
return autoCompleteInstance;
}
};
})();
var autoComplete = AutoComplete.getInstance();
```

knockout-kendo.autocompletebinding.js

```
(function ()
{
var createBinding = ko.kendo.bindingFactory.createBinding.bind(ko.kendo.bindingFactory);
//use constants to ensure consistency and to help reduce minified file size
var CLICK = "click",
CLICKED = "clicked",
CLOSE = "close",
COLLAPSE = "collapse",
CONTENT = "content",
DATA = "data",
ENABLE = "enable",
EXPAND = "expand",
ENABLED = "enabled",
EXPANDED = "expanded",
ERROR = "error",
FILTER = "filter",
HIDE = "hide",
INFO = "info",
ISOPEN = "isOpen",
MAX = "max",
MIN = "min",
OPEN = "open",
PALETTE = "palette",
READONLY = "readonly",
RESIZE = "resize",
SEARCH = "search",
SELECT = "select",
SELECTED = "selected",
SHOW = "show",
SUCCESS = "success",
SIZE = "size",
TARGET = "target",
TITLE = "title",
VALUE = "value",
VALUES = "values",
WARNING = "warning";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
createBinding({
name: "kendoAutoComplete",
events: {
change: VALUE,
open: {
writeTo: ISOPEN,
value: true
},
close: {
writeTo: ISOPEN,
value: false
}
},
watch: {
enabled: ENABLE,
search: [SEARCH, CLOSE],
data: function (value)
{
ko.kendo.setDataSource(this, value);
},
value: VALUE
}
});
}());
```

BaseFormWithValidationScript.xml

```
<BaseForm SaveButtonText='@@Finish@@' FormTitle='' ProductVersion='1'
BuildVersion='41202100' Id='NF1' FormMode='NextGenMode' XmlNodeBoundTo='SKRootDefinition'
Name='frm45' xmlns='clr-
namespace:Skelta.Forms.Core.Controls;assembly:Skelta.Forms.Core'><ScriptCode><![CDATA[]]></
ScriptCode><Controls><BaseForm FormTitle='' ProductVersion='' BuildVersion='' Id='B1'
FormMode='NextGenMode' XmlNodeBoundTo='Baseform1' Name='Baseform1' ViewMode='Grid'
AddPosition='Bottom' EnableResponsiveView='False'><Controls><TextInput
XmlNodeBoundTo='DataText' Name='Textinput1'
Id='T1'><OnDataChangedScript><![CDATA[]]></OnDataChangedScript><PropertyExpressions><Proper
ty Name='Validate'><Code><![CDATA[if (control.findById('T2').value === 'ab')
{
return new ValidationOptions(false, 'Enter your comments.');
}
else
{
return new ValidationOptions(true, '');
}]]></Code></Property></PropertyExpressions></TextInput><TextInput
XmlNodeBoundTo='DataValue' Name='Textinput2'
Id='T2'><OnDataChangedScript><![CDATA[]]></OnDataChangedScript></TextInput></Controls><OnRo
wAddedScript><![CDATA[]]></OnRowAddedScript><OnRowDeletedScript><![CDATA[]]></OnRowDeletedS
cript></BaseForm></Controls></BaseForm>
```

PropertyLookup.htm

```
<!DOCTYPE html>
<html class="skep">
<head>
<title>Property Lookup</title>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<meta http-equiv="content-type" content="Type=text/html; charset=utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<script type="text/javascript" src="js/jquery-1.9.1.min.js"></script>
<script type="text/javascript" src="js/knockout-3.1.0.min.js"></script>
<link rel="stylesheet" href="css/SkeltaForms.min.css" />
<style type="text/css">
body
{
font-family: arial;
font-size: 14px;
}
.liveExample
{
padding: 1em;
background-color: #EEEEDD;
border: 1px solid #CCC;
max-width: 655px;
}
.liveExample input
{
font-family: Arial;
}
.liveExample b
{
font-weight: bold;
}
.liveExample p
{
margin-top: 0.9em;
margin-bottom: 0.9em;
}
.liveExample select[multiple]
{
width: 100%;
height: 8em;
}
.liveExample h2
{
margin-top: 0.4em;
font-weight: bold;
font-size: 1.2em;
}
</style>
</head>
<body style="margin: 20px;">
<p>
URL: <input data-bind="value: url" />
</p>
<p>
Data Text Field: <input data-bind="value: dataTextField" />
</p>
<button data-bind="click: cancelClick">Cancel</button>
<button data-bind="click: submitClick">Submit</button>
</body>
<script type="text/javascript">
var editPageVm = window.parent.frames["editframe"].viewModelObject ?
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
window.parent.frames["editframe"].viewModelObject :
window.parent.frames["editframe"].contentWindow.viewModelObject;
var ViewModel = function (dataurl, datatext)
{
this.url = ko.observable(dataurl);
this.dataTextField = ko.observable(datatext);
this.submitClick = function ()
{
// URL property control in edit page
editPageVm.findById("E5").value = this.url();
// Data Text Field property control in edit page
editPageVm.findById("E20").value = this.dataTextField();
parent.document.getElementById("propertyLookupFrame").attributes["data-skfsw"].value = 0;
parent.document.getElementById("propertyLookupFrame").attributes["src"].value = "";
};
this.cancelClick = function ()
{
parent.document.getElementById("propertyLookupFrame").attributes["data-skfsw"].value = 0;
parent.document.getElementById("propertyLookupFrame").attributes["src"].value = "";
};
};
// Load data in edit mode
var uri = editPageVm.findById("E5").value;
var dtf = editPageVm.findById("E20").value;
ko.applyBindings(new ViewModel(uri, dtf));
</script>
</html>
```

Custom Control with Standard Properties

You can create a custom control with a standard property.

The **Data Lookup** property can be used for consuming the Data Lookups

1. Create a custom control (Tree).
   a. To use the DataLookupProperty as a property (Configuration Details) for the Tree View control, code in the Tree.cs control initialization file
   as follows:

```
private DataLookupProperty dataLookupDetails = null;/// <summary>
/// Gets or Sets the configurationDetails property that holds lookup name and the
parameter collection
/// </summary>
[DefaultValue(null)]
[DesignerProperty(775000, Categories.Behaviour,
[BindingAttribute(882700,
"Skelta.Forms.Core.CommonObjects.DataLookupBindingProperty")]
public DataLookupProperty DataLookupDetails
{
get
{
return dataLookupDetails;
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
set
{
dataLookupDetails = value;
}
}
```

2. Build the custom control.
3. In **Forms Designer**, add the **TreeView** control to the form.
4. Modify the properties of the **TreeView** control as follows:
   a. In the **Basic** tab, click the **Data Source** icon. The **Lookup Source Configuration** page appears.
   b. In the **Data Source Type** drop-down, select a database.
   c. In the **Data Source** drop-down, select a lookup.
   d. Save the form.
5. Preview the form.
   You can see the the data lookup name and the parameter collection in the preview.

   Note: The **Tree** control is available as a standard control in Forms. This example is given to demonstrate the use of `DataLookupBindingProperty`.

Example: TreeView Control

A custom **TreeView** control using Kendo UI widget with control initialization, class, and JavaScript files.

- SFControl.cs (control initialization file)
- TreeView.cs (control class file)
- TreeViewBinding.cs (Bindings for the control)
- corp.customControls.treeView.js (JavaScript file for custom Knockout binding to support height, width, and read-only behavior)
- kendo.treeview.js
- knockout-kendo.treeview

Selecting the JavaScript files for the specific Kendo UI widget

The included kendo.all.min.js is limited to the following widgets:

- Grid
- DateTimePicker
- DatePicker
- TimePicker
- NumericTextBox
- ColorPicker
- PDF with the export features, pdf and drawing (built using the gulp command) for performance benefits.

To support additional widgets and features, you must refer only to the JavaScript files for the specific Kendo UI widget other than the JavaScript files for the included widgets. For this, you must be aware of script

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

dependencies and requirements for the Kendo UI widgets. Refer to Include Individual Widget Scripts for Kendo for the complete list of JavaScript files required for each Kendo UI widget.

Ensure you do not build the required Kendo UI widget JavaScript files using the gulp command as the script files created and registered will contain duplicate code parts.

SFControl.cs

```
using Corp.CustomControls;
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.External;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Skelta.Forms.ExternalControls
{
class SFControls : IExternalControls
{
#region IExternalControls Members
BaseControl IExternalControls.GetControl(string controlFullName)
{
switch (controlFullName)
{
case "Corp.CustomControls.TreeView":
return new TreeView();
default:
return null;
}
}
object[] IExternalControls.GetControls()
{
return new object[] { new TreeView() };
}
#endregion
}
}
```

TreeView.cs

```
namespace Corp.CustomControls
{
using System;
using System.ComponentModel;
using System.Text;
using System.Web;
using Skelta.Forms.Core.Controls;
using Skelta.Forms.Core.Interfaces;
using Skelta.Forms.Core.PropertyControls;
using Skelta.Forms.Core.CommonObjects;
[DesignerControl(992200, CoreDesigner.NextGenDesigner, DesignerCategory.Control)]
[SuppressedPropertiesNextGen("MobileDisplay;Height;Width;ReadOnly;MessageWhenEmpty;DefaultV
alue;ExpDefaultValue;OnDataChangedScript;ExpValue;ExpValidate;IncludeInSummaryView;")]
public class TreeView : BaseDataExternalControl
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
public override BaseControl GetNewInstanceForClone()
{
TreeView clonedTreeView = new TreeView();
clonedTreeView.dataLookupDetails = this.dataLookupDetails;
return clonedTreeView;
}
private DataLookupProperty dataLookupDetails = null;
/// <summary>
/// Gets or Sets the configurationDetails property that holds lookup name and the parameter
collection
/// </summary>
[DefaultValue(null)]
[DesignerProperty(775000, Categories.Behaviour, CoreDesigner.NextGenDesigner,
typeof(DataLookupProperty))]
[BindingAttribute(882700, "Skelta.Forms.Core.CommonObjects.DataLookupBindingProperty")]
public DataLookupProperty DataLookupDetails
{
get
{
return dataLookupDetails;
}
set
{
dataLookupDetails = value;
}
}
/// <summary>
/// Gets or sets the Custom Bindings
/// </summary>
[SerializationVisibility(SerializationVisibilityValues.Hidden)]
[BindingAttribute(882700, true, "Corp.CustomControls.TreeViewBinding")]
public string CustomBindings
{
get;
set;
}
/// <summary>
/// Method to add the custom Js files associated to the custom control
/// </summary>
private void AddJsFiles()
{
// "this.TopLevelForm.RuntimeCss" is the collection which holds your custom CSS files.
// It is a key-value pair collection where the key is the file name and the value is the
file path(absolute or relative).
//Important point is the JS file should be present inside "BPMUITemplate/Default/
NextGenForms/js" folder or else it won't get rendered.
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["kendo.treeview"]))
{
this.TopLevelForm.RuntimeJS.Add("kendo.treeview", "js/kendo.treeview.js");
}
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["corp.customControls.treeView"]))
{
this.TopLevelForm.RuntimeJS.Add("corp.customControls.treeView", "js/
corp.customControls.treeView.js");
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (string.IsNullOrEmpty(this.TopLevelForm.RuntimeJS["knockout-kendo.treeview"]))
{
this.TopLevelForm.RuntimeJS.Add("knockout-kendo.treeview", "js/knockout-
kendo.treeview.js");
}
}
public override string GetControlHtml()
{
AddJsFiles();
StringBuilder stringBuilder = new StringBuilder();
if (!this.TopLevelForm.InDesignMode)
{
StringBuilder dataBindAttribute = new StringBuilder();
stringBuilder.Append("<div" + this.GetControlIdAttribute + " title=\"" +
HttpUtility.HtmlEncode(this.ToolTip) + "\"");
dataBindAttribute.Append(" " + DataBindAttribute + "=\"kendoTreeView: {data: " + this.Id +
"._source, dataTextField: " + this.Id + "._dataTextField}\"");
stringBuilder.Append(dataBindAttribute.ToString() + ">");
stringBuilder.Append("</div>");
}
return stringBuilder.ToString();
}
public override string GetViewModelScript()
{
StringBuilder viewModelScript = new StringBuilder();
viewModelScript.Append(base.GetViewModelScript());
string controlString = "_" + this.ParentForm.Id + "." + this.Id;
viewModelScript.AppendLine("corp.customControls.treeView.initializeTreeView(" +
controlString + ");");
return viewModelScript.ToString();
}
public override IBindingProperty GetBindingClassObject(string fullyQualifiedClassPath)
{
try
{
return (IBindingProperty)Activator.CreateInstance(Type.GetType(fullyQualifiedClassPath));
}
catch (Exception ex)
{
return null;
}
}
}
}
```

TreeViewBinding.cs

```
namespace Corp.CustomControls
{
using System;
using System.Text;
using Skelta.Forms.Core;
using Skelta.Forms.Core.Classes;
using Skelta.Forms.Core.CommonObjects;
using Skelta.Forms.Core.Controls;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using Skelta.Forms.Core.Interfaces;
public class TreeViewBinding : IBindingProperty
{
public string RenderBinding(object bindingHelper)
{
StringBuilder stringBuilder = new StringBuilder();
BindingHelper bindingHelperObject = (BindingHelper)bindingHelper;
string treeViewControlString = bindingHelperObject.ViewModelContainerName + "." +
bindingHelperObject.Control.Id;
TreeView treeViewControl = bindingHelperObject.Control as TreeView;
stringBuilder.AppendLine(treeViewControlString + "._source = ko.observableArray();");
stringBuilder.AppendLine(treeViewControlString + "._dataTextField =
ko.observableArray();");
stringBuilder.AppendLine(treeViewControlString + ".refreshTreeView = function () {
corp.customControls.treeView.getTreeViewData(" + treeViewControlString + "); };");
return stringBuilder.ToString();
}
}
}
```

corp.customControls.treeView.js

```
(function ()
{
//// Defining Namespaces
window.corp = window.corp || {};
corp.customControls = corp.customControls || {};
corp.customControls.treeView = corp.customControls.treeView || {};
corp.customControls.treeView = (function ()
{
//// Initializes Tree View
function initializeTreeView(treeViewControl)
{
var lookupName = treeViewControl.lookupName,
parameterCollection = treeViewControl.parameterCollection;
if (lookupName === "")
{
return;
}
//////// Subscribes to parameter control
//////// When value of the parameter control changes, the function passed as the 3rd
parameter is called
if (!SFU.isUndefined(parameterCollection) && parameterCollection.length > 0)
{
SFU.subscribeToControlParameters(treeViewControl, parameterCollection, function () {
getTreeViewData(treeViewControl); });
}
//////// Calls the Lookup for getting the Schema and Data
getTreeViewData(treeViewControl);
}
//// Calls the Lookup for getting the Schema and Data
function getTreeViewData(treeViewControl)
{
var lookupName = treeViewControl.lookupName,
parameterCollection = SFU.getLookupParameterValues(treeViewControl,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
treeViewControl.parameterCollection);
var lookupData = SFU.getLookupSchemaAndData(lookupName, parameterCollection, false);
if (!SFU.isUndefined(lookupData))
{
treeViewControl._dataTextField(lookupData.DisplayColumn);
treeViewControl._source(lookupData.Data);
}
}
//// Public methods
return {
initializeTreeView: initializeTreeView,
getTreeViewData: getTreeViewData
};
})();
})();
```

kendo.treeview.js

```
/*
* Kendo UI v2015.2.624 (http://www.telerik.com/kendo-ui)
* Copyright 2015 Telerik AD. All rights reserved.
*
* Kendo UI commercial licenses may be obtained at
* http://www.telerik.com/purchase/license-agreement/kendo-ui-complete
* If you do not own a commercial license, this file shall be governed by the trial license
terms.
*/
(function(f, define){
define([ "./kendo.data", "./kendo.treeview.draganddrop" ], f);
})(function(){
/*jshint eqnull: true */
(function($, undefined){
var kendo = window.kendo,
ui = kendo.ui,
data = kendo.data,
extend = $.extend,
template = kendo.template,
isArray = $.isArray,
Widget = ui.Widget,
HierarchicalDataSource = data.HierarchicalDataSource,
proxy = $.proxy,
keys = kendo.keys,
NS = ".kendoTreeView",
SELECT = "select",
CHECK = "check",
NAVIGATE = "navigate",
EXPAND = "expand",
CHANGE = "change",
ERROR = "error",
CHECKED = "checked",
INDETERMINATE = "indeterminate",
COLLAPSE = "collapse",
DRAGSTART = "dragstart",
DRAG = "drag",
DROP = "drop",
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
DRAGEND = "dragend",
DATABOUND = "dataBound",
CLICK = "click",
VISIBILITY = "visibility",
UNDEFINED = "undefined",
KSTATEHOVER = "k-state-hover",
KTREEVIEW = "k-treeview",
VISIBLE = ":visible",
NODE = ".k-item",
STRING = "string",
ARIASELECTED = "aria-selected",
ARIADISABLED = "aria-disabled",
TreeView,
subGroup, nodeContents, nodeIcon,
spriteRe,
bindings = {
text: "dataTextField",
url: "dataUrlField",
spriteCssClass: "dataSpriteCssClassField",
imageUrl: "dataImageUrlField"
},
isDomElement = function (o){
return (
typeof HTMLElement === "object" ? o instanceof HTMLElement : //DOM2
o && typeof o === "object" && o.nodeType === 1 && typeof o.nodeName === STRING
);
};
function contentChild(filter) {
return function(node) {
var result = node.children(".k-animation-container");
if (!result.length) {
result = node;
}
return result.children(filter);
};
}
function templateNoWith(code) {
return kendo.template(code, { useWithBlock: false });
}
subGroup = contentChild(".k-group");
nodeContents = contentChild(".k-group,.k-content");
nodeIcon = function(node) {
return node.children("div").children(".k-icon");
};
function checkboxes(node) {
return node.find("> div .k-checkbox [type=checkbox]");
}
function insertAction(indexOffset) {
return function (nodeData, referenceNode) {
referenceNode = referenceNode.closest(NODE);
var group = referenceNode.parent(),
parentNode;
if (group.parent().is("li")) {
parentNode = group.parent();
}
return this._dataSourceMove(nodeData, group, parentNode, function (dataSource, model) {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
return this._insert(dataSource.data(), model, referenceNode.index() + indexOffset);
});
};
}
spriteRe = /k-sprite/;
function moveContents(node, container) {
var tmp;
while (node && node.nodeName.toLowerCase() != "ul") {
tmp = node;
node = node.nextSibling;
if (tmp.nodeType == 3) {
tmp.nodeValue = $.trim(tmp.nodeValue);
}
if (spriteRe.test(tmp.className)) {
container.insertBefore(tmp, container.firstChild);
} else {
container.appendChild(tmp);
}
}
}
function updateNodeHtml(node) {
var wrapper = node.children("div"),
group = node.children("ul"),
toggleButton = wrapper.children(".k-icon"),
checkbox = node.children(":checkbox"),
innerWrapper = wrapper.children(".k-in");
if (node.hasClass("k-treeview")) {
return;
}
if (!wrapper.length) {
wrapper = $("<div />").prependTo(node);
}
if (!toggleButton.length && group.length) {
toggleButton = $("<span class='k-icon' />").prependTo(wrapper);
} else if (!group.length || !group.children().length) {
toggleButton.remove();
group.remove();
}
if (checkbox.length) {
$("<span class='k-checkbox' />").appendTo(wrapper).append(checkbox);
}
if (!innerWrapper.length) {
innerWrapper = node.children("a").eq(0).addClass("k-in");
if (!innerWrapper.length) {
innerWrapper = $("<span class='k-in' />");
}
innerWrapper.appendTo(wrapper);
if (wrapper.length) {
moveContents(wrapper[0].nextSibling, innerWrapper[0]);
}
}
}
TreeView = kendo.ui.DataBoundWidget.extend({
init: function (element, options) {
var that = this,
dataInit,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
inferred = false,
hasDataSource = options && !!options.dataSource,
list;
if (isArray(options)) {
dataInit = true;
options = { dataSource: options };
}
if (options && typeof options.loadOnDemand == UNDEFINED && isArray(options.dataSource)) {
options.loadOnDemand = false;
}
Widget.prototype.init.call(that, element, options);
element = that.element;
options = that.options;
list = (element.is("ul") && element) ||
(element.hasClass(KTREEVIEW) && element.children("ul"));
inferred = !hasDataSource && list.length;
if (inferred) {
options.dataSource.list = list;
}
that._animation();
that._accessors();
that._templates();
// render treeview if it's not already rendered
if (!element.hasClass(KTREEVIEW)) {
that._wrapper();
if (list) {
that.root = element;
that._group(that.wrapper);
}
} else {
// otherwise just initialize properties
that.wrapper = element;
that.root = element.children("ul").eq(0);
}
that._tabindex();
that.root.attr("role", "tree");
that._dataSource(inferred);
that._attachEvents();
that._dragging();
if (!inferred) {
if (options.autoBind) {
that._progress(true);
that.dataSource.fetch();
}
} else {
that._syncHtmlAndDataSource();
}
if (options.checkboxes && options.checkboxes.checkChildren) {
that.updateIndeterminate();
}
if (that.element[0].id) {
that._ariaId = kendo.format("{0}_tv_active", that.element[0].id);
}
kendo.notify(that);
},
_attachEvents: function() {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
var that = this,
clickableItems = ".k-in:not(.k-state-selected,.k-state-disabled)",
MOUSEENTER = "mouseenter";
that.wrapper
.on(MOUSEENTER + NS, ".k-in.k-state-selected", function(e) { e.preventDefault(); })
.on(MOUSEENTER + NS, clickableItems, function () { $(this).addClass(KSTATEHOVER); })
.on("mouseleave" + NS, clickableItems, function () { $(this).removeClass(KSTATEHOVER); })
.on(CLICK + NS, clickableItems, proxy(that._click, that))
.on("dblclick" + NS, ".k-in:not(.k-state-disabled)", proxy(that._toggleButtonClick, that))
.on(CLICK + NS, ".k-plus,.k-minus", proxy(that._toggleButtonClick, that))
.on("keydown" + NS, proxy(that._keydown, that))
.on("focus" + NS, proxy(that._focus, that))
.on("blur" + NS, proxy(that._blur, that))
.on("mousedown" + NS, ".k-in,.k-checkbox :checkbox,.k-plus,.k-minus",
proxy(that._mousedown, that))
.on("change" + NS, ".k-checkbox :checkbox", proxy(that._checkboxChange, that))
.on("click" + NS, ".k-checkbox :checkbox", proxy(that._checkboxClick, that))
.on("click" + NS, ".k-request-retry", proxy(that._retryRequest, that))
.on("click" + NS, function(e) {
if (!$(e.target).is(":kendoFocusable")) {
that.focus();
}
});
},
_checkboxClick: function(e) {
var checkbox = $(e.target);
if (checkbox.data(INDETERMINATE)) {
checkbox
.data(INDETERMINATE, false)
.prop(INDETERMINATE, false)
.prop(CHECKED, true);
this._checkboxChange(e);
}
},
_syncHtmlAndDataSource: function(root, dataSource) {
root = root || this.root;
dataSource = dataSource || this.dataSource;
var data = dataSource.view();
var uidAttr = kendo.attr("uid");
var expandedAttr = kendo.attr("expanded");
var inferCheckedState = this.options.checkboxes;
var items = root.children("li");
var i, item, dataItem;
for (i = 0; i < items.length; i++) {
dataItem = data[i];
item = items.eq(i);
item.attr("role", "treeitem").attr(uidAttr, dataItem.uid);
dataItem.expanded = item.attr(expandedAttr) === "true";
if (inferCheckedState) {
dataItem.checked = checkboxes(item).prop(CHECKED);
}
this._syncHtmlAndDataSource(item.children("ul"), dataItem.children);
}
},
_animation: function() {
var options = this.options,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
animationOptions = options.animation;
if (animationOptions === false) {
animationOptions = {
expand: { effects: {} },
collapse: { hide: true, effects: {} }
};
} else if (!animationOptions.collapse || !("effects" in animationOptions.collapse)) {
animationOptions.collapse = extend({ reverse: true }, animationOptions.expand);
}
extend(animationOptions.collapse, { hide: true });
options.animation = animationOptions;
},
_dragging: function() {
var enabled = this.options.dragAndDrop;
var dragging = this.dragging;
if (enabled && !dragging) {
var widget = this;
this.dragging = new ui.HierarchicalDragAndDrop(this.element, {
reorderable: true,
$angular: this.options.$angular,
autoScroll: this.options.autoScroll,
filter: "div:not(.k-state-disabled) .k-in",
allowedContainers: ".k-treeview",
itemSelector: ".k-treeview .k-item",
hintText: proxy(this._hintText, this),
contains: function(source, destination) {
return $.contains(source, destination);
},
dropHintContainer: function(item) {
return item;
},
itemFromTarget: function(target) {
var item = target.closest(".k-top,.k-mid,.k-bot");
return {
item: item,
content: target.closest(".k-in"),
first: item.hasClass("k-top"),
last: item.hasClass("k-bot")
};
},
dropPositionFrom: function(dropHint) {
return dropHint.prevAll(".k-in").length > 0 ? "after" : "before";
},
dragstart: function(source) {
return widget.trigger(DRAGSTART, { sourceNode: source[0] });
},
drag: function(options) {
widget.trigger(DRAG, {
originalEvent: options.originalEvent,
sourceNode: options.source[0],
dropTarget: options.target[0],
pageY: options.pageY,
pageX: options.pageX,
statusClass: options.status,
setStatusClass: options.setStatus
});
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
},
drop: function(options) {
return widget.trigger(DROP, {
originalEvent: options.originalEvent,
sourceNode: options.source,
destinationNode: options.destination,
valid: options.valid,
setValid: options.setValid,
dropTarget: options.dropTarget,
dropPosition: options.position
});
},
dragend: function(options) {
var source = options.source;
var destination = options.destination;
var position = options.position;
function triggerDragEnd(source) {
widget.updateIndeterminate();
widget.trigger(DRAGEND, {
originalEvent: options.originalEvent,
sourceNode: source && source[0],
destinationNode: destination[0],
dropPosition: position
});
}
// perform reorder / move
// different handling is necessary because append might be async in remote bound tree
if (position == "over") {
widget.append(source, destination, triggerDragEnd);
} else {
if (position == "before") {
source = widget.insertBefore(source, destination);
} else if (position == "after") {
source = widget.insertAfter(source, destination);
}
triggerDragEnd(source);
}
}
});
} else if (!enabled && dragging) {
dragging.destroy();
this.dragging = null;
}
},
_hintText: function(node) {
return this.templates.dragClue({
item: this.dataItem(node),
treeview: this.options
});
},
_templates: function() {
var that = this,
options = that.options,
fieldAccessor = proxy(that._fieldAccessor, that);
if (options.template && typeof options.template == STRING) {
options.template = template(options.template);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
} else if (!options.template) {
options.template = templateNoWith(
"# var text = " + fieldAccessor("text") + "(data.item); #" +
"# if (typeof data.item.encoded != 'undefined' && data.item.encoded === false) {#" +
"#= text #" +
"# } else { #" +
"#: text #" +
"# } #"
);
}
that._checkboxes();
that.templates = {
wrapperCssClass: function (group, item) {
var result = "k-item",
index = item.index;
if (group.firstLevel && index === 0) {
result += " k-first";
}
if (index == group.length-1) {
result += " k-last";
}
return result;
},
cssClass: function(group, item) {
var result = "",
index = item.index,
groupLength = group.length - 1;
if (group.firstLevel && index === 0) {
result += "k-top ";
}
if (index === 0 && index != groupLength) {
result += "k-top";
} else if (index == groupLength) {
result += "k-bot";
} else {
result += "k-mid";
}
return result;
},
textClass: function(item) {
var result = "k-in";
if (item.enabled === false) {
result += " k-state-disabled";
}
if (item.selected === true) {
result += " k-state-selected";
}
return result;
},
toggleButtonClass: function(item) {
var result = "k-icon";
if (item.expanded !== true) {
result += " k-plus";
} else {
result += " k-minus";
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (item.enabled === false) {
result += "-disabled";
}
return result;
},
groupAttributes: function(group) {
var attributes = "";
if (!group.firstLevel) {
attributes = "role='group'";
}
return attributes + (group.expanded !== true ? " style='display:none'" : "");
},
groupCssClass: function(group) {
var cssClass = "k-group";
if (group.firstLevel) {
cssClass += " k-treeview-lines";
}
return cssClass;
},
dragClue: templateNoWith(
"#= data.treeview.template(data) #"
),
group: templateNoWith(
"<ul class='#= data.r.groupCssClass(data.group) #'#= data.r.groupAttributes(data.group) #>"
+
"#= data.renderItems(data) #" +
"</ul>"
),
itemContent: templateNoWith(
"# var imageUrl = " + fieldAccessor("imageUrl") + "(data.item); #" +
"# var spriteCssClass = " + fieldAccessor("spriteCssClass") + "(data.item); #" +
"# if (imageUrl) { #" +
"<img class='k-image' alt='' src='#= imageUrl #'>" +
"# } #" +
"# if (spriteCssClass) { #" +
"<span class='k-sprite #= spriteCssClass #' />" +
"# } #" +
"#= data.treeview.template(data) #"
),
itemElement: templateNoWith(
"# var item = data.item, r = data.r; #" +
"# var url = " + fieldAccessor("url") + "(item); #" +
"<div class='#= r.cssClass(data.group, item) #'>" +
"# if (item.hasChildren) { #" +
"<span class='#= r.toggleButtonClass(item) #' role='presentation' />" +
"# } #" +
"# if (data.treeview.checkboxes) { #" +
"<span class='k-checkbox' role='presentation'>" +
"#= data.treeview.checkboxes.template(data) #" +
"</span>" +
"# } #" +
"# var tag = url ? 'a' : 'span'; #" +
"# var textAttr = url ? ' href=\\'' + url + '\\'' : ''; #" +
"<#=tag# class='#= r.textClass(item) #'#= textAttr #>" +
"#= r.itemContent(data) #" +
"</#=tag#>" +
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"</div>"
),
item: templateNoWith(
"# var item = data.item, r = data.r; #" +
"<li role='treeitem' class='#= r.wrapperCssClass(data.group, item) #' " +
kendo.attr("uid") + "='#= item.uid #' " +
"aria-selected='#= item.selected ? \"true\" : \"false \" #' " +
"#=item.enabled === false ? \"aria-disabled='true'\" : ''#" +
"# if (item.expanded) { #" +
"data-expanded='true' aria-expanded='true'" +
"# } #" +
">" +
"#= r.itemElement(data) #" +
"</li>"
),
loading: templateNoWith(
"<div class='k-icon k-loading' /> #: data.messages.loading #"
),
retry: templateNoWith(
"#: data.messages.requestFailed # " +
"<button class='k-button k-request-retry'>#: data.messages.retry #</button>"
)
};
},
items: function() {
return this.element.find(".k-item > div:first-child");
},
setDataSource: function(dataSource) {
var options = this.options;
options.dataSource = dataSource;
this._dataSource();
this.dataSource.fetch();
if (options.checkboxes && options.checkboxes.checkChildren) {
this.updateIndeterminate();
}
},
_bindDataSource: function() {
this._refreshHandler = proxy(this.refresh, this);
this._errorHandler = proxy(this._error, this);
this.dataSource.bind(CHANGE, this._refreshHandler);
this.dataSource.bind(ERROR, this._errorHandler);
},
_unbindDataSource: function() {
var dataSource = this.dataSource;
if (dataSource) {
dataSource.unbind(CHANGE, this._refreshHandler);
dataSource.unbind(ERROR, this._errorHandler);
}
},
_dataSource: function(silentRead) {
var that = this,
options = that.options,
dataSource = options.dataSource;
function recursiveRead(data) {
for (var i = 0; i < data.length; i++) {
data[i]._initChildren();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
data[i].children.fetch();
recursiveRead(data[i].children.view());
}
}
dataSource = isArray(dataSource) ? { data: dataSource } : dataSource;
that._unbindDataSource();
if (!dataSource.fields) {
dataSource.fields = [
{ field: "text" },
{ field: "url" },
{ field: "spriteCssClass" },
{ field: "imageUrl" }
];
}
that.dataSource = dataSource = HierarchicalDataSource.create(dataSource);
if (silentRead) {
dataSource.fetch();
recursiveRead(dataSource.view());
}
that._bindDataSource();
},
events: [
DRAGSTART,
DRAG,
DROP,
DRAGEND,
DATABOUND,
EXPAND,
COLLAPSE,
SELECT,
CHANGE,
NAVIGATE,
CHECK
],
options: {
name: "TreeView",
dataSource: {},
animation: {
expand: {
effects: "expand:vertical",
duration: 200
}, collapse: {
duration: 100
}
},
messages: {
loading: "Loading...",
requestFailed: "Request failed.",
retry: "Retry"
},
dragAndDrop: false,
checkboxes: false,
autoBind: true,
autoScroll: false,
loadOnDemand: true,
template: "",
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
dataTextField: null
},
_accessors: function() {
var that = this,
options = that.options,
i, field, textField,
element = that.element;
for (i in bindings) {
field = options[bindings[i]];
textField = element.attr(kendo.attr(i + "-field"));
if (!field && textField) {
field = textField;
}
if (!field) {
field = i;
}
if (!isArray(field)) {
field = [field];
}
options[bindings[i]] = field;
}
},
// generates accessor function for a given field name, honoring the data*Field arrays
_fieldAccessor: function(fieldName) {
var fieldBindings = this.options[bindings[fieldName]],
count = fieldBindings.length,
result = "(function(item) {";
if (count === 0) {
result += "return item['" + fieldName + "'];";
} else {
result += "var levels = [" +
$.map(fieldBindings, function(x) {
return "function(d){ return " + kendo.expr(x) + "}";
}).join(",") + "];";
result += "return levels[Math.min(item.level(), " + count + "-1)](item)";
}
result += "})";
return result;
},
setOptions: function(options) {
Widget.fn.setOptions.call(this, options);
this._animation();
this._dragging();
this._templates();
},
_trigger: function (eventName, node) {
return this.trigger(eventName, {
node: node.closest(NODE)[0]
});
},
_setChecked: function(datasource, value) {
if (!datasource || !$.isFunction(datasource.view)) {
return;
}
for (var i = 0, nodes = datasource.view(); i < nodes.length; i++) {
nodes[i][CHECKED] = value;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (nodes[i].children) {
this._setChecked(nodes[i].children, value);
}
}
},
_setIndeterminate: function(node) {
var group = subGroup(node),
siblings, length,
all = true,
i;
if (!group.length) {
return;
}
siblings = checkboxes(group.children());
length = siblings.length;
if (!length) {
return;
} else if (length > 1) {
for (i = 1; i < length; i++) {
if (siblings[i].checked != siblings[i-1].checked ||
siblings[i].indeterminate || siblings[i-1].indeterminate) {
all = false;
break;
}
}
} else {
all = !siblings[0].indeterminate;
}
return checkboxes(node)
.data(INDETERMINATE, !all)
.prop(INDETERMINATE, !all)
.prop(CHECKED, all && siblings[0].checked);
},
updateIndeterminate: function(node) {
// top-down update of inital indeterminate state for all nodes
node = node || this.wrapper;
var subnodes = subGroup(node).children();
var i;
var checkbox;
if (subnodes.length) {
for (i = 0; i < subnodes.length; i++) {
this.updateIndeterminate(subnodes.eq(i));
}
checkbox = this._setIndeterminate(node);
if (checkbox && checkbox.prop(CHECKED)) {
this.dataItem(node).checked = true;
}
}
},
_bubbleIndeterminate: function(node) {
// bottom-up setting of indeterminate state of parent nodes
if (!node.length) {
return;
}
var parentNode = this.parent(node),
checkbox;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (parentNode.length) {
this._setIndeterminate(parentNode);
checkbox = parentNode.children("div").find(".k-checkbox :checkbox");
if (checkbox.prop(INDETERMINATE) === false) {
this.dataItem(parentNode).set(CHECKED, checkbox.prop(CHECKED));
} else {
this.dataItem(parentNode).checked = false;
}
this._bubbleIndeterminate(parentNode);
}
},
_checkboxChange: function(e) {
var checkbox = $(e.target);
var isChecked = checkbox.prop(CHECKED);
var node = checkbox.closest(NODE);
var dataItem = this.dataItem(node);
if (dataItem.checked != isChecked) {
dataItem.set(CHECKED, isChecked);
this._trigger(CHECK, node);
}
},
_toggleButtonClick: function (e) {
this.toggle($(e.target).closest(NODE));
},
_mousedown: function(e) {
var node = $(e.currentTarget).closest(NODE);
this._clickTarget = node;
this.current(node);
},
_focusable: function (node) {
return node && node.length && node.is(":visible") && !node.find(".k-in:first").hasClass("k-
state-disabled");
},
_focus: function() {
var current = this.select(),
clickTarget = this._clickTarget;
// suppress initial focus state on touch devices (until keyboard is used)
if (kendo.support.touch) {
return;
}
if (clickTarget && clickTarget.length) {
current = clickTarget;
}
if (!this._focusable(current)) {
current = this.current();
}
if (!this._focusable(current)) {
current = this._nextVisible($());
}
this.current(current);
},
focus: function() {
var wrapper = this.wrapper,
scrollContainer = wrapper[0],
containers = [],
offsets = [],
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
documentElement = document.documentElement,
i;
do {
scrollContainer = scrollContainer.parentNode;
if (scrollContainer.scrollHeight > scrollContainer.clientHeight) {
containers.push(scrollContainer);
offsets.push(scrollContainer.scrollTop);
}
} while (scrollContainer != documentElement);
wrapper.focus();
for (i = 0; i < containers.length; i++) {
containers[i].scrollTop = offsets[i];
}
},
_blur: function() {
this.current().find(".k-in:first").removeClass("k-state-focused");
},
_enabled: function(node) {
return !node.children("div").children(".k-in").hasClass("k-state-disabled");
},
parent: function(node) {
var wrapperRe = /\bk-treeview\b/,
itemRe = /\bk-item\b/,
result,
skipSelf;
if (typeof node == STRING) {
node = this.element.find(node);
}
if (!isDomElement(node)) {
node = node[0];
}
skipSelf = itemRe.test(node.className);
do {
node = node.parentNode;
if (itemRe.test(node.className)) {
if (skipSelf) {
result = node;
} else {
skipSelf = true;
}
}
} while (!wrapperRe.test(node.className) && !result);
return $(result);
},
_nextVisible: function(node) {
var that = this,
expanded = that._expanded(node),
result;
function nextParent(node) {
while (node.length && !node.next().length) {
node = that.parent(node);
}
if (node.next().length) {
return node.next();
} else {
return node;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
}
if (!node.length || !node.is(":visible")) {
result = that.root.children().eq(0);
} else if (expanded) {
result = subGroup(node).children().first();
// expanded node with no children
if (!result.length) {
result = nextParent(node);
}
} else {
result = nextParent(node);
}
if (!that._enabled(result)) {
result = that._nextVisible(result);
}
return result;
},
_previousVisible: function(node) {
var that = this,
lastChild,
result;
if (!node.length || node.prev().length) {
if (node.length) {
result = node.prev();
} else {
result = that.root.children().last();
}
while (that._expanded(result)) {
lastChild = subGroup(result).children().last();
if (!lastChild.length) {
break;
}
result = lastChild;
}
} else {
result = that.parent(node) || node;
}
if (!that._enabled(result)) {
result = that._previousVisible(result);
}
return result;
},
_keydown: function(e) {
var that = this,
key = e.keyCode,
target,
focused = that.current(),
expanded = that._expanded(focused),
checkbox = focused.find(".k-checkbox:first :checkbox"),
rtl = kendo.support.isRtl(that.element);
if (e.target != e.currentTarget) {
return;
}
if ((!rtl && key == keys.RIGHT) || (rtl && key == keys.LEFT)) {
if (expanded) {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
target = that._nextVisible(focused);
} else {
that.expand(focused);
}
} else if ((!rtl && key == keys.LEFT) || (rtl && key == keys.RIGHT)) {
if (expanded) {
that.collapse(focused);
} else {
target = that.parent(focused);
if (!that._enabled(target)) {
target = undefined;
}
}
} else if (key == keys.DOWN) {
target = that._nextVisible(focused);
} else if (key == keys.UP) {
target = that._previousVisible(focused);
} else if (key == keys.HOME) {
target = that._nextVisible($());
} else if (key == keys.END) {
target = that._previousVisible($());
} else if (key == keys.ENTER) {
if (!focused.find(".k-in:first").hasClass("k-state-selected")) {
if (!that._trigger(SELECT, focused)) {
that.select(focused);
}
}
} else if (key == keys.SPACEBAR && checkbox.length) {
checkbox.prop(CHECKED, !checkbox.prop(CHECKED))
.data(INDETERMINATE, false)
.prop(INDETERMINATE, false);
that._checkboxChange({ target: checkbox });
target = focused;
}
if (target) {
e.preventDefault();
if (focused[0] != target[0]) {
that._trigger(NAVIGATE, target);
that.current(target);
}
}
},
_click: function (e) {
var that = this,
node = $(e.currentTarget),
contents = nodeContents(node.closest(NODE)),
href = node.attr("href"),
shouldNavigate;
if (href) {
shouldNavigate = href == "#" || href.indexOf("#" + this.element.id + "-") >= 0;
} else {
shouldNavigate = contents.length && !contents.children().length;
}
if (shouldNavigate) {
e.preventDefault();
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (!node.hasClass(".k-state-selected") && !that._trigger(SELECT, node)) {
that.select(node);
}
},
_wrapper: function() {
var that = this,
element = that.element,
wrapper, root,
wrapperClasses = "k-widget k-treeview";
if (element.is("ul")) {
wrapper = element.wrap('<div />').parent();
root = element;
} else {
wrapper = element;
root = wrapper.children("ul").eq(0);
}
that.wrapper = wrapper.addClass(wrapperClasses);
that.root = root;
},
_group: function(item) {
var that = this,
firstLevel = item.hasClass(KTREEVIEW),
group = {
firstLevel: firstLevel,
expanded: firstLevel || that._expanded(item)
},
groupElement = item.children("ul");
groupElement
.addClass(that.templates.groupCssClass(group))
.css("display", group.expanded ? "" : "none");
that._nodes(groupElement, group);
},
_nodes: function(groupElement, groupData) {
var that = this,
nodes = groupElement.children("li"),
nodeData;
groupData = extend({ length: nodes.length }, groupData);
nodes.each(function(i, node) {
node = $(node);
nodeData = { index: i, expanded: that._expanded(node) };
updateNodeHtml(node);
that._updateNodeClasses(node, groupData, nodeData);
// iterate over child nodes
that._group(node);
});
},
_checkboxes: function() {
var options = this.options;
var checkboxes = options.checkboxes;
var defaultTemplate;
if (checkboxes) {
defaultTemplate = "<input type='checkbox' #= (item.enabled === false) ? 'disabled' : '' #
#= item.checked ? 'checked' : '' #";
if (checkboxes.name) {
defaultTemplate += " name='" + checkboxes.name + "'";
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
defaultTemplate += " />";
checkboxes = extend({
template: defaultTemplate
}, options.checkboxes);
if (typeof checkboxes.template == STRING) {
checkboxes.template = template(checkboxes.template);
}
options.checkboxes = checkboxes;
}
},
_updateNodeClasses: function (node, groupData, nodeData) {
var wrapper = node.children("div"),
group = node.children("ul"),
templates = this.templates;
if (node.hasClass("k-treeview")) {
return;
}
nodeData = nodeData || {};
nodeData.expanded = typeof nodeData.expanded != UNDEFINED ? nodeData.expanded :
this._expanded(node);
nodeData.index = typeof nodeData.index != UNDEFINED ? nodeData.index : node.index();
nodeData.enabled = typeof nodeData.enabled != UNDEFINED ? nodeData.enabled :
!wrapper.children(".k-in").hasClass("k-state-disabled");
groupData = groupData || {};
groupData.firstLevel = typeof groupData.firstLevel != UNDEFINED ? groupData.firstLevel :
node.parent().parent().hasClass(KTREEVIEW);
groupData.length = typeof groupData.length != UNDEFINED ? groupData.length :
node.parent().children().length;
// li
node.removeClass("k-first k-last")
.addClass(templates.wrapperCssClass(groupData, nodeData));
// div
wrapper.removeClass("k-top k-mid k-bot")
.addClass(templates.cssClass(groupData, nodeData));
// span
wrapper.children(".k-in").removeClass("k-in k-state-default k-state-disabled")
.addClass(templates.textClass(nodeData));
// toggle button
if (group.length || node.attr("data-hasChildren") == "true") {
wrapper.children(".k-icon").removeClass("k-plus k-minus k-plus-disabled k-minus-disabled")
.addClass(templates.toggleButtonClass(nodeData));
group.addClass("k-group");
}
},
_processNodes: function(nodes, callback) {
var that = this;
that.element.find(nodes).each(function(index, item) {
callback.call(that, index, $(item).closest(NODE));
});
},
dataItem: function(node) {
var uid = $(node).closest(NODE).attr(kendo.attr("uid")),
dataSource = this.dataSource;
return dataSource && dataSource.getByUid(uid);
},
_insertNode: function(nodeData, index, parentNode, insertCallback, collapsed) {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
var that = this,
group = subGroup(parentNode),
updatedGroupLength = group.children().length + 1,
childrenData,
groupData = {
firstLevel: parentNode.hasClass(KTREEVIEW),
expanded: !collapsed,
length: updatedGroupLength
}, node, i, item, nodeHtml = "",
append = function(item, group) {
item.appendTo(group);
};
for (i = 0; i < nodeData.length; i++) {
item = nodeData[i];
item.index = index + i;
nodeHtml += that._renderItem({
group: groupData,
item: item
});
}
node = $(nodeHtml);
if (!node.length) {
return;
}
that.angular("compile", function(){
return {
elements: node.get(),
data: nodeData.map(function(item){
return { dataItem: item };
})
};
});
if (!group.length) {
group = $(that._renderGroup({
group: groupData
})).appendTo(parentNode);
}
insertCallback(node, group);
if (parentNode.hasClass("k-item")) {
updateNodeHtml(parentNode);
that._updateNodeClasses(parentNode);
}
that._updateNodeClasses(node.prev().first());
that._updateNodeClasses(node.next().last());
// render sub-nodes
for (i = 0; i < nodeData.length; i++) {
item = nodeData[i];
if (item.hasChildren) {
childrenData = item.children.data();
if (childrenData.length) {
that._insertNode(childrenData, item.index, node.eq(i), append,
!that._expanded(node.eq(i)));
}
}
}
return node;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
},
_updateNodes: function(items, field) {
var that = this;
var i, node, nodeWrapper, item, isChecked, isCollapsed;
var context = { treeview: that.options, item: item };
var render = field != "expanded" && field != "checked";
function setCheckedState(root, state) {
root.find(".k-checkbox :checkbox")
.prop(CHECKED, state)
.data(INDETERMINATE, false)
.prop(INDETERMINATE, false);
}
if (field == "selected") {
item = items[0];
node = that.findByUid(item.uid).find(".k-in:first")
.removeClass("k-state-hover")
.toggleClass("k-state-selected", item[field])
.end();
if (item[field]) {
that.current(node);
}
node.attr(ARIASELECTED, !!item[field]);
} else {
var elements = $.map(items, function(item) {
return that.findByUid(item.uid).children("div");
});
if (render) {
that.angular("cleanup", function() { return { elements: elements }; });
}
for (i = 0; i < items.length; i++) {
context.item = item = items[i];
nodeWrapper = elements[i];
node = nodeWrapper.parent();
if (render) {
nodeWrapper.children(".k-in")
.html(that.templates.itemContent(context));
}
if (field == CHECKED) {
isChecked = item[field];
setCheckedState(nodeWrapper, isChecked);
if (that.options.checkboxes.checkChildren) {
setCheckedState(node.children(".k-group"), isChecked);
that._setChecked(item.children, isChecked);
that._bubbleIndeterminate(node);
}
} else if (field == "expanded") {
that._toggle(node, item, item[field]);
} else if (field == "enabled") {
node.find(".k-checkbox :checkbox").prop("disabled", !item[field]);
isCollapsed = !nodeContents(node).is(VISIBLE);
node.removeAttr(ARIADISABLED);
if (!item[field]) {
if (item.selected) {
item.set("selected", false);
}
if (item.expanded) {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
item.set("expanded", false);
}
isCollapsed = true;
node.attr(ARIASELECTED, false)
.attr(ARIADISABLED, true);
}
that._updateNodeClasses(node, {}, { enabled: item[field], expanded: !isCollapsed });
}
if (nodeWrapper.length) {
this.trigger("itemChange", { item: nodeWrapper, data: item, ns: ui });
}
}
if (render) {
that.angular("compile", function(){
return {
elements: elements,
data: $.map(items, function(item) {
return [{ dataItem: item }];
})
};
});
}
}
},
_appendItems: function(index, items, parentNode) {
var group = subGroup(parentNode);
var children = group.children();
var collapsed = !this._expanded(parentNode);
if (typeof index == UNDEFINED) {
index = children.length;
}
this._insertNode(items, index, parentNode, function(item, group) {
// insert node into DOM
if (index >= children.length) {
item.appendTo(group);
} else {
item.insertBefore(children.eq(index));
}
}, collapsed);
if (this._expanded(parentNode)) {
this._updateNodeClasses(parentNode);
subGroup(parentNode).css("display", "block");
}
},
_refreshChildren: function(parentNode, items, index) {
var i, children, child;
var options = this.options;
var loadOnDemand = options.loadOnDemand;
var checkChildren = options.checkboxes && options.checkboxes.checkChildren;
subGroup(parentNode).empty();
if (!items.length) {
updateNodeHtml(parentNode);
} else {
this._appendItems(index, items, parentNode);
children = subGroup(parentNode).children();
if (loadOnDemand && checkChildren) {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
this._bubbleIndeterminate(children.last());
}
for (i = 0; i < children.length; i++) {
child = children.eq(i);
this.trigger("itemChange", {
item: child.children("div"),
data: this.dataItem(child),
ns: ui
});
}
}
},
_refreshRoot: function(items) {
var groupHtml = this._renderGroup({
items: items,
group: {
firstLevel: true,
expanded: true
}
});
if (this.root.length) {
this._angularItems("cleanup");
var group = $(groupHtml);
this.root
.attr("class", group.attr("class"))
.html(group.html());
} else {
this.root = this.wrapper.html(groupHtml).children("ul");
}
this.root.attr("role", "tree");
for (var i = 0; i < items.length; i++) {
var elements = this.root.children(".k-item");
this.trigger("itemChange", {
item: elements.eq(i),
data: items[i],
ns: ui
});
}
this._angularItems("compile");
},
refresh: function(e) {
var node = e.node;
var action = e.action;
var items = e.items;
var parentNode = this.wrapper;
var options = this.options;
var loadOnDemand = options.loadOnDemand;
var checkChildren = options.checkboxes && options.checkboxes.checkChildren;
var i;
if (e.field) {
if (!items[0] || !items[0].level) {
return;
}
return this._updateNodes(items, e.field);
}
if (node) {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
parentNode = this.findByUid(node.uid);
this._progress(parentNode, false);
}
if (checkChildren && action != "remove") {
var bubble = false;
for (i = 0; i < items.length; i++) {
if ("checked" in items[i]) {
bubble = true;
break;
}
}
if (!bubble && node && node.checked) {
for (i = 0; i < items.length; i++) {
items[i].checked = true;
}
}
}
if (action == "add") {
this._appendItems(e.index, items, parentNode);
} else if (action == "remove") {
this._remove(this.findByUid(items[0].uid), false);
} else if (action == "itemchange") {
this._updateNodes(items);
} else if (action == "itemloaded") {
this._refreshChildren(parentNode, items, e.index);
} else {
this._refreshRoot(items);
}
if (action != "remove") {
for (i = 0; i < items.length; i++) {
if (!loadOnDemand || items[i].expanded) {
items[i].load();
}
}
}
this.trigger(DATABOUND, { node: node ? parentNode : undefined });
},
_error: function(e) {
var node = e.node && this.findByUid(e.node.uid);
var retryHtml = this.templates.retry({ messages: this.options.messages });
if (node) {
this._progress(node, false);
this._expanded(node, false);
nodeIcon(node).addClass("k-i-refresh");
e.node.loaded(false);
} else {
this._progress(false);
this.element.html(retryHtml);
}
},
_retryRequest: function(e) {
e.preventDefault();
this.dataSource.fetch();
},
expand: function (nodes) {
this._processNodes(nodes, function (index, item) {
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
this.toggle(item, true);
});
},
collapse: function (nodes) {
this._processNodes(nodes, function (index, item) {
this.toggle(item, false);
});
},
enable: function (nodes, enable) {
enable = arguments.length == 2 ? !!enable : true;
this._processNodes(nodes, function (index, item) {
this.dataItem(item).set("enabled", enable);
});
},
current: function(node) {
var that = this,
current = that._current,
element = that.element,
id = that._ariaId;
if (arguments.length > 0 && node && node.length) {
if (current) {
if (current[0].id === id) {
current.removeAttr("id");
}
current.find(".k-in:first").removeClass("k-state-focused");
}
current = that._current = $(node, element).closest(NODE);
current.find(".k-in:first").addClass("k-state-focused");
id = current[0].id || id;
if (id) {
that.wrapper.removeAttr("aria-activedescendant");
current.attr("id", id);
that.wrapper.attr("aria-activedescendant", id);
}
return;
}
if (!current) {
current = that._nextVisible($());
}
return current;
},
select: function (node) {
var that = this,
element = that.element;
if (!arguments.length) {
return element.find(".k-state-selected").closest(NODE);
}
node = $(node, element).closest(NODE);
element.find(".k-state-selected").each(function() {
var dataItem = that.dataItem(this);
if (dataItem) {
dataItem.set("selected", false);
delete dataItem.selected;
} else {
$(this).removeClass("k-state-selected");
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
});
if (node.length) {
that.dataItem(node).set("selected", true);
}
that.trigger(CHANGE);
},
_toggle: function(node, dataItem, expand) {
var options = this.options;
var contents = nodeContents(node);
var direction = expand ? "expand" : "collapse";
var loaded, empty;
if (contents.data("animating")) {
return;
}
if (!this._trigger(direction, node)) {
this._expanded(node, expand);
loaded = dataItem && dataItem.loaded();
empty = !contents.children().length;
if (expand && (!loaded || empty)) {
if (options.loadOnDemand) {
this._progress(node, true);
}
contents.remove();
dataItem.load();
} else {
this._updateNodeClasses(node, {}, { expanded: expand });
if (!expand) {
contents.css("height", contents.height()).css("height");
}
contents
.kendoStop(true, true)
.kendoAnimate(extend(
{ reset: true },
options.animation[direction],
{ complete: function() {
if (expand) {
contents.css("height", "");
}
} }
));
}
}
},
toggle: function (node, expand) {
node = $(node);
if (!nodeIcon(node).is(".k-minus,.k-plus,.k-minus-disabled,.k-plus-disabled")) {
return;
}
if (arguments.length == 1) {
expand = !this._expanded(node);
}
this._expanded(node, expand);
},
destroy: function() {
var that = this;
Widget.fn.destroy.call(that);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
that.wrapper.off(NS);
that._unbindDataSource();
if (that.dragging) {
that.dragging.destroy();
}
kendo.destroy(that.element);
that.root = that.wrapper = that.element = null;
},
_expanded: function(node, value) {
var expandedAttr = kendo.attr("expanded"),
dataItem = this.dataItem(node);
if (arguments.length == 1) {
return node.attr(expandedAttr) === "true" || (dataItem && dataItem.expanded);
}
if (nodeContents(node).data("animating")) {
return;
}
if (dataItem) {
dataItem.set("expanded", value);
// necessary when expanding an item yields an error and the item is not expanded as a
result
value = dataItem.expanded;
}
if (value) {
node.attr(expandedAttr, "true");
node.attr("aria-expanded", "true");
} else {
node.removeAttr(expandedAttr);
node.attr("aria-expanded", "false");
}
},
_progress: function(node, showProgress) {
var element = this.element;
var loadingText = this.templates.loading({ messages: this.options.messages });
if (arguments.length == 1) {
showProgress = node;
if (showProgress) {
element.html(loadingText);
} else {
element.empty();
}
} else {
nodeIcon(node).toggleClass("k-loading", showProgress).removeClass("k-i-refresh");
}
},
text: function (node, text) {
var dataItem = this.dataItem(node),
fieldBindings = this.options[bindings.text],
level = dataItem.level(),
length = fieldBindings.length,
field = fieldBindings[Math.min(level, length-1)];
if (text) {
dataItem.set(field, text);
} else {
return dataItem[field];
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
},
_objectOrSelf: function (node) {
return $(node).closest("[data-role=treeview]").data("kendoTreeView") || this;
},
_dataSourceMove: function(nodeData, group, parentNode, callback) {
var referenceDataItem,
destTreeview = this._objectOrSelf(parentNode || group),
destDataSource = destTreeview.dataSource;
var loadPromise = $.Deferred().resolve().promise();
if (parentNode && parentNode[0] != destTreeview.element[0]) {
referenceDataItem = destTreeview.dataItem(parentNode);
if (!referenceDataItem.loaded()) {
destTreeview._progress(parentNode, true);
loadPromise = referenceDataItem.load();
}
if (parentNode != this.root) {
destDataSource = referenceDataItem.children;
if (!destDataSource || !(destDataSource instanceof HierarchicalDataSource)) {
referenceDataItem._initChildren();
referenceDataItem.loaded(true);
destDataSource = referenceDataItem.children;
}
}
}
nodeData = this._toObservableData(nodeData);
return callback.call(this, destDataSource, nodeData, loadPromise);
},
_toObservableData: function(node) {
var dataItem = node, dataSource, uid;
if (node instanceof window.jQuery || isDomElement(node)) {
dataSource = this._objectOrSelf(node).dataSource;
uid = $(node).attr(kendo.attr("uid"));
dataItem = dataSource.getByUid(uid);
if (dataItem) {
dataItem = dataSource.remove(dataItem);
}
}
return dataItem;
},
_insert: function(data, model, index) {
if (!(model instanceof kendo.data.ObservableArray)) {
if (!isArray(model)) {
model = [model];
}
} else {
// items will be converted to new Node instances
model = model.toJSON();
}
var parentNode = data.parent();
if (parentNode && parentNode._initChildren) {
parentNode.hasChildren = true;
parentNode._initChildren();
}
data.splice.apply(data, [ index, 0 ].concat(model));
return this.findByUid(data[index].uid);
},
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
insertAfter: insertAction(1),
insertBefore: insertAction(0),
append: function (nodeData, parentNode, success) {
var that = this,
group = that.root;
if (parentNode) {
group = subGroup(parentNode);
}
return that._dataSourceMove(nodeData, group, parentNode, function (dataSource, model,
loadModel) {
var inserted;
function add() {
if (parentNode) {
that._expanded(parentNode, true);
}
var data = dataSource.data(),
index = Math.max(data.length, 0);
return that._insert(data, model, index);
}
loadModel.then(function() {
inserted = add();
success = success || $.noop;
success(inserted);
});
return inserted || null;
});
},
_remove: function (node, keepData) {
var that = this,
parentNode,
prevSibling, nextSibling;
node = $(node, that.element);
this.angular("cleanup", function(){
return { elements: node.get() };
});
parentNode = node.parent().parent();
prevSibling = node.prev();
nextSibling = node.next();
node[keepData ? "detach" : "remove"]();
if (parentNode.hasClass("k-item")) {
updateNodeHtml(parentNode);
that._updateNodeClasses(parentNode);
}
that._updateNodeClasses(prevSibling);
that._updateNodeClasses(nextSibling);
return node;
},
remove: function (node) {
var dataItem = this.dataItem(node);
if (dataItem) {
this.dataSource.remove(dataItem);
}
},
detach: function (node) {
return this._remove(node, true);
},
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
findByText: function(text) {
return $(this.element).find(".k-in").filter(function(i, element) {
return $(element).text() == text;
}).closest(NODE);
},
findByUid: function(uid) {
var items = this.element.find(".k-item");
var uidAttr = kendo.attr("uid");
var result;
for (var i = 0; i < items.length; i++) {
if (items[i].getAttribute(uidAttr) == uid) {
result = items[i];
break;
}
}
return $(result);
},
expandPath: function(path, complete) {
path = path.slice(0);
var treeview = this;
var dataSource = this.dataSource;
var node = dataSource.get(path[0]);
complete = complete || $.noop;
function tryExpand(node, complete, context) {
if (node && !node.loaded()) {
node.set("expanded", true);
} else {
complete.call(context);
}
}
// expand loaded nodes
while (path.length > 0 && node && (node.expanded || node.loaded())) {
node.set("expanded", true);
path.shift();
node = dataSource.get(path[0]);
}
if (!path.length) {
return complete.call(treeview);
}
// expand async nodes
dataSource.bind("change", function expandLevel(e) {
// listen to the change event to know when the node has been loaded
var id = e.node && e.node.id;
var node;
// proceed if the change is caused by the last fetching
if (id && id === path[0]) {
path.shift();
if (path.length) {
node = dataSource.get(path[0]);
tryExpand(node, complete, treeview);
} else {
complete.call(treeview);
}
}
});
tryExpand(node, complete, treeview);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
},
_parents: function(node) {
var parent = node && node.parentNode();
var parents = [];
while (parent && parent.parentNode) {
parents.push(parent);
parent = parent.parentNode();
}
return parents;
},
expandTo: function(node) {
if (!(node instanceof kendo.data.Node)) {
node = this.dataSource.get(node);
}
var parents = this._parents(node);
for (var i = 0; i < parents.length; i++) {
parents[i].set("expanded", true);
}
},
_renderItem: function (options) {
if (!options.group) {
options.group = {};
}
options.treeview = this.options;
options.r = this.templates;
return this.templates.item(options);
},
_renderGroup: function (options) {
var that = this;
options.renderItems = function(options) {
var html = "",
i = 0,
items = options.items,
len = items ? items.length : 0,
group = options.group;
group.length = len;
for (; i < len; i++) {
options.group = group;
options.item = items[i];
options.item.index = i;
html += that._renderItem(options);
}
return html;
};
options.r = that.templates;
return that.templates.group(options);
}
});
ui.plugin(TreeView);
})(window.kendo.jQuery);
return window.kendo;
}, typeof define == 'function' && define.amd ? define : function(_, f){ f(); });
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

knockout-kendo.treeview.js

```
// Knockout binding for Kendo Tree View
(function ()
{
var createBinding = ko.kendo.bindingFactory.createBinding.bind(ko.kendo.bindingFactory);
//use constants to ensure consistency and to help reduce minified file size
var ENABLE = "enable"
VALUE = "value";
// Knockout Kendo Tree View
createBinding({
name: "kendoTreeView",
watch: {
data: function (value, options)
{
ko.kendo.setDataSource(this, value, options);
}
},
events: {
change: function (options, e)
{
if (ko.isWriteableObservable(options.value))
{
var tree = e.sender;
options.value(tree.dataItem(tree.select()));
}
}
},
async: true
});
}());
```

Add Custom Control Members as White List for Check syntax validations

To add custom control members as whiteList for Check syntax validations

1. Open the *Skelta.CustomControlsMembers.js* file from **BPMUITemplates\Default\NextGenForms\app\custom**
2. Uncomment the self invoking function.

```
(function ()
{
var standardControlMembers = skelta.forms.constants.controlsCollection;
var genericControlFunctions =
skelta.forms.constants.controlFunctions;//genericControlFunctions has properties -
nonContainerFunctions and containerFunctions which is a collection
of function names [{ name: "functionName"}]
var customControlMembers = {
functions: genericControlFunctions.nonContainerFunctions,//.concat([{ name:
"functionName" }
isContainerControl: false,
controlType: "ToolBar",
properties: [{ propertyName: "xmlNodeBoundTo", propertyType: "2", isWritable: false
},
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{ propertyName: "tagName", propertyType: "0", isWritable: true },
{ propertyName: "id", propertyType: "2", isWritable: false },
{ propertyName: "value", propertyType: "0", isWritable: true },
{ propertyName: "isMandatory", propertyType: "0", isWritable: true },
{ propertyName: "readOnly", propertyType: "0", isWritable: true },
{ propertyName: "enable", propertyType: "0", isWritable: true },
{ propertyName: "resizable", propertyType: "0", isWritable: true },
{ propertyName: "visible", propertyType: "0", isWritable: true }]
}
standardControlMembers.push(customControlMembers);
})();
```

3. In the code:
   - The `genericControlFunctions` has the properties, `nonContainerFunctions` and `containerFunctions` which are a collection of function names - `[{ name: "functionName"}]`.
   - `controlType` is the name of the custom control.
4. Use the `customControlMembers` object as per the custom control by adding properties to the properties array.
   - `propertyName` is the name of the property.
   - `propertyType` is "0" for observable, "1" for observableArray and "2" for nonObservable.
   - `isWritable` is true or false - depending on whether the property is assignable or readonly.
5. Push the `customControlMembers` to the `standardControlMembers` collection.

For examples on adding "MaskedInput" and "Barchart" properties for Check syntax validations, see the Adding Masked Input Property for Check Syntax Validation and Adding Barchart Properties for Check Syntax Validations topics respectively.

Default Values

Default value is the value assigned to the control automatically when you do not set any value to the control explicitly.

Control	**Default Value**
Boolean	"False"
Check Box	Empty string
Date	Empty string
Drop-Down	Empty string (if **Options Caption** is set) First Options Value (if **Options Caption** is not set)
Email	Empty string
List	Empty string

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Control	Default Value
Number	Empty string
Radio Button	Empty string
Text	Empty string
Time	Empty string

Validation of Values

Value rules are triggered when the value of a control changes.

A data type mismatch occurs when an invalid value is set for a control either through the user interface (UI) or through the scripts. In such cases, the value of the control is reset.

When an invalid value is set for a control either through the UI or through the scripts, the control behaves as follows:

Control	User Interface	Scripts
• Boolean	• Not applicable.	• Updates the value of the control with the invalid value. • Shows the error message in the user interface. • Prevents submission of the form.
• Check Box • Drop-Down • List • Radio Button	• Not applicable.	• Updates the value of the control with the invalid value. • No error message is shown in the user interface. • Allows submission of the form with the invalid value.
• Date • Date Time • Time	• Updates the value of the control with the invalid value. • Shows the error message. • Prevents submission of the form.	• Updates the value of the control with the invalid value. • Shows the error message in the user interface. • Prevents submission of the form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Control	User Interface	Scripts
• Email • Text	• Updates the value of the control with the invalid value. • Shows the error message. • Prevents submission of the form.	• Updates the value of the control with the invalid value. • Shows the error message in the user interface. • Prevents submission of the form.

Note: When a value is set through the scripts for the **Value** or the **Default Value** property of a control, the validation for the property is not triggered on page load.

Form Layouts

Layouts are the organization of controls on a form using container controls. Layouts are used to design complex forms.

Design Form Layouts

Design a layout to organize controls on a form using container controls.

To design a form layout

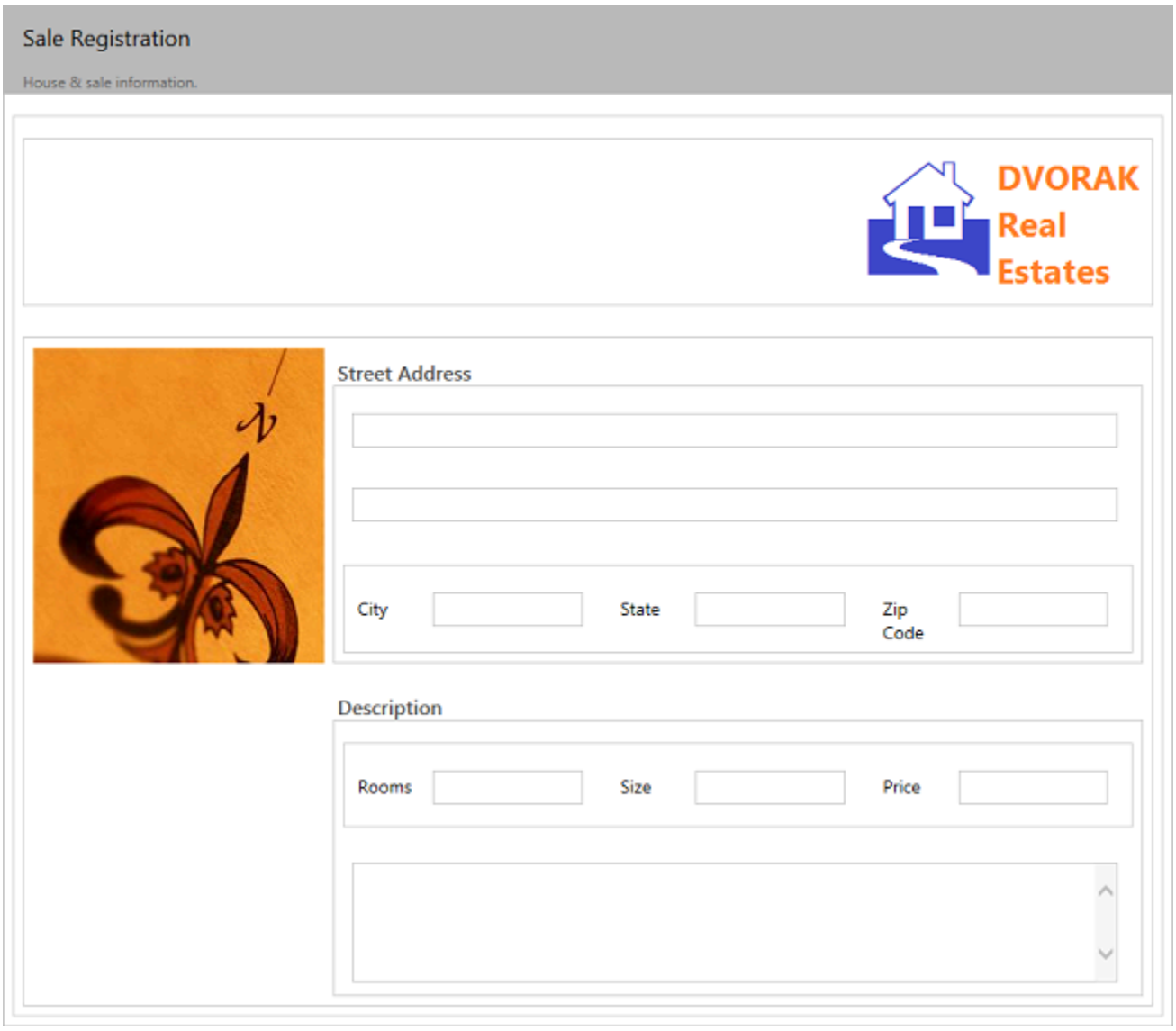
1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click **Design**). The **Forms Designer** page appears.
3. Drag a **Panel Form** container from the toolbox to the desired location on the canvas. The selected control appears on the canvas.
4. Point to the **Panel Form** container, and then click the **Properties** button. The **Properties** dialog box appears.
5. In the **Name** box, delete the name.
6. Click the **Appearance** tab, and then in the **Columns** box, type the number of columns required.
7. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
8. Drag the required controls or containers from the toolbox to the desired column on the **Panel Form** container.
9. Set the required properties for the new controls or containers.
10. Preview the form to check look and feel of the form with the layout at run-time.
11. Click the **Close** button. The form is saved and the **Forms Designer** page is closed.

Note:
- When placing controls and containers in **Base Form** and **Panel Form**, ensure the placed controls and containers are visible and accessible to the user.
- Controls in the other columns of the **Panel Form** cannot be aligned when there are no controls in the first column of the **Panel Form**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example: Sale Registration Form

A sale registration form for capturing information about a house for sale requires many controls. Placing controls one after the other on the form results in poor form design. Instead, use container controls to design layouts for organizing controls on the form. Layouts improve usability of the form.



To create the form

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. Click **New**, and then click **Form**. The **New Form** wizard appears.
3. In the **Name** box, type **Sale Registration**, and in the **Description** box, type **House & Sale information**.
4. Click **Save & Continue**. The **Sale Registration** form is created, and the **Form Options** wizard appears.

To design the form with layouts

1. Click **Design** this form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Click **Finish**. The **Forms Designer** page appears.
3. Drag **Panel Form** container from the toolbox to the canvas. The **Panel Form** appears on the canvas.
4. Point to the **Panel Form (P1)** container, and then click the **Properties** button. The **Properties** dialog box appears.
5. In the **Name** box, delete the name.
6. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
7. Drag **Panel Form** container from the toolbox and place it on the **Panel Form (P1)** container. The **Panel Form** appears on the canvas.
8. Point to the **Panel Form (P2)** container, and then click the **Properties** button. The **Properties** dialog box appears.
9. In the **Name** box, delete the name.
10. Click the **Appearance** tab, and then in the **Columns** box, type **2**. A two-column layout is created.
11. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
12. Drag **Image** control from the toolbox and place it on the second column of the **Panel Form (P2).** The **Image** control appears on the canvas.
13. Point to the **Image** control, and then click the **Properties** button. The **Properties** dialog box appears.
14. In the **Image Path** box, type the path to the image, and then click **Apply**. The properties are saved and the **Properties** dialog box is closed.
15. Drag **Panel Form** container from the toolbox and place it on the **Panel Form (P1)** container below the **Panel Form (P2)** container. The **Panel Form** appears on the canvas.
16. Point to the **Panel Form (P3)** container, and then click the **Properties** button. The **Properties** dialog box appears.
17. In the **Name** box, delete the name.
18. Click the **Appearance** tab, and then in the **Columns** box, type **2**. A two-column layout is created.
19. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
20. Drag **Image** control from the toolbox and place it on the first column of the **Panel Form (P3).** The **Image control** appears on the canvas.
21. Point to the **Image control**, and then click the **Properties** button. The **Properties** dialog box appears.
22. In the **Image Path** box, type the path to the image, and then click **Apply**. The properties are saved and the **Properties** dialog box is closed.
23. Drag **Base Form** container from the toolbox and place it on the second column of the **Panel Form (P3)** container. The **Base Form** container appears on the canvas.
24. In the **Name** box, type **Street Address**.
25. Click the **Appearance** tab, and then in the **Label Position** list, click **Suppress**.
26. Click the **Validation** tab, and then in the **Repeating** list, click **False**.
27. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
28. Drag **Text** control from the toolbox and place it on the **Base Form (B1)** container. The **Text** control appears on the canvas.
29. Point to the **Text** control, and then click the **Properties** button. The **Properties** dialog box appears.
30. In the **Name** box, delete the name.
31. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

32. Drag **Text** control from the toolbox and place it on the **Base Form (B1)** container. The **Text** control appears on the canvas.
33. Point to the **Text** control, and then click the **Properties** button. The **Properties** dialog box appears.
34. In the **Name** box, delete the name.
35. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
36. Drag **Panel Form** container from the toolbox and place it on the **Base Form (B1)** container below the text controls. The Panel Form appears on the canvas.
37. Point to the **Panel Form (P4)** container, and then click the **Properties** button. The **Properties** dialog box appears.
38. In the **Name** box, delete the name.
39. Click the **Appearance** tab, and then in the **Columns** box, type 3. A three-column layout is created.
40. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
41. Drag **Text** control from the toolbox and place it on the first column of the **Panel Form (P4).** The **Text** control appears on the canvas.
42. Point to the **Text** control, and then click the **Properties** button. The **Properties** dialog box appears.
43. In the **Name** box, type **City**.
44. Click the **Appearance** tab, and then in the **Label Position** list, click **Left**.
45. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
46. Drag **Text** control from the toolbox and place it on the second column of the **Panel Form (P4).** The **Text** control appears on the canvas.
47. Point to the **Text** control, and then click the **Properties** button. The **Properties** dialog box appears.
48. In the **Name** box, type State.
49. Click the **Appearance** tab, and then in the **Label Position** list, click **Left**.
50. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
51. Drag **Text** control from the toolbox and place it on the third column of the **Panel Form (P4).** The **Text** control appears on the canvas.
52. Point to the **Text** control, and then click the **Properties** button. The **Properties** dialog box appears.
53. In the **Name** box, type **Zip Code**.
54. Click the **Appearance** tab, and then in the **Label Position** list, click **Left**.
55. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
56. Drag **Base Form** container from the toolbox and place it on the second column of the **Panel Form (P3)** container below the **Base Form (B1)** container. The **Base Form** container appears on the canvas.
57. In the **Name** box, type **Description**.
58. Click the **Appearance** tab, and then in the **Label Position** list, click **Suppress**.
59. Click the **Validation** tab, and then in the **Repeating** list, click **False**.
60. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
61. Drag **Panel Form** container from the toolbox and place it on the **Base Form (B2)** container. The **Panel Form** appears on the canvas.
62. Point to the **Panel Form (P5)** container, and then click the **Properties** button. The **Properties** dialog box appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

63. In the **Name** box, delete the name.
64. Click the **Appearance** tab, and then in the **Columns** box, type **3**. A three-column layout is created.
65. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
66. Drag **Text** control from the toolbox and place it on the first column of the **Panel Form (P5)**. The **Text** control appears on the canvas.
67. Point to the **Text** control, and then click the **Properties** button. The **Properties** dialog box appears.
68. In the **Name** box, type **Rooms**.
69. Click the **Appearance** tab, and then in the **Label Position** list, click **Left**.
70. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
71. Drag **Text** control from the toolbox and place it on the second column of the **Panel Form (P5)**. The **Text** control appears on the canvas.
72. Point to the **Text** control, and then click the **Properties** button. The **Properties** dialog box appears.
73. In the **Name** box, type **Size**.
74. Click the **Appearance** tab, and then in the **Label Position** list, click **Left**.
75. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
76. Drag **Text** control from the toolbox and place it on the third column of the **Panel Form (P5).** The **Text** control appears on the canvas.
77. Point to the **Text** control, and then click the **Properties** button. The **Properties** dialog box appears.
78. In the **Name** box, type **Price**.
79. Click the **Appearance** tab, and then in the **Label Position** list, click **Left**.
80. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
81. Drag **Text** control from the toolbox and place it on the **Base Form (B2)** below the **Panel Form (P5).** The **Text** control appears on the canvas.
82. Point to the **Text** control, and then click the **Properties** button. The **Properties** dialog box appears.
83. Click the **Appearance** tab, and then in the **Display Mode** list, click **Multi Line**.
84. In the **Rows** box, type **5**.
85. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.

To preview the form

1. On the action bar, click **Preview**. The **Preview** dialog box appears.
2. Check look and feel of the **Sale Registration** form.
3. Click the **Close** button. The **Preview** dialog box is closed.

To publish the form

1. On the action bar, click **Publish**. The form is published and an information message appears.
2. Click **OK**. The information message is closed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Forms in Custom Pages

Forms in custom pages allow you to customize your business application by loading a required form. The forms in custom pages are applicable only for **Fill Form** scenarios and not for **Work Item Inbox** scenarios.

Load Single Form

Load a required form in a custom page to customize your business application.

To load a form in a custom page

1. Create a custom page using the Single Form Custom Page Template in the *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\NextGenForms* folder.
2. Edit the custom page using an HTML editor to add a form by replacing the placeholders with the **Repository Name**, **User ID**, **Form Name**, **Form Version**, and **Form ID** as follows:

```
var appName = "Forms Repository"; // Repository Name – Case Sensitive value should be provided
var userId = "skeltalist::326E19DC-A292-49A0-B8A0-B8E62A9F3F27"; // Repository Provider with the User ID separated by the “::”
var formName = "TestForm"; // Form Name – Case Sensitive value should be provided
var formVersion = "1"; // Form Version
var formId = "bb48516f-d69f-43ce-99f3-aa6e39d70281"; // Form ID – should be in small case letters
```

Note:
- `appName` is the Repository Name and is a case sensitive value.
- `userId` is the internal User ID in the form of Repository Provider Instance Name and the User Id separated by the “::”.
- `formName` is the Form Name, which needs to be shown in the custom page and is a case sensitive value.
- `formId` is the Form GUID, which needs to be shown in the custom page and should be in small case letters.

To access forms parameters in a custom page

To access form parameters in a custom page, refer Setting Form Parameters

To access a form in a custom page placed in a folder other than NextGenForms

If the custom page is not placed in the folder relative to the AVEVA Work Tasks installed path, *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\NextGenForms\..*,then declare the following constant with the variables defined while loading and accessing forms in a custom page.

Also, the value depends on the location of the custom page and is a relative path. Consider the custom page to be in the path *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Repository\Site*.. The below line of code should be added before the “`var appName …`” code.

```
skelta.serverUrl = "../../NextGenForms/";
```

We also need to update the Javascript and CSS file references in the custom page code as below (listed few references below as an example):

```
<script type="text/javascript" src="../../Common/JavaScripts/spin.min.js"></script>
<script type="text/javascript" src="../../NextGenForms/app/Skelta.min.js"></script>
<script type="text/javascript" src="../../NextGenForms/js/date.min.js"></script>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<link rel="stylesheet" href="../../NextGenForms/css/SkeltaForms.min.css" />
<link rel="stylesheet" href="../../CommonUXControls/StyleSheet/se.ui.min.css" />
```

To access a Pop-Up Form in a custom page

If the custom page is not placed in the folder relative to the AVEVA Work Tasks installed path, **<AVEVA Work Tasks Install Path>\BPMUITemplates\Default\NextGenForms\..**,then declare the following constant with the variables defined while loading and accessing forms in a custom page.

The value is pre-fixed while opening the PopupForm HTML page to create a relevant relative path. Also, the value of the constant depends on the location of the custom page.

Consider the custom page to be in the path *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Repository\Site\..* The below line of code should be added before the "`var appName …`" code.

```
skelta.forms.constants.popupFormContants.HtmRelativePath = "../../NextGenForms/";
skelta.serverUrl = "../../NextGenForms/";
```

Single Form Custom Page Template

Use this **Custom Page** template to load a form in a custom page.

```
<!doctype html>
<html>
<head>
<title>Loading a Form</title>
<meta http-equiv="content-type" content="Type=text/html; charset=utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<script type="text/javascript" src="../Common/JavaScripts/spin.min.js"></script>
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/knockout.min.js"></script>
<script type="text/javascript" src="js/xml2json.min.js"></script>
<script type="text/javascript" src="app/Skelta.min.js"></script>
<script type="text/javascript" src="js/kendodependencies.min.js"></script>
<script type="text/javascript" src="js/kendo.all.min.js"></script>
<script type="text/javascript" src="js/knockout-kendo.min.js"></script>
<script type="text/javascript" src="js/date.min.js"></script>
<link rel="stylesheet" href="css/SkeltaForms.min.css" />
<link rel="stylesheet" href="css/SkeltaTabs.min.css" />
<link rel="stylesheet" href="css/kendo.common.min.css" />
<link rel="stylesheet" href="css/kendo.default.min.css" />
<link rel="stylesheet" href="../CommonUXControls/StyleSheet/se.ui.min.css" />
<link rel="stylesheet" href="../CommonUXControls/StyleSheet/UXControls.min.css" />
<script type="text/javascript">
var skeltaUtils = skelta.forms.utilities;

skelta.DeviceOS = "Web";

var logger = skeltaUtils.getLogger();

var appName = "<Repository Name>";
var userId = "<User ID>";
var formName = "<Form Name>";
var formVersion = "<Form Version>";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
var formId = "<Form ID>";
var listName = "Forms List";

$(document).ready(function ()
{
var spinnerEl = skeltaUtils.getLoaderObject();
$("#skloader")[0].appendChild(spinnerEl);

skelta.userContext.setUserContextFor("appN", appName);
skelta.userContext.setUserContextFor("uId", userId);
skelta.userContext.setUserContextFor("fName", formName);
skelta.userContext.setUserContextFor("vStamp", formVersion);
skelta.userContext.setUserContextFor("itemId", formId);
skelta.userContext.setUserContextFor("listName", listName);

skeltaUtils.loadPageDefault();

loadViewAndViewModelFromServer();
});

function loadViewAndViewModelFromServer()
{
var jsonHelper = skeltaUtils.createJsonRequestHelper({
"applicationName": appName,
"userId": userId,
"formName": formName,
"versionStamp": formVersion,
"listName": listName,
"deviceType": "Web"
});

skeltaUtils.callServerMethod("GET",
skeltaUtils.getServerUrl("NextGenFormService.aspx/GetViewAndViewModel"),
"application/json; charset=utf-8",
jsonHelper,
"json",
loadViewAndViewModelSucceeded,
loadViewAndViewModelFailed,
function ()
{
skeltaUtils.pageLoadComplete("");
});
}

function loadViewAndViewModelSucceeded(resultData)
{
$("#viewModel").html(JSON.parse(resultData["d"])["ViewAndViewModel"]);

var viewModelObject = skeltaUtils.getViewModelObject(skeltaUtils.getFormUniqueKey(appName,
formId, formVersion));

// Form Finish Click Event
viewModelObject.onSave.subscribeEvent(onFormFinish);

// Form Cancel Click Event
viewModelObject.onCancel.subscribeEvent(onFormCancel);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
skelta.localize.divTextLocalize();
skelta.localize.divTitleLocalize();

$("#viewModel")[0].style.display = "";
viewModelObject.topLevelForm.initializeFocus(0);
skelta.forms.delay.loader.scrollEventHandler();
}

function loadViewAndViewModelFailed(result)
{
skeltaUtils.showError(document.title,
skelta.localize.getString("FormNGFFillFormAjaxFailed").replace("<@paramApplicationName@>",
appName).replace("<@paramFormName@>", formName).replace("<@paramVersionStamp@>",
formVersion).replace("<@status@>", result.status).replace("<@statusText@>",
result.statusText).replace("<@responseText@>", result.responseText));
logger.logError(skelta.localize.getString("FormNGFFillFormAjaxFailed").replace("<@paramAppl
icationName@>", appName).replace("<@paramFormName@>",
formName).replace("<@paramVersionStamp@>", formVersion).replace("<@status@>",
result.status).replace("<@statusText@>", result.statusText).replace("<@responseText@>",
result.responseText));

skelta.forms.utilities.hideLoader();
}

function onFormFinish(data, evt)
{
//
// Add code for required action to be performed, when Finish button is clicked.
//

skeltaUtils.showAlert(document.title, data.InstanceJSON);

//
// Use values of Forms Parameters as required
//
}

function onFormCancel(data, evt)
{
//
// Add code for required action to be performed, when Cancel button is clicked.
//

skeltaUtils.showAlert(document.title, "Form - Cancel clicked");
}
</script>
</head>
<body style="margin: 0;">
<div id="skloader" class="skloader" data-skloader="1" style="height: 100%; width: 100%;
position: absolute;"></div>
<form class="sksa">
<div id="viewModel" style="display: none;">
</div>
</form>
</body>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</html>
```

Localization

Internationalization and localization make software ready for the world by adapting the software to a specific market.

Internationalization

Internationalization is the process of adapting software to different languages and regional differences. Internationalization focuses on separating the language-related aspects of the software from the code.

Localization

Localization is the process of adapting internationalized software to a specific language, script, and culture. Localization focuses on cultural adaptation for specific countries, regions, and groups.

Localize Forms

Localize forms to make it usable with a specific language, script, and culture by localizing controls, standard validation messages, and messages in validation scripts.

Workflow Management Forms provides two types of localization in Enterprise Console.

- Design time localization
- Run time localization

Design time localization

Design time localization happens when a form is published. At the time of publishing a form, the Forms Engine will find all the keys (prefixed and suffixed with **@@**, for example **@@entity@@**) mentioned inside the form (either in the properties like tag name or button text or in the script) and replace the values from the text files inside **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2** folder including subfolders.

AVEVA Work Tasks saves the localized View Model Object in the database. When you open/render a form, it will be loaded from the View Model Object (except when preview). This means that any changes made in the locale files after the form is published will not reflect until the respective forms are republished.

Run time localization

Run time localization happens when a form is loaded either from Navigation Link, Pop-Up form, or is rendered in HTML page. The Forms Engine will load all the key value pairs from all the text files inside **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2\Standard** folder at client side.

To find the value of a key, use the **skelta.localize.getString()** API.

The keys mentioned with **@@** will be localized at the time of publishing of the form. At client side, **skelta.localize.getString()** can be used for run time localization.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Changing the key/value pair for design time localization will need all the dependent forms to be republished.
- Design time localization is faster than run time localization.

Localize Control Names and Descriptions

Localize a control name and description to make it usable with a specific language, script, and culture.

To localize a control name and description

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click **Design**). The **Forms Designer** page appears.
3. Point to the control, and then click the **Properties** button. The **Properties** dialog box appears.
4. In the **Name** box, type **@@**<*localize_name*>**@@.**
5. In the **Description** box, type **@@**<*localize_description*>**@@.**
6. Click **Apply**. The properties are saved and the **Properties** dialog box is closed.
7. Add a key-value pair for name and description in an existing or a new text file in the **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2** and **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2\Standard** folders in the following format: **@@**<*localization_key*>**@@:Localized Value**.

   That is **@@**<*localize_name*>**@@:Localized Name** for **Name** and for **@@**<*localize_description*>**@@:Localized Description** for **Description**.

   Note: In the format **@@**<*localization_key*>**@@,** the <*localization_key*> key name should not have **space** and **special characters**. However, you can use **-** and **_** characters for the key names.
8. On the action bar, click **Preview**.
9. Check look and feel of the form for browser with the localized values.
10. Click the **Close** button. The **Preview** dialog box is closed.

Example

To localize a **Text** control name to **Employee Name** and description to **Contract Employee**, first set the **Name** property to **@@emp_name@@** (the localization key you have chosen for the name) and set the **Description** property to **@@emp_desc@@** (the localization key you have chosen for the description), then add a key-value pair in an existing or a new text file in the **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2** folder as **@@emp_name@@:Employee Name** and **@@emp_desc@@:Contract Employee**.

Note:
- When you localize **Button Text** of **Button** control, use the unresolved localized text for **Value** while creating conditions for **Button Pressed** in **Possible Distinct Outputs** property for **Invoke Form** activity of workflow.
- For example, if the **Button Text** is **Approve** and the localized text is **@@Approve@@**, then use **@@Approve@@** for **Value** of **Button Pressed** in **Possible Distinct Outputs** property for **Invoke Form** activity of the workflow.
- For information about Invoke Form activity and Possible Distinct Outputs, see **Invoke Form Activity** in *Developer Guide*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Localize Standard Validation Messages

Localize a standard validation message to make it usable with a specific language, script, and culture.

To localize a standard validation message

1. Identify the key for a standard validation message from the text file in the **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2\Standard** folder.
2. Add a key-value pair in an existing or a new text file in the following format: **FormNGFMandatory:<@Name@><localized text>.**

Example

When the **Name** property of a control is left blank, the **Name is mandatory** validation message appears. The key for this message in the text file is **FormNGFMandatory**. To localize or change this message to, say, **Name must be specified.**, add a key-value pair in an existing or a new text file in the **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2\Standard** folder as **FormNGFMandatory: <@Name@> must be specified**.

Localize Messages in Validation Scripts

Localize a message in a validation script to make it usable with a specific language, script, and culture.

To localize a message in a validation script

1. Open an existing text file or create a new text file in the **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2\Standard folder**.
2. Add a key-value pair in the following format: **<message>:<localized text**>.

Example

To display a localized message, say, Special characters are not allowed., when validating the Employee Name property through a script, add a key-value pair in an existing or a new text file in the **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2\Standard** folder as **InvalidEmployeeName: Special characters are not allowed**.

Validation Script

```
if (currentValue==="%")
{
return new ValidationOptions(false,skelta.localize.getString("InvalidEmployeeName"));
}
```

Key-Value Pair

```
InvalidEmployeeName: Special characters are not allowed.
```

Localize Column Names in Data Grid

Data Grid column names considers localization for names with **@@** prefix and suffix.

Example of a Key-Value Pair:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

@@FN@@:First Name

Where, **@@FN@@** is the localized key given for a **Display Column** while configuring grids. Ensure not to provide any space in the localized key.

In the Configure Grids, double click the **Display Column** and specify the key with **@@** as prefix and suffix.

Add the key/value pair to the run time localization file. For more information, see section *Run time localization* in Localize Forms.

Number and Date Formats

Number Formats

Number formatting is used to convert a numeric value to a human readable string using culture specific settings.

Standard Numeric Formats

Format	Description	Example (for en-US)
n	Number	n 1234.567 is displayed as 1,234.57
		n0 10.12 is displayed as 10
		n5 10.12 is displayed as 10.12000
c	Currency	c 1234.567 is displayed as $1,234.57
		c0 1234.567 is displayed as $1,235
p	Percentage	p 0.222 is displayed as 22.20%
		p0 0.222 is displayed as 22%
e	Exponential	e4 0.122 is displayed as 1.2200e-1

Custom Numeric Formats

Create custom numeric format string using one or more custom numeric specifiers.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Format	Description	Example (for en-US)
0	Zero placeholder. Replaces the zero with the corresponding digit if one is present; otherwise, zero appears in the result string.	00000 1234.5678 is displayed as 01235
#	Digit placeholder. Replaces the hash sign with the corresponding digit if one is present; otherwise, no digit appears in the result string. **Note:** '#' specifier cannot be used to format a number as telephone number, that is (###)-###-####.	##### 1234.5678 is displayed as 1235
.	Decimal placeholder. Determines the location of the decimal separator in the result string. The '.' symbol is replaced with the localized decimal symbol. **Note:** You can set the decimal values only if the value in the **Decimal Places** box has been set to a number greater than 0.	0.00 0.45678 is displayed as 0.46
,	Group separator placeholder. Determines whether a group separator is required for a number. The ',' symbol is replaced with the localized group separator symbol. **Note:** - The numbers are grouped or separated based on the current server culture. This property is applicable only when the **Show Group Separator** property is set **to Yes**. - If you want to use comma as the group separator irrespective of the current server culture, then enter '**,**' in the **Group Separator** box.	##,# 12345678 is displayed as 12,345,678

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

%	Percentage placeholder The '%' symbol is replaced with the localized percentage symbol. **Note:** '%' symbol is interpreted as a format specifier in the format string. If you need to prevent this, you will need to precede the '%' symbol with a '\'.	• If the format is **p** and the value of the number is '0.25', then the number is displayed as '25%'. • If the format is **#.## %** and the value of the number is '0.25', then the number is displayed as '25%'. • If the format is **#.## \%** and the value of the number is '0.25', then the number is displayed as '0.25 %'.
$	Currency placeholder. The number is formatted based on the culture settings for currency in the current server. The $ symbol is replaced with the localized currency symbol. **Note:** '$' symbol is interpreted as a format specifier in the format string. If you need to prevent this, you will need to precede the '$' symbol with a '****'.	If the current server culture is “en-US”, then the currency is displayed as '$'. However, if the current server culture is “Brazilian”, then the currency is displayed as 'R$'. • If the format is **# $** and the number is 12, then the number is displayed as 12 R$. • If the format is **# \$** and the number is 12, then the number is displayed as 12 $.
;	Section separator. Defines sections with separate format strings for positive, negative, and zero numbers.	#.00 $;#.00 - $;#.00 $ If you enter the value as '125', then it is displayed as '125.00 $'. If you enter the value as '-125', then it is displayed as '125.00 - $'. If you enter the value as '0', then it is displayed as '0.00 $'.
"string"/'string'	Literal string delimiter. Indicates that the enclosed characters should be copied to the result string.	#,##,###.00 ‘₹’ or #,##,###.00 “₹” If you enter the value as '125.00', then it is displayed as '125.00 ₹'.

Date and Time Formats

Date and time formatting is to convert date and time value to a human readable string using culture's specific

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

settings.

Standard Date and Time Formats

Format	Description	Example (for en-US)
d	Short date pattern.	M/d/yyyy 11/6/2000
"D"	Long date pattern.	dddd, MMMM dd, yyyy Monday, November 06, 2000
F	Full date/time pattern.	dddd, MMMM dd, yyyy h:mm:ss tt Monday, November 06, 2000 12:00:00 AM
g	General date/time pattern (short time).	M/d/yyyy h:mm tt 11/6/2000 12:00 AM
G	General date/time pattern (long time).	M/d/yyyy h:mm:ss tt 11/6/2000 12:00:00 AM
M/m	Month/day pattern.	MMMM dd November 06
u	Universal sortable date/time pattern.	yyyy'-'MM'-'dd HH':'mm':'ss'Z' 2000-11-06 00:00:00Z
Y/y	Year/month pattern.	MMMM, yyyy November, 2000

Custom Date and Time Formats

Create custom date and time format string using one or more custom date and time specifiers.

Format	Description
d	The day of the month, from 1 through 31.
dd	The day of the month, from 01 through 31.
ddd	The abbreviated name of the day of the week.
dddd	The full name of the day of the week.
f	The tenths of a second in a date and time value.
ff	The hundredths of a second in a date and time value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

fff	The milliseconds in a date and time value.
M	The month, from 1 through 12.
MM	The month, from 01 through 12.
MMM	The abbreviated name of the month.
MMMM	The full name of the month.
h	The hour, using a 12-hour clock from 1 to 12.
hh	The hour, using a 12-hour clock from 01 to 12.
H	The hour, using a 24-hour clock from 1 to 23.
HH	The hour, using a 24-hour clock from 01 to 23.
m	The minute, from 0 through 59.
mm	The minute, from 00 through 59.
s	The second, from 0 through 59.
ss	The second, from 00 through 59.
tt	The AM/PM designator.
yy	The last two characters from year value.
yyyy	The year full value.

Form Scripts

Scripts are a list of instructions to accomplish a desired operation. You can use various methods in the Forms APIs in your scripts to customize your business applications. Forms use JavaScript to deliver rich and dynamic web content.

Use scripts to create business rules and complex scenarios in your forms.

Note:
- Forms customization through scripting can introduce security vulnerabilities if the End-Users who author those scripts are unfamiliar with Secure Coding principles or are acting maliciously. Forms Designer permission role should only be assigned to users who are trusted not to abuse the system and are familiar with secure coding principles.
- AVEVA Work Tasks supports JavaScript based on the ES5 (ES2014) standard, applicable for Forms.

Framework

The following framework is used to validate data on a form before submitting the form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- JavaScript
- Script Editor
- SOA through Scripts

JavaScript

JavaScript is a dynamic, object-oriented, and interpreted client-side scripting language for web pages. JavaScript is used to validate data on a form before submitting the form. JavaScript is also used to make content interactive or load new content on a page and submit data to a server without reloading the page.

Script Editor

Script Editor is a custom editor to code in JavaScript.

To work with the Script Editor

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click **Design**). The **Forms Designer** page appears.
3. Point to the control, and then click the **Properties** button. The **Properties** dialog box appears.
4. Click the **Scripts** tab, and then click the **Script Editor** button for the required property. The **Script Editor** page appears.
5. Code the script, and then click **Save & Close**.

   The script is saved and the **Script Editor** page is closed.

   Note: If you click **Cancel**, the **Script Editor** page is closed. Any changes made to the code since it was last saved, is discarded.

Script Editor Interface

Action Bar

Action bar has commands to edit the script and preview the form.

Menu/Command	Description
Undo	Undo the last change made to the script.
Redo	Redo the last change that was undone.
Revert	Revert to a previously saved script.
Clear All	Clear the code from the editor.
Indent All	Indent the code.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Check Syntax	Validate the code. **Note:** - Use Camel case when coding in JavaScript. - We recommend to validate the code before saving the script to ensure there are no errors in the code.
Preview	Preview the form. For more information, see Preview Forms.
Device Preview	Preview the form from a device. For more information, see Preview Forms.

XML Structure

The XML Structure displays all the controls that contain XML nodes in the same hierarchy as the form is designed. When you select a node, the default script is inserted in the script editor. If you select the same control as that in which you are coding the script, then the `currentValue` is displayed. If you select a container control, the default script inserted is `control.findByXmlNode("Xml node")`. If you select any other control, the default script inserted is `control.findByXmlNode("Xml node").value`.

APIs

The APIs displays all the `Skelta.form.utilities` (SFU) APIs. When you select an API, the API is inserted inside the script editor.

Control Structure

The Control Structure displays all the control nodes in the same hierarchy as the form is designed. When you select a node, the default script is inserted in the script editor. If you select the same control as that in which you are coding the script, then the currentValue is displayed. If you select a container control, the default script inserted is `control.findById("ID")`. If you select any other control, the default script inserted is `control.findById("ID").value`. If you select the top level root node, the script inserted is `control.topLevelForm`.

Intellisense

Use intellisene to code scripts in the Scripts Editor . Intellisense in the Script Editor is triggered when you press Ctrl + Space.

- When you press Ctrl + Space in a new line or when the Script Editor is empty, then a list of all the available global objects appears.
- When you write `control.` and press Ctrl + Space, then depending on the control or container that you are scripting for in the script editor, a list of the control or container properties and methods appears. For example:
  - To add scripts in the script editor of a **Text** control, when you write `control.` and press Ctrl + Space, a list of the properties and methods applicable only for Text control appears.
  - To add scripts in the script editor of a **Base Form** container, when you write `control.` and press Ctrl + Space, a list of the properties and methods applicable only for **Base Form** container appears.
- When you write `control.findById( , control.findByName(`, or `control.findByXmlNode(` and press Ctrl + Space, then a list of the Ids, names or XML nodes applicable for the control or container appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:
- The editor must not have any string with **]]>** or **CDATA**, otherwise you will not be able to save the script.
- The script must return a value for all the properties under the **Scripts** tab of the respective control.
- The script is auto saved every 1 minute.
- To write the scripts for functions, see Best Practices.

Validate Scripts

Validate a script to ensure the code is valid.

To validate a script

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click Design). The **Forms Designer** page appears.
3. Point to the control, and then click the **Properties** button. The **Properties** dialog box appears.
4. Click the **Scripts** tab, and then click the **Script Editor** button for the required property. The **Script Editor** page appears.
5. Code the script, and then click **Check Syntax**. The code is validated.
6. Correct the invalid code and then click **Save & Close**. The script is saved and the **Script Editor** page is closed.

Properties and Methods

The following properties and methods are validated:

Properties	Methods
• enable • visible • readOnly • isMandatory • value • topLevelForm	• findById() • findByXmlNode() • findByName()

Validation

- Code is validated only when the **Check Syntax** button is clicked and not when the **Save** or **Save & Close** button is clicked.
- Control specific syntax is validated only if the script starts with the **control** keyword.
- User referenced JavaScript files are not validated.
- Run-time scenarios are not validated.

```
var b = "T1";
control.findById(b).value;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
var a = control;
// This line of code is not validated.
a.findById("T1").value
```

SOA through Scripts

1. Create a dynamic-link library (DLL) or an assembly with your custom business logic.
2. Add the assembly to the global assembly cache (GAC) or place the assembly in a dedicated folder.
3. If the assembly is placed in a dedicated folder, then register the location of the folder with the **SOA Folder List**.
4. In order to call the SOA method, code your script for your form taking care to ensure the type and order of the method parameters should match the method.

For information on using SOA through scripts, see SFU.soaMethod() API. For information about SOA, see *Workflow Management User Guide*.

Guidelines

This section defines the guidelines to be followed while using the form containers and controls.

Properties and Events

This section guides you on using the scripts to define the various properties and events of a control.

Enable

Use scripts for the **Enable** property to enable or disable a control.

To enable or disable a control

- Code the script as follows for the **Enable** property of a control:

Task	Script
• Enable the control.	`return true;`
• Disable the control.	`return false;`

To enable or disable a control based on the value of another control

You can enable or disable a control based on the value of another control.

Example

If you want to enable the **Last Name** (T2) control only when a name is entered in the **First Name** (T1) control, then you can code the script for the **Enable** property of the **Last Name** (T2) control as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if(control.findById("T1").value !== "")
{
return true;
}
else
{
return false;
}
```

Note:
- The **currentValue** method returns the value for the current property of a control.
- For the **Enable** property, the value is true when the control is enabled and the value is false when the control is disabled.

Visible

Use scripts for the **Visible** property to show or hide a control.

To show or hide a control

- Code the script as follows for the Visible property of a control:

Task	Script
• Show the control.	`return true;`
• Hide the control.	`return false;`

To show or hide a control based on the value of another control

You can show or hide a control based on the value of another control.

Example

If you want to show the **Last Name** (T2) control only when a name is entered in the **First Name** (T1) control, then you can code the script for the **Visible** property of the **Last Name** (T2) control as follows:

```
if(control.findById("T1").value !== "")
{
return true;
}
else
{
return false;
}
```

Note:
- The **currentValue** method returns the value for the current property of a control.
- For the **Visible** property, the value is true when the control is shown and the value is false when the control is hidden.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Read-Only

Use scripts for the **Read-Only** property to allow or deny a user to enter a value for the control.

To allow or deny a user to enter a value for a control

- Code the script as follows for the **Read-Only** property of a control:

Task	Script
• Deny a user to enter a value for the control.	`return true;`
• Allow a user to enter a value for the control.	`return false;`

To allow or deny a user to enter a value for a control based on the value of another control

You can allow or deny a user to enter a value for a control based on the value of another control.

Example

If you want to deny a user to change the value in the **Comments** (T1) control when the **Edit Comments** (D1) control is set to **No** (here the value for **No** is **0** and the value for **Yes** is **1**), then you can code the script for the **Read-Only** property of the **Comments** (T1) control as follows.

```
if(control.findById("D1").value !== "1")
{
return true;
}
else
{
return false;
}
```

Note:
- The **currentValue** method returns the value for the current property of a control.
- For the **Read-Only** property, the value is true when the control is only viewable and the value is false when the control is editable.

Mandatory

Use scripts for the **Mandatory** property to enforce or bypass entering a value for a control.

To enforce or bypass entering a value for a control

- Code the script as follows for the **Mandatory** property of a control:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Script
• Enforce entering a value for the control.	`return true;`
• Bypass entering a value for the control.	`return false;`

To enforce or bypass entering a value for a control based on the value of another control

You can enforce or bypass entering a value for a control based on the value of another control.

Example

If you want to enforce entering a value for the **Last Name** (T2) control when a name is entered in the **First Name** (T1) control, then you can code the script for the **Mandatory** property of the **Last Name** (T2) control as follows:

```
if(control.findById("T1").value !== "")
{
return true;
}
else
{
return false;
}
```

Note:
- The **currentValue** method returns the value for the current property of a control.
- For the **Mandatory** property, the value is true when the control is enforced and the value is false when the control is bypassed.

Validate

Use scripts for the **Validate** property to validate custom validations for a control.

Validations are triggered only when a form is submitted and the submission of the form fails even if one validation fails.

To validate custom validations for a control

- Code the script as follows for the **Validate** property of a control:

Task	Script
• Validate custom validations for the control.	`return new ValidationOptions(true, "");`
• Show an error message when custom validations for the control fails.	`return new ValidationOptions(false, "Error Message");`

To validate custom validations for a control based on the value of another control

You can validate custom validations for a control based on the value of another control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example

If you want to enforce entering a value for the **Comments** (T1) control as per the selection in the **Action** (D1) control, then you can code the script for the **Validate** property of the **Comments** (T1) control as follows:

```
if(control.findById("D1").value !== "1")
{
if (control.findById("T1").value === "")
{
return new ValidationOptions(false, "Enter your comments.");
}
else
{
return new ValidationOptions(true, "");
}
}
else
{
return new ValidationOptions(true, "");
}
```

If you want to enforce entering a value for the **Password** (P1) control with at least eight characters containing at least a number, a lowercase letter, and an uppercase letter, then you can code the script for the **Validate** property of the **Password** (P1) control as follows:

```
var re = /(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}/;
if( re.test(currentValue) === false)
{
return new ValidationOptions(false, "Password must have at least eight characters  and
contain a number, a lowercase letter, and an uppercase letter.");
}
else
{
return new ValidationOptions(true, "");
}
```

Options

Use scripts for the **Options** property to add options (options text and options value) for a **Check Box**, **Drop-down**, **Radio Button**, or **List** control.

Although JavaScript is used for scripting, the return value of the **Options** script is specified in JSON (JavaScript Object Notation) string format with attribute–value pairs.

To add options for a Check Box, Drop-down, Radio Button, or List control

- Code the script as follows for the **Options** property of a **Check Box**, **Drop-down**, **Radio Button**, or **List** control:

```
// Name is the Text Parameter and ID is the Value Parameter.
return [{"Name":"India","ID":"1"},{"Name":"UK","ID":"2"},{"Name":"USA","ID":"3"}];
```

Note:
- The options **Text** and **Value** parameters are mandatory and must be unique across all options.
- **Text Parameter** and **Value Parameter** properties must be specified before using the **Options** property script.
- Ensure that the value for the **Value Parameter** property is either String or Integer.
- If you add values using both **Options List** and **Options** script, then the values added using **Options** script takes

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

precedence.
- If you add options using scripts, then the value for the selected options persists even when the options are changed.
- To ensure no values are persisted, you need to explicitly clear the values before changing the options as given below.

```
control.value = "";
```

- If you explicitly clear the values, then the **On Data Change** event is triggered.
- Ensure the script returns a value.

To add options using Data Grid Values

You can add options for a **Check Box**, **Drop-down**, **Radio Button**, or **List** control using the values from a **Data Grid** control. To do this, first set the **Text Parameter** and the **Value Parameter** for the options of the control with the corresponding column name in the **Data Grid** control, and then code the script as follows:

When the Behavior property of Data Grid control is set as Data Grid

```
// Grid1 is the Data Grid whose values are used in the options of the control.
var gridCtrl = control.findByXmlNode("Grid1");
if (gridCtrl.source && gridCtrl.sourceData)
{
return gridCtrl.sourceData;
}
return [];
```

When the Behavior property of Data Grid control is set as Data Source

```
// Grid1 is the Data Grid whose values are used in the options of the control.
return control.findById("Grid1").source;
```

Note: We recommend to set the **Execution Mode** property of **Data Grid** control to **Synchronous** when the **Behavior** property of **Data Grid** control is set to **Data Source** and used in **Options** script.

To add options using Ajax function

You can add options for a **Check Box**, **Drop-down**, **Radio Button**, or **List** control using the ajax (Asynchronous JavaScript and XML) function.

Example

If you want to add options, which include country name and ID, to the **Country** (D1) control, then first code the options information in the JSON attribute-value pair format as follows for the following options text and value, and save the file as **CountryName.json** in the **BPMUITEMPLATES\Default** folder.

```
[{"Name":"India","ID":"1"},{"Name":"UK","ID":"2"},{"Name":"USA","ID":"3"}]
```

Options Text Parameter	Options Value Parameter
Name	ID

Options Text	Options Value
India	1
UK	2
USA	3

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Then, code the script as follows for the **Options** property of the **Country** (D1) control:

```
var resultData;
$.ajax(
{
type: "GET",
url: "CountryName.json",
contentType: "application/json; charset=utf-8",
async: false
,
dataType: "json",
timeout: 10000, //10 Seconds
// CountryName.json file located in the BPMUITEMPLATES\Default\ folder
// contains options information in the JSON attribute-value pair format.
success: function (result)
{
resultData = result;
// alert("Success Message :: " + result);
},
error: function (err)
{
alert("Failure Message :: " + err);
}
}
);
return resultData;
```

Note:
- If all the parameters in the script are not represented as strings (with double quotes), then the Ajax call may fail in the Azure environment.
- Ajax calls are supported only in synchronous mode.

Behavior

Use scripts for the **Behavior** property to manage records and edit values at run-time on a **Base Form** container.

To manage records and edit values at run-time on a Base Form container

- Code the script as follows for the **Behavior** property of a **Base Form** container:

Task	Script
• Allow a user to add or delete records for the control, and to edit values of the child controls.	`readEditBehaviourEnum.editable;`
• Deny a user to add or delete records for the control, and deny to edit values of the child controls.	`readEditBehaviourEnum.readOnly;`
• Allow a user to add or delete records for the	`readEditBehaviourEnum.allowOnlyAdd;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Script
control, but deny to edit values of the child controls.	

To manage records and edit values at run-time on a Base Form container based on the value of another control

You can manage records and edit values at run-time on a **Base Form** container based on the value of another control.

Example

If you want to manage records and edit values at run-time on the **Employee Details** (B1) container as per the selection in the **Edit Information** (R1) control, then you can code the script for the **Behavior** property of the **Employee Details** (B1) container as follows:

```
if(control.findById("R1").value === "0")
{
readEditBehaviourEnum.editable;
}
else if(control.findById("R1").value === "1")
{
readEditBehaviourEnum.readOnly;
}
else
{
readEditBehaviourEnum.allowOnlyAdd;
}
```

Extended Information

Use scripts for the **Extended Information** property to allow or deny a user to access the extended information (that is columns other than the display and value columns of the result set) based on the selection for a **Data Grid**, **Data Lookup**, **Pop-Up**, or **User Lookup** control.

When a **Data Lookup**, **Data Grid**, or **Hierarchical Grid** is mapped to a **Web API Lookup**, the JSON object is returned as a JSON string.

To allow or deny a user to access the extended information based on the selection for the control

- Code the script as follows for the **Extended Information** property of a **Data Grid** or **Data Lookup** control:

Task	Script
• Allow a user to access the extended information based on the selection for the control.	`return true;`
• Deny a user to access the extended information based on the selection for the control.	`return false;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To get extended information based on the selection on data change event

Example

Data Lookup

```
// Get extended information based on the selection on data change event.
var extendedInformation = control.extendedInformation;
if(extendedInformation !== null && extendedInformation !== undefined)
{
// Here Property Name is replaced with EmployeeName, which is the column name for the
database lookup.
control.findById("T1").value = extendedInformation["Title"];
//Alternatively use extendedInformation.EmployeeName;
}
```

Data Grid

```
// Get extended information based on the selection on data change event.
var rowData = control.findById("G1").extendedInformation; // Array
if(rowData.length > 0)
{
control.findById("T1").value = rowData[0].ID;
}
else
{
control.findById("T1").value = ""; // Undefined
}
```

Note: The DateTime column types would contain DateTime values (not String values), when the **Use Culture Specific Values** property is set to **No**.

Hierarchical Grid

```
var ex = control.extendedInformation;
var jobs = ex[0].WorkOrder.Job;// if multiple jobs is selected then Job will be array
if ($.isArray(jobs))
{
control.findById("T1").value ="";
for (var i = 0; i < jobs.length; i++)
{
control.findById("T1").value = control.findById("T1").value + " " + jobs[i].oper_id;
}
}
else
{
control.findById("T1").value = jobs.oper_id;
}
```

Note: The lookup names used in the hierarchical grid are WorkOrder and Job.

User Lookup

```
// Get extended information based on the selection on data change event.
// Property Names for Active Directory Provider.
var extendedInformation = control.extendedInformation;
if(extendedInformation !== null || extendedInformation !== undefined)
{
control.findById("T1").value = extendedInformation["UserDepartment"];
control.findById("T2").value = extendedInformation["UserDesignation"];
control.findById("T3").value = extendedInformation["UserDisplayName"];
control.findById("T4").value = extendedInformation["UserDistinguishedName"];
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.findById("T5").value = extendedInformation["UserEmail"];
control.findById("T6").value = extendedInformation["UserId"];
control.findById("T7").value = extendedInformation["UserManager"];
control.findById("T8").value = extendedInformation["UserName"];
control.findById("T9").value = extendedInformation["UserRole"];
}
```

```
// Using User Lookup extended information in scripts of other controls.
var userLookupControl = control.findById("U1");
if(userLookupControl.value !== "")
{
var userLookupExtendedInfo = userLookupControl.extendedInformation;
if(!SFU.isUndefined(userLookupExtendedInfo))
{
return userLookupExtendedInfo.UserEmail;
}
}
```

Note:
- In `extendedInformation["Property Name"]`, replace `Property Name` with column name for database lookup and XML Node for list lookup.

On Click

Use scripts for the **On Click** event of **Button** and **Card** controls, with the **Button Type** property set to **Custom** or **Navigate To**, to accomplish a desired operation.

You can code a script to save the details passed in the current form, on the **On Click** event of a **Button** control, of the **Button Type** "**Navigate To"**. You can then code anther script to retrieve this information in the embedded form.

Note: The **Navigate To** option uses the **SFU.getFormUrl** API to navigate to a different form. If the API returns an empty value for any reason, then the **On Click** event does not get executed and the navigation to another form does not happen.

Example

If you want to submit a form on click of the **Register** (B1) button with the **Button Type** property set to **Custom**, then you can code the script for the **On Click** event of the **Register** (B1) button as follows:

```
// Submit the form.
control.submitForm(event);
```

On Data Bound

Use scripts for the **On Data Bound** event of a **Data Grid** control to accomplish a desired operation when data is bound to the **Data Grid** control. For a Hierarchical Data Grid, On Data Bound is triggered on expansion of each row.

Data Grid and Hierarchical Data Grid

If you want to select the second record in the **Projects** (G1) control when data is bound to the **Projects** (G1) control, then you can code the script for the **On Data Bound** event of the **Projects** (G1) control as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// Select the second record in the Projects data grid.
var kGrid = e.sender;
var row = kGrid.tbody.find("tr").eq(1);
kGrid.select(row);
```

Note:
- The **On Data Bound** event of a **Data Grid** control is triggered every time the page is updated or when the grid is sorted or filtered.
- The **On Data Bound** event of a **Data Grid** control is triggered every time the grid is viewed in mobile view or in a mobile device.
- The **On Data Bound** event is fired only once on row expansion while on the same page.

Tree

If you want to change the text of the first node in the Projects (T1) control when data is bound to the Projects (T1) control, then you can code the script for the **On Data Bound** event of the Projects (T1) control as follows:

```
//Change the text of the first node in the tree
data[0].text = "Node 1";
```

For more information, see Show Icons in Data Grid Columns.

On Data Change

Use scripts for the **On Data Change** event of a control to accomplish a desired operation when the value of the control changes.

Use the `currentValue` variable to fetch the changed value of the control, and the `oldValue` variable to get the value which was replaced with the current value.

Examples

Consider three Text controls, FirstName (T1), SecondName (T2), and FullName (T3). If you want to display the full name (T3), everytime the FirstName (T1) and SecondName (T2) are updated, then you can code the script for the **On Data Change** event of T1 and (T2) control as follows:

On Data Change event for T1:

```
control.findByXmlNode("FullName").value=currentValue+"
"+control.findByXmlNode("SecondName").value;
```

On Data Change event for T2:

```
control.findByXmlNode("FullName").value=control.findByXmlNode("FirstName").value+"
"+currentValue;
```

Note: The currentValue is of type string for all controls except the **Number** control, which is of type number.

If you want to show the **Employee Name** (T2) when you enter the **Employee ID** using a Web API, then you can code the script for the **On Data Change** event of the **Employee ID** (T1) control as follows

```
// Use Web API to retrieve the Employee Name using the captured Employee ID.
// Here currentValue is passed as parameter.
$.ajax(
{
type: "GET",
url: "http://emp.corp.com/api/Employee/"+currentValue,
contentType: "application/json; charset=utf-8",
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
async:false,
dataType: "json",
success: function (result)
{
control.findById("T2").value = result[0].EmployeeName;
},
error: function (err)
{
}
}
);
```

Accessing old values

If the value of the check box **Job State** (**C1**) changes on an event and if you want to access the old value, then you can code the script for the **On Data Change** event of the **Job State** (**C1**) control as follows:

```
var oldValues = oldValue.split(';#');
var newValues = currentValue.split(';#');
var stateAdded = newValues.filter(function(n) {
return oldValues.indexOf(n) == -1;
});
var alertBody = "'" + stateAdded + "' State added for the Job."
SFU.showAlert("Job State Changed", alertBody,
function()
              {
                //Enter code to perform actions after the alert is closed.
              });
```

Data Grid

If you want to show the value from the the currently checked record in the **Projects Backlog** (G1) control for the **Projects Approved** (G2) control, then you can code the script for the **On Data Change** event of the **Projects Backlog** (G1) control as follows.

```
control.findById("G2").value = control.findById("G1").value;
```

If you want to show the value from the columns **Id**, **Title**, and **Application** of the currently checked record in the **Projects** (G1) control for the **Project ID** (T1), **Project Title** (T2), and **Project Application** (T3) controls respectively, then you can code the script for the **On Data Change** event of the **Projects** (G1) control as follows

```
var gridControl = control.findById("G1");
var rowData = SFU.getGridValue(gridControl);
control.findById("T1").value = rowData[0].<Lookupname>.Id;
control.findById("T2").value = rowData[0].<Lookupname>.Title;
control.findById("T3").value = rowData[0].<Lookupname>.Application;
```

Note:

- The DateTime column types would contain DateTime values (not String values), when the **Use Culture Specific Values** property is set to **No**.

- Data Grid with **Execution Mode** as **Synchronous** will not fire an **On Data Change** event when the form is loading. In case an **On Data Change** event is required to be fired, below are the options:

Option 1: Set the **Execution Mode** as **Asynchronous**.
Option 2: Retain the **Execution Mode** as **Synchronous**. Set the **Auto-load Grid** to **Never**. In the **On Form Load**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

script, write the below script:

```
control.findByXmlNode("GridXmlNode").autoLoadGrid =
skelta.grid.autoLoadGrid.OnFormLoadAndParameterChange;
setTimeout(function(){
control.findByXmlNode("GridXmlNode ").refreshGrid();
}, 0);
```

Hierarchical Data Grid

If you want to show the value from the hierarchical data grid , columns of each data source can be accessed as follows:

//Here the WorkOrder is the parent data source with the columns wo_id, wo_desc, and process_id. The Job is the child data source with columns wo_Id, oper_id, and seq_no.

```
var rowData= SFU.getGridValue(control.findById("G1"));
control.findById("T1").value = rowData[0].WorkOrder.wo_id;
control.findById("T2").value = rowData[0].WorkOrder.Job.oper_id;
```

Note:
- The above code is applicable only for one selected value (that is the currently checked record).
- The column names must be one of the selected column names in the data grid.

On Form Load

Use scripts for the **On Form Load** event of a form to accomplish a desired operation when the form loads.

Example

If you want to show an information message when you load the form, then you can code the script for the **On Form Load** event of the form as follows:

```
// Show an information message.
SFU.showAlert("Form Message", "Please fill all fields in the form.");
```

Note: Commented statements in the **On Form Load** script are executed.

On Record Add

Use scripts for the **On Record Add** event of a **Base Form** container to accomplish a desired operation when a record is added to the **Base Form** at run-time.

Example

If you want to show an information message when you add records to the **Employee Details (B1)** container at run-time, then you can code the script for the **On Record Add** event of the **Employee Details (B1)** container as follows:

```
// Show the number of records present.
alert("Number of records present: " + control.records().length);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

On Record Delete

Use scripts for the **On Record Delete** event of a **Base Form** container to accomplish a desired operation when a record is deleted from the **Base Form** at run-time.

Example

If you want to show an information message when you delete records from the **Employee Details (B1)** container at run-time, then you can code the script for the **On Record Delete** event of the **Employee Details (B1)** container as follows:

```
// Show the number of records present.
alert("Number of records present: " + control.records().length);
```

On Selection Change

Use scripts for the **On Selection Change** event of a control to accomplish a desired operation when the selection of the record for the control changes.

Data Grid

If you want to show the value from the column **CountryName** of the currently selected record in the **Countries** (G1) control for the **Country Name** (T1) control, then you can code the script for the **On Selection Change** event of the **Countries** (G1) control as follows:

```
// CountryName is the column name in the Countries data grid.
if (currentRowValue !== undefined)
{
control.findById("T1").value = currentRowValue.CountryName;
}
else
{
control.findById("T1").value = "";
}
```

Note: The DateTime column types would contain DateTime values (not String values), when the **Use Culture Specific Values** property is set to **No**.

Hierarchical Data Grid

If you want to show the value of the column oper_id of the Job data source from the hierarchical data grid ,then you can access it as follows:

```
//Here the WorkOrder is the parent lookup with the columns wo_id, wo_desc, and process_id.
The Job is the child lookup with columns wo_Id, oper_id, and seq_no.
control.findById("T2").value = currentRowValue.WorkOrder.Job.oper_id;
control.findById("T1").value = currentRowValue.WorkOrder.wo_id;
```

Selection Lost Behavior

The current record selection is lost when:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The data grid is sorted or filtered.
- The next or previous page of records is loaded in the data grid.
- The form is reloaded.
- The screen is changed in the mobile devices.
- The screen is switched between web and mobile view.
- The parametrized grid is refreshed.

Note:
- The **currentRowValue** method returns the value for the current selected record without the extended information.
- The value of the **currentRowValue** method is undefined when the form loads and when the record selection is lost.
- In a Data Grid configuration, if **Disable Selection** option is selected, then the **On Selection Change** event will fire only when the row is selected.
It will not fire on paging, sorting, and filtering.

Tree

If you want to show an alert message when a node is selected and output the data of the selected node as the alert message, then you can code the script for the **On Selection Change** event of the control as follows:

```
//Show an alert dialog when a node is selected and output the selected node's data as the alert message
window.alert(JSON.stringify(data));
```

The **On Selection Change** event passes a 'data' parameter which is a JavaScript object that contains the data values bound to the selected node.

On Tab Index Change

Use scripts for the **On Tab Index Change** event of a **Tabs** container to accomplish a desired operation when the tab index changes, that is when the selected tab changes.

Example

If you want to show an information message when you select a tab in the **Employee (T1)** container at run-time, then you can code the script for the **On Tab Index Change** event of the **Employee (T1)** container as follows:

```
// Show the selected tab index.
alert("You are now in tab " + (control.selectedTabIndex + 1));
```

For information on selecting a tab (that is switching to a tab) and determining the the tab index through APIs, see control.selectTab() and control.selectedTabIndex() respectively.

On Check Change

Use scripts for the **On Check Change** event of a **Tree** control to accomplish a desired operation when the state of a checkbox of a node is set or cleared.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example

If you want to show an alert indicating status of the checkbox status of the effected node., then you can code the script for the **On Check Change** event of the control as follows:

```
//Show an alert indicating the checkbox status of the effected node.
var status = isChecked ? "set" : "cleared";
window.alert("node's checkbox has been " + status);
```

Note:
- This event is fired when the checkbox status of every node is modified.
- In cases where the **Inherit Checked State** property is set to **Yes** or the **Checkbox Behavior** property is set to **Single**, then the event is fired once for each effected node as the tree automatically sets or clears the check states.

On Initialize

Use scripts for the **On Initialize** event of **Arc**, **Card, Gauge,** and **Progress Bar** controls to accomplish a desired operation after the control has been initialized, but before it is rendered to the UI. You can therefore change the configuration of the gauge at runtime.

Example

If you want to:

- set the value of the control to 50
- set the minimum and maximum value of the scale to 0 and 180
- set the start and end angle of the scale to -20 and 200
- set the target value to 120
- set the minor and major units of the scale to 15 and 30

then code the script for the **On Initialize** event of the control as follows:

```
control.value=50;
control.scaleMin=0;
control.scaleMax=180;
control.scaleAngleStart=-20;
control.scaleAngleEnd=200;
control.targetValue=120;
control.minorUnit=15;
control.majorUnit=30;
```

Record Visible

Use scripts for the **Record Visible** property to show or hide a record.

To show or hide a record

- Code the script as follows for the **Record Visible** property of a **Base Form** container:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Script	Task
• Show the record	`return true;`
• Hide the record	`return false;`

Example

Consider a **Base Form** container **Products** (**B1)** containing a **Text** control **Product Name (T1**) and a **Number** control **Quantity Available** (**N1**).

If you want to show the record only when **Quantity Available** (**N1**) for a product is more than 500 units, then you can code the script for the **Record Visible** property of the **Base Form** container **Products** (**B1**) control as follows:

```
var productQuantityAvailable = Number(control.findByXmlNode("QuantityAvailable").value);
if(productQuantityAvailable > 500)
{
//// This will show Records having QuantityAvailable more than 500 units
return true;
}
//// This will hide Records having QuantityAvailable less than or equal to 500 units
return false;
```

Range Color

Use scripts for the **Range Color** property to define a range of values with colors.

Example

If you want to define a range of values from 0 to 30 with yellow, 30 to 60 with green, and 60 to 100 with red, then code the script for the **Range Color** property of the control as follows:

```
return [
{
from: 0,
to: 30,
color: 'yellow'
},{
from: 30,
to: 60,
color:'green'
},
{
from: 60,
to: 100,
color:'red'}];
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Containers

This section defines the script guidelines to be followed while using the form containers.

Base Form

Use scripts to set and get values for the properties of the **Base Form** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Retain Space	No	Not Applicable
Display Mode	No	Not Applicable
Records Caption	No	Not Applicable
Record Position	No	Not Applicable
Image Path	No	Not Applicable
Adapt UI	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example
Show Confirmation Message	No	Not Applicable
Confirmation Message	No	Not Applicable
Repeating	No	Not Applicable
Minimum Records	No	Not Applicable
Maximum Records	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Record Add	Yes	For more information, see scripting guidelines for On Record Add property.
On Record Delete	Yes	For more information, see scripting guidelines for On Record Delete property.
ID	No	Not Applicable

Behavior	Yes	**To set the property**
		`control.findById("ID").readEditBehaviour = readEditBehaviourEnum.editable;`
		`control.findById("ID").readEditBehaviour =`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

		`readEditBehaviourEnum.readOnly;`
		OR
		`control.findById("ID").readEditBehaviour = readEditBehaviourEnum.allowOnlyAdd;`
		OR
		To get the value of the property
		`control.findById("ID").readEditBehaviour;`

Default Number of Records	No	Not Applicable
Fast Append	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Behavior	Yes	For more information, see scripting guidelines for Behavior property.
Record Visible	Yes	For more information, see scripting guidelines for Record Visible property.

Note:
- In a Form with Delay Loading enabled, **Chart**, **Slider**, **Card**, and **Hierarchical List** controls inside a container are not supported.
- The **Chart** control loads in the Inbox only when the **Repeating** property of the Base Form is set to **False**.

Updating Column Names in a Base Form with Display Mode set to Grid

You can update the column name for a **Base Form** with **Display Mode** set to **Grid** by updating the **Tag Name** of the control present in the column.

For more details, see Update Column Names in a Base Form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Panel Form

Use scripts to set and get values for the properties of the **Panel Form** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	Yes Readable	**To get the value of the property** `control.findById("ID").width;`
Height	Yes	**To get the value of the property** `control.findById("ID").height`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
	Readable	`;`
Retain Space	No	Not Applicable
Adapt Layout	No	Not Applicable
Show Panel Details	No	Not Applicable
Columns	No	Not Applicable
Tab Order	No	Not Applicable
Image Path	No	Not Applicable
Adapt UI	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
ID	No	Not Applicable
Behavior	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

Behavior	Yes	`return 0;`
		Allow a user to edit values of the child controls.
		`return 1;`
		Deny a user to edit values of the child controls.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Action Bar

Use scripts to configure the **Name**, **Enable**, and **Visible** properties for an Action Item.

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

Tabs

Use scripts to set and get values for the properties of the **Tabs** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description= "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Appearance

Property	Scripting Supported	Script
Visible	Yes Writable	**To set the property** `control.findById("ID").visibl`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script

	Readable	`e = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Default Tab Index	No	Not Applicable
Show Border and Margin	No	Not Applicable
Image Path	No	Not Applicable
Adapt UI	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Tab Index Change	Yes	For more information, see scripting guidelines for On Tab Index Change property.
ID	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

Tab

Use scripts to set and get values for the properties of the **Tab** control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description= "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable

Advanced

Property	Scripting Supported	Example
ID	No	Not Applicable

Scripts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

Controls

This section defines the script guidelines to be followed while using the form controls.

Arc

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Lookup Source	No	Not Applicable
Unit of Measurement	Yes Writable Readable	**To set the property** `control.findById("ID").arcUom="kg";` **To get the value of the property**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

		`control.findById("ID").arcUom;`

Target Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").arcTargetValue= 80;`
		To get the value of the property
		`control.findById("ID").arcTargetValue;`

Default Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").value=50;`
		To get the value of the property
		`control.findById("ID").value;`

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Retain Space	No	Not Applicable

Title	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").arcTitle = "Value for Title";`
		To get the value of the property
		`control.findById("ID").arcTitle;`

Decimal Places	Yes Writable Readable	**To set the property** `control.findById("ID").arcDecimalPlaces=3;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

		To get the value of the property `control.findById("ID").arcDecimalPlaces;`
Show Scale	No	Not Applicable

Scale Minimum Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").arcScaleMinValue=0;`
		To get the value of the property
		`control.findById("ID").arcScaleMinValue;`

Scale Maximum Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").arcScaleMaxValue=0;`
		To get the value of the property
		`control.findById("ID").arcScaleMaxValue;`

Scale Start Angle	No	Not Applicable
Scale Span Angle	No	Not Applicable
Control Background Color	No	Not Applicable
Background Color	No	Not Applicable
Bar Color	No	Not Applicable

Range	Yes
	Writable
	Readable
		To set the property
		`control.findById(“ID”).options=[{optionvalue:20,foreColor:"red"},{optionvalue:60,foreColor:"green"}];`
		To get the value of optionvalue property of the first element
		control.findById(“ID").options[0].optionvalue
		To get the value of foreColor property of the first element
		`control.findById(“ID").options[0].foreColor`
		Note: The **Options** property exposed in scripting is used for the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

		Range property.
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Initialize	Yes	For more information, see scripting guidelines for **On Initialize**.
ID	No	Not Applicable
Enable Refresh	No	Not Applicable
Refresh Rate	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;`
Value	Yes	`return 90;`

Attachment

Use scripts to set and get values for the properties of the **Attachment** control.

Basic

Property	Scripting Supported	Example
Name	Yes Writable	**To set the property** `control.findById("ID").tagName = "Value for Name";`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	Readable	**To get the value of the property** `control.findById("ID").tagName;`
XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Retain Space	No	Not Applicable
File Name Length	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Validation

Property	Scripting Supported	Example
Attachment Type	No	Not Applicable
Custom Attachment Type	No	Not Applicable
Show Delete Confirmation Message	No	Not Applicable
Delete Confirmation Message	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Storage Type	No	Not Applicable
Persist File Name	No	Not Applicable
Persist Custom File Name	No	Not Applicable
File Path Template	No	Not Applicable
Allow Multiple Attachments	No	Not Applicable
Maximum Attachments	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

		For more information, see scripting guidelines for Visible property.
File Path Template	Yes	`return "<StoragePath>\\" + control.findByXmlNode("Department").value + "\\<GUID>_<FileName>.<FileExtension>";` For example, if the attachment storage path is *C:\Program files\AVEVA\Work Tasks\ Attachments*, the attached file is **Keywords.doc**, and the department is **Finance**, then at run time the file path is set to *C:\Program files\AVEVA\Work Tasks\\Attachments\Finance\ 527ee1d8-2c01-4d0a-b3c1-a8b121ca020b\Keywords.doc.*

Note:

- We do not recommend editing values using APIs.

- You can retrieve the file name via script as follows:

```
var fileListArray = control.findById("ID").fileList;
if (fileListArray.length > 0)
{
return fileListArray[0].name(); // Gives the file name.
}
else
{
return “”;
}
```

- You can retrieve the file via script as follows:

```
var fileListArray = control.findById("ID").fileList;
if (fileListArray.length > 0)
{
return fileListArray[0].fileObject(); // Gives the file object.
}
else
{
return “”;
}
```

- The **fileList** represents an array of **fileDetails**. You can access each file’s details using the above syntax.

- Multiple files can be attached only when the **Storage Type** property is set to **File System**. You can retrieve file paths of uploaded files via script as follows:

```
var filePathsSeparatedByPipeCharacters = control.findById("ID").skFilePath; // In case of multiple files, the file paths in this property are separated by a pipe character ‘|’.
```

- The **skFilePath** property is not supported in the Script Editor's intelligent code completion.

- You can empty the attachment control value via script as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.findById("ID").skFilePath = “”;
control.findById("ID").value = “”;
```

Boolean

Use scripts to set and get values for the properties of the **Boolean** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes Writable Readable	**To set the property** `control.findById("ID").visible = true;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
		To get the value of the property `control.findById("ID").visible;`
Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Retain Space	No	Not Applicable
Options Direction	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

Default Value	Yes	`return "True";`
		OR
		`return "False";`
		Note: The **Value** script takes precedence over the **Default Value** script.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

Value	Yes	`return "True";`
		OR
		`return "False";`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Note:
- Display value is readable. You can retrieve the display value via script as follows:
return control.findById("ID").displayValue;
- **Enable** and **Visible** properties hold their value as true or false. However, the **Value** property has a string value, which can be either **"True"** or **"False"**.

Button

Use scripts to set and get values for the properties of the **Button** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Tooltip	No	Not Applicable

Button Text	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").buttonText = "Value for Button Text";`
		To get the value of the property
		`control.findById("ID").buttonText;`

Button Type	No	Not Applicable

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Alignment	No	Not Applicable
Retain Space	No	Not Applicable
Display Mode	No	Not Applicable
Button Size	No	Not Applicable
Image Path	No	Not Applicable
Image Position	No	Not Applicable
Show as Link	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example
Show Confirmation Message	No	Not Applicable
Confirmation Message Text	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Click	Yes	For more information, see scripting guidelines for On Click property.
ID	No	Not Applicable
Form Button	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

Note:
You can identify the last button clicked on a form through script as follows:

```
control.topLevelForm.lastButtonClickedText === "Name of the Button Text";
```

OR

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.topLevelForm.lastButtonClickedId === "Name of the Button ID";
```

This script is helpful when the control is used with the **Possible Distinct Outputs** property for **Invoke Workflow** activity of the workflow.

For example, if you want to enforce a user to enter comments when rejecting a form, then the following script can be coded for the **Mandatory** property of the **Text** control.

```
return (control.topLevelForm.lastButtonClickedText === "Reject")?true:false;
```

OR

```
return (control.topLevelForm.lastButtonClickedId === "B2")?true:false;
```

Card

You can use scripts to set and get values for the properties of the Card control.

Basic

Property	Scripting	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Lookup Source	No	Not Applicable
Enable Dynamic Label	No	Not Applicable
HeaderValue Left	Yes Readable	**To get the value of the property** `control.findByXmlNode("Card1").headerValueLeft;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

HeaderValue Right	Yes Readable	**To get the value of the property** `control.findByXmlNode("Card1").headerValueRight;`
Value (1-5)	Yes Readable	**To get the value of the property** `control.findByXmlNode("Card1").value1;`
Value (1-5) Label	Yes Readable	**To get the value of the property** `control.findByXmlNode("Card1").value3Label;`

Appearance

Property	Scripting Supported	Example

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Width	Yes Readable	**To get the value of the property** `control.findByXmlNode("Card1").width;`
Height	Yes Readable	**To get the value of the property** `control.findByXmlNode("Card1").height;`
Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Retain Space	No	Not Applicable

Image Path	Yes
	Writable
		To set the property
		`var setimg = "https://www.image.com/image.png";`
		`control.CardImage(setimg);`

HeaderValue Left Display Limit	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
On Click	Yes	For more information, see scripting guidelines for On Click.
ID	Yes Readable	
Enable Lookup Color	No	Not Applicable
Color	No	Not Applicable
Enable Refresh	No	Not Applicable
Refresh Rate	No	Not Applicable
Enable Dynamic Action Buttons Label	No	Not Applicable
Enable Action (1-3)	No	Not Applicable
ActionButton (1-3) Label	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes Writable	`return "Value for Name";`
Description	Yes Writable	`return "Value for Description";`
Visible	Yes Writable Readable	`return true;`

Action Script (1-3)	Yes	**To show the Pop-Up control**
		`var popupCtr =control.findByXmlNode("Popup form1");`
		`SFU.showPopupControl(popupCtr);`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example Scripting Scenarios and Rendering

The following sections provide example scenarios where scripting is used with the Card control. It also has example configuration with its corresponding Card control rendering.

Scripting Scenarios

Example scripts provided in this section can be used independently in the Card control's **On Click** event and **Action Script** properties.

Scenario 1: Calling a popup

Use the following script to call a popup through the **On Click** event property in the **Advanced** tab of the control properties page.

```
// Shows the Pop-Up control.
var popupCtr = control.findByXmlNode("Popupform2");
SFU.showPopupControl(popupCtr);
```

Scenario 2: Calling a workflow

Use the following script to call a workflow through the **Action Script** properties in the **Scripts** tab of the control properties page.

```
// Triggers the Click event of the invoke workflow control.
var gridControl = control.findByXmlNode("Grid1");
if(gridControl)
{
gridSelectedValue = gridControl.selectedValue;
if(gridSelectedValue)
{
control.findByXmlNode("hfGridEntID").value = gridSelectedValue.entID;
}
}
var invokeWorkflowControl = control.findById("I1");
SFU.invokeWorkflow(invokeWorkflowControl);
```

Scenario 3: Calling a form

Use the following script to call a form through the **Action Script** properties in the **Scripts** tab of the control properties page.

```
// Gets the form that has to be embedded in the current form.
var formsURL=SFU.getFormUrl("MDMTestForm","1");
control.findById("E1").url = formsURL;
```

Example Card Control Rendering

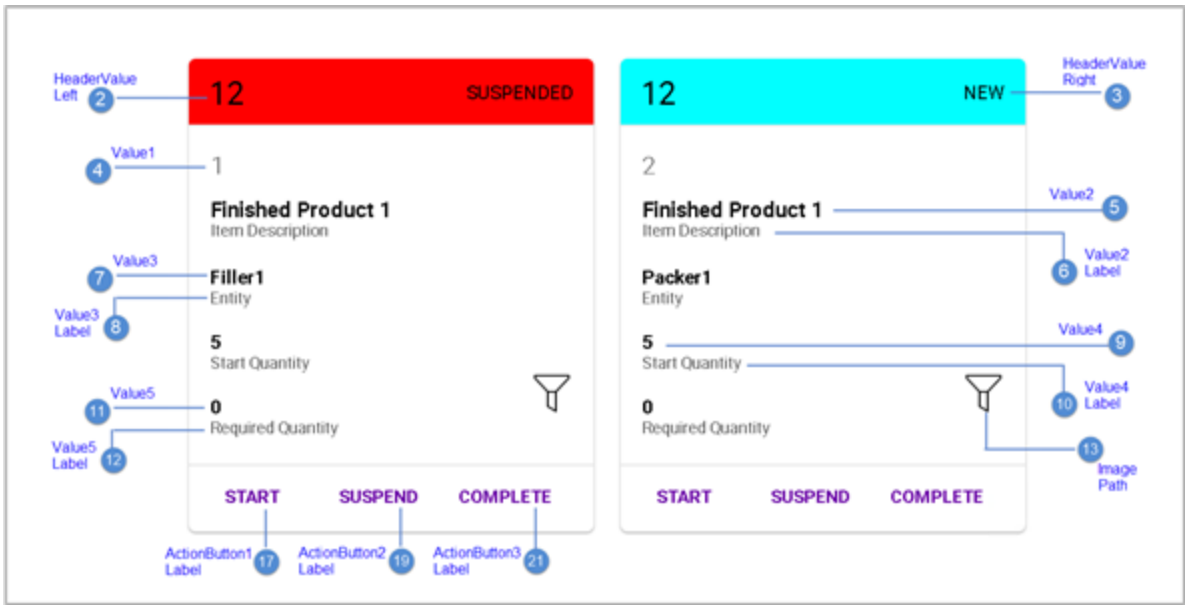
The following rendering example for Card control includes its corresponding configuration. The configured properties are listed in the table and the rest of the properties that are kept to their default values are no longer mentioned.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Basic	Appearance	Advanced
1. **Lookup Source**: `JobqueueLookup` 2. **HeaderValue Left**: `wo_id` 3. **HeaderValue Right**: `state_desc_h` 4. **Value1**: `oper_id` 5. **Value2**: `item_desc` 6. **Value2 Label**: `Item Description` 7. **Value3**: `ent_name` 8. **Value3 Label**: `Entity` 9. **Value4**: `qty_at_start` 10. **Value4 Label**: `Start Quantity` 11. **Value5**: `qty_reqd` 12. **Value4 Label**: `Required Quantity`	13. **Image Path**: `https://img.icons8.com/ios/2x/filter.png`	14. **Enable Lookup Color**: `Yes` (*selected*) 15. **Color:** `color_h` *(column in the Web API)* 16. **Enable Action1**: `Yes` (*selected*) 17. **ActionButton1 Label**: `Start` 18. **Enable Action2**: `Yes` (*selected*) 19. **ActionButton2 Label**: `Suspend` 20. **Enable Action3**: `Yes` (*selected*) 21. **ActionButton3 Label**: `Complete`

The displayed cards depend on the number of records in the lookup data source that you mapped. You can set a script in each of the action buttons using the **Action Script** properties in the **Scripts** tab.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Chart

Use scripts to set and get values for the properties of the **Chart** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Lookup Source	No	Not Applicable

Appearance

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Chart Type	No	Not Applicable
Show Axis Titles	No	Not Applicable
Show Legend	No	Not Applicable
Sankey Chart Node Unit	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Initialize	Yes	For more information, see the scripting guidelines for On Initialize.
ID	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

Check Box

Use scripts to set and get values for the properties of the **Check Box** control.

Basic

Property	**Scripting Supported**	**Example**

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBound To;`
Description	Yes Writable Readable	**To set the property** `control.findById("ID").description = "Value for Description";`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		To get the value of the property `control.findById("ID").description;`
Tooltip	No	Not Applicable
Options	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Options Per Row	No	Not Applicable
Options Direction	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Text Parameter	Yes Writable	**To set the property**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

	Readable	`control.findById("ID").optionText = "OText";`
		
		To get the value of the property
		
		`control.findById("ID").optionText;`

Value Parameter	Yes
	Writable
	Readable
		To set the property
		
		`control.findById("ID").optionValue = "OValue";`
		
		To get the value of the property
		
		`control.findById("ID").optionValue;`

Forms 2011 Compatible	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.

Default Value	Yes	`return "ValueOne";`
		
		OR
		
		`return "ValueOne;#ValueTwo;#ValueThree";`
		
		Note:
		- If the value is not one of the values set for the Value parameter in the Options property, then that value will not be a part of the options but will be a part of the XML instance.
		- Multiple values must be separated with ;#.
		- The Value script takes precedence over the Default Value script.

Value	Yes	`return "ValueOne";`
		
		OR
		
		`return "ValueOne;#ValueTwo;#ValueThr`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		`ee";`

Validate	Yes	return new ValidationOptions(false, "Error Message");
		
		OR
		
		return new ValidationOptions(true, "");
		
		For more information, see scripting guidelines for Validate property.

Options	Yes	`return "[{\"OValue\":\"1\",\"OText\":\"ValueOne\"},{\"OValue\":\"2\",\"OText\":\"ValueTwo\"},{\"OValue\":\"3\",\"OText\":\"ValueThree\"}]";` Here OText is the Text Parameter and OValue is the Value Parameter. For more information, see scripting guidelines for Options property. **Note:** - Text Parameter and Value Parameter properties must be specified before using the Options property script. - Fore Color and Back Color properties cannot be set through scripts.

Note:
- Display value is readable.
- You can retrieve the display value via script as follows:

```
return control.findById("ID").displayValue;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Data Grid

Use scripts to set and get values for the properties of the **Data Grid** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable
Configuration Name	No	Not Applicable

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Read-Only	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").readOnly = true;`
		To get the value of the property
		`control.findById("ID").readOn`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
		`ly;`
Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Rearrange Columns	No	Not Applicable
Resize Columns	No	Not Applicable
Retain Space	No	Not Applicable
Multiple Selection	No	Not Applicable
Show Export Button	No	Not Applicable
Export Type	No	Not Applicable
Export All Pages	No	Not Applicable
Adapt UI	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
On Data Bound	Yes	For more information, see scripting guidelines for On Data Bound property.
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
On Selection Change	Yes	For more information, see scripting guidelines for On Selection Change property.
ID	No	Not Applicable
Behavior	No	Not Applicable
Page Size	No	Not Applicable
Extended Information	No	Not Applicable
Select First Record	No	Not Applicable
Auto-select Record	No	Not Applicable
Summary Columns	No	Not Applicable

Auto-load Grid	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").autoLoadGrid = skelta.grid.autoLoadGrid.OnFormLoadAndParameterChange;`
		`control.findById("ID").autoLoadGrid = skelta.grid.autoLoadGrid.OnFormLoad;`
		`control.findById("ID").autoLoadGrid = skelta.grid.autoLoadGrid.Never;`
		To get the value of the property
		`control.findById("ID").autoLoadGrid;`

Include in Summary	No	Not Applicable
Use Culture Specific Values	No	Not Applicable

Scripts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Read-Only	Yes	`return false;` For more information, see scripting guidelines for Read-Only property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.

Value	Yes	`return "%3CGrid%3E%3CData%3E%3CCountryLP%3E%3CCountryId%3E%3C!%5BCDATA%5B10%5D%5D%3E%3C%2FCountryId%3E%3CCountryName%3E%3C!%5BCDATA%5BIN%5D%5D%3E%3C%2FCountryName%3E%3C%2FCountryLP%3E%3C%2FData%3E%3C%2FGrid%3E";`
		Note:
		- The value represents the records whose check box is selected. To clear check box of all records, set the value to an empty string.
		- The return value must be URL encoded.
		- For example, `"<Grid><Data><CountryLP><CountryId><![CDATA[10]]></CountryId><CountryName><![CDATA[IN]]></CountryName></CountryLP></Data></Grid>"`
		
		where **CountryLP** is the Lookup name, **CountryId** and **CountryName** are columns persisted in the grid configuration is URL encoded as
		
		`"%3CGrid%3E%3CData%3E%3CCountryLP%3E%3CCountryId%3E%3C!%5B`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		CDATA%5B10%5D%5D%3E%3C%2FCountryId%3E%3CCountryName%3E%3C!%5BCDATA%5BIN%5D%5D%3E%3C%2FCountryName%3E%3C%2FCountryLP%3E%3C%2FData%3E%3C%2FGrid%3E"

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Improving Performance of Parameterized Data Grid

The behavior of parameterized data grids on data change of parameters with Refresh On Parameter Change property is as follows.

Form Mode	Property Value	Control Behavior
New Mode (Fill Form, Preview Form)	Yes	Data grid is refreshed when a parameter is changed.
	No	Data grid is not refreshed when a parameter is changed. To refresh the data grid when the last parameter is changed, code the script for the **On Data Change** event of the last parameter as follows: `control.findById("G1").refreshOnParamChange = true;`
Edit Mode (Inbox)	Yes	Data grid is refreshed when a parameter is changed.
	No	Data grid is not refreshed when a parameter is changed. To refresh the data grid when a parameter is changed, code the script for the **On Data Change** event of that parameter as follows: `var editMode =`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Form Mode	Property Value	Control Behavior

		`SFU.isFormInEditMode();`
		`if(editMode)`
		`{`
		`control.findById("G1").refreshOnParamChange = true;`
		`}`

For more information, see Show Icons in Data Grid Columns.

Note:
- Display value is readable.
- You can retrieve the display value via script as follows:

```
return control.findById("ID").displayValue;
```

Data Lookup

Use scripts to set and get values for the properties of the **Data Lookup** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable
Lookup Source	No	Not Applicable

Appearance

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Read-Only	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").readOnly = true;`
		To get the value of the property
		`control.findById("ID").readOnly;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Watermark	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Drop-Down Height	No	Not Applicable
Adapt UI	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Autocomplete Length	No	Not Applicable
Show Confirmation Message	No	Not Applicable
Confirmation Message Text	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Default Number of Records	No	Not Applicable
Retain Selected Value	No	Not Applicable
Persist Display Value	Yes Readable	**To get the value of the property** `control.findById("ID").displayValue;`

Extended Information	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").requiresExtendedInformation = true;`
		To get the value of the property
		`control.findById("ID").requiresExtendedInformation;`
		For more information, see Extended Information.

Include in Summary	No	Not Applicable

Scripts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Read-Only	Yes	`return false;` For more information, see scripting guidelines for Read-Only property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.
Value	Yes	`return "Value for the control";`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Date

Use scripts to set and get values for the properties of the **Date** control.

Basic

Property	Scripting Supported	Example
Name	Yes	**To set the property**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

	Writable
	Readable
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Read-Only	Yes Writable Readable	**To set the property** `control.findById("ID").readOnly = true;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
		To get the value of the property `control.findById("ID").readOnly;`
Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Minimum Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").min = new Date("01/01/1901 00:00:00");`
		To get the value of the property
		`control.findById("ID").min;`

Maximum Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").max = new Date("12/31/2099 23:59:59");`
		To get the value of the property
		`control.findById("ID").max;`

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		property.
ID	No	Not Applicable
Date Format	Yes Readable	**To get the value of the property** `control.findById("ID").dateFormat;`
Forms 2011 Compatible	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, scripting guidelines for Visible property.
Read-Only	Yes	`return false;` For more information, scripting guidelines for Read-Only property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.
Default Value	Yes	`return "12/31/2099 00:00:00";` **Note:** The **Value** script takes precedence over the **Default Value** script. The format of the date should be MM/dd/yyyy 00:00:00.
Value	Yes	`return "12/31/2099 00:00:00";` **Note:** The format of the date

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		should be MM/dd/yyyy 00:00:00.

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Note:
- Display value is readable.
- You can retrieve the display value via script as follows:

```
return control.findById("ID").displayValue;
```

Date Time

Use scripts to set and get values for the properties of the **Date Time** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Tooltip	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Read-Only	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").readOnly = true;`
		To get the value of the property
		`control.findById("ID").readOnly;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
Width	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Minimum Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").min = new Date("01/01/1901 00:00:00");`
		To get the value of the property
		`control.findById("ID").min;`

Maximum Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").max = new Date("12/31/2099 23:59:59");`
		To get the value of the property
		`control.findById("ID").max;`

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Date Format	No	Not Applicable
Time Format	No	Not Applicable
Time Interval	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	return true; For more information, see scripting guidelines for Visible property.
Read-Only	Yes	`return false;` For more information, see scripting guidelines for Read-Only property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Default Value	Yes	`return "12/31/2099 00:00:00";` **Note:** The **Value** script takes precedence over the **Default Value** script. The format of the datetime should be MM/dd/yyyy HH:mm:ss.
Value	Yes	`return "12/31/2099 00:00:00";` **Note:** The format of the datetime should be MM/dd/yyyy HH:mm:ss.

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Example

If you want to show date and time in the **Formatted Time** (D1) Date Time control when you enter date and time in the **Entered Time** (T1) Text control, then you can code the script for the **On Data Change** event of the **Entered Time** (T1) Text control as follows:

```
// Show date and time in the Date Time control from the Text control.
// The date and time in the Text control is not converted to UTC.
var dateTime = new Date();
var dateTimeUtc = SFU.getDateTimeInServerUTCFormat(dateTime);
control.findById("D1").value=SFU.getDateTimeInStringFormat(dateTimeUtc,"MM/dd/yyyy
HH:mm:ss");
```

Note:
- Display value is readable.
_ You can retrieve the display value via script as follows:
`return control.findById("ID").displayValue;`

Drop-Down

Use scripts to set and get values for the properties of the **Drop Down** control.

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable

Options Caption	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").optionCaption = "Value for Options Caption";`
		To get the value of the property
		`control.findById("ID").optionCaption;`

Options	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Appearance

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property. **Note:** If you use scripts for **On Data Change property** of a **Drop-Down** control to add a record to a **Base Form**, then the focus will be on the first control of the **Base Form**.
ID	No	Not Applicable

Text Parameter	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").optionText = "OText";`
		To get the value of the property
		`control.findById("ID").optionText;`

Value Parameter	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").optionValue = "OValue";`
		To get the value of the property
		`control.findById("ID").optionValue;`

Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.

Default Value	Yes	`return "ValueOne";`
		OR
		`return "ValueOne;#ValueTwo;#ValueThree";`
		Note:
		- If the value is not one of the values set for the **Value** parameter in the **Options** property, then that value will not be a part of the options but will be a part of the XML instance.
		- Multiple values must be separated with ;#
		- The **Value** script takes precedence over the **Default Value** script.

Value	Yes	`return "ValueOne";`
		OR
		`return "ValueOne;#ValueTwo;#ValueThree";`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Options	Yes	`return "[{\"OValue\":\"1\",\"OText\":\"ValueOne\"},{\"OValue\":\"2\",\"OText\":\"ValueTwo\"},{\"OValue\":\"3\",\"OText\":\"ValueThree\"}]";`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		Here **OText** is the **Text Parameter** and **OValue** is the **Value Parameter**. For more information, see scripting guidelines for Options property. **Note:** -**Text Parameter** and **Value Parameter** properties must be specified before using the **Options** property script. -**Fore Color** and **Back Color** properties cannot be set through scripts.

Note:
- Display value is readable.
- You can retrieve the display value via script as follows:

```
return control.findById("ID").displayValue;
```

Email

Use scripts to set and get values for the properties of the **Email** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").descri`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		`ption;`
Tooltip	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Read-Only	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").readOnly = true;`
		To get the value of the property
		`control.findById("ID").readOnly;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Watermark	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Maximum Length	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").maxLength = 100;`
		To get the value of the property
		`control.findById("ID").maxLength;`

Multiple Email IDs	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMultipleEmail = true;`
		To get the value of the property
		`control.findById("ID").isMultipleEmail;`

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Read-Only	Yes	`return false;` For more information, see scripting guidelines for Read-Only property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.
Default Value	Yes	`return "name@corp.com";` **Note:** The **Value** script takes precedence over the **Default Value** script.
Value	Yes	`return "name@corp.com";`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Embed Form

Use scripts to set and get values for the properties of the **Embed Form** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagNam`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		`e;`
XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Persist Form Definition	No	Not Applicable
Form Details	No	Not Applicable

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Show Panel Details	No	Not Applicable
Adapt UI	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
ID	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

Note: Syntax Check for the scripts written for the controls inside the **Embed Form** control is not supported. A warning message appears in the script editor, when syntax check is performed and also whenever the scripts are saved. However, you can proceed to save the scripts.

Best Practices

- We recommend using unique XML nodes for controls in source forms to aid in scripting.
- We recommend using the `control.findByXmlNode()` method instead of the `control.findById()` method in source forms to avoid conflict of IDs between forms

Scenario 1 – Updating Value of T2 (Present inside Embed Form) from a control inside Embed Form

```
control.findByXmlNode("T2").value = "SomeValue";
```

Scenario 2 – Updating Value of T2 (Present Outside Embed Form) from a control inside Embed Form

```
Control.topLevelForm.findByXmlNode("T2").value = "SomeValue";
```

Scenario 3 – Updating Value of T2 (Present Inside Embed Form) from a control Outside Embed Form

```
control.findByXmlNode("EmbedFormXmlNode").findByXmlNode("T2").value= "SomeValue";
```

Embed Page

Use scripts to set and get values for the properties of the **Embed Page** control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

URL	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").url = "http://www.corp.com";`
		To get the value of the property
		`control.findById("ID").url;`

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Show Scrollbars	No	Not Applicable
Retain Space	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	On Data Change script always works with the value script for the Embed Page. For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
URL	Yes	`return "http://www.corp.com";`

E-signature

Use scripts to set and get values for the properties of the **E-signature** control.

Basic

Property	Scripting Supported	Example
Name	Yes Writable	**To set the property** `control.findById("ID").tagNam`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	Readable	`e = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description= "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Retain Space	No	Not Applicable
Show Header	No	Not Applicable
Signed By Text	No	Not Applicable
User Name Text	No	Not Applicable
Password Text	No	Not Applicable
Signed On Text	No	Not Applicable
Comments Text	No	Not Applicable
Image Path	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Adapt UI	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example
Comments Required	No	Not Applicable

Advanced

Property	Scripting Supported	Example
ID	No	Not Applicable
Date Format	No	Not Applicable
Time Format	No	Not Applicable
User Mode	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

Gantt Chart

Basic

Property	**Scripting Supported**	**Example**
Name	Yes Writable	**To set the property** `control.findById("ID").tagName =`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

	Readable	`"Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Lookup Source	No	Not Applicable
Event State	No	Not Applicable
Event Start Time	No	Not Applicable
Event End Time	No	Not Applicable
Event Name	No	Not Applicable
Event Color	No	Not Applicable

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
Retain Space	No	Not Applicable
Bar Thickness	No	Not Applicable

Time Period	Yes	**To set the property**
		For Duration
		`control.findById("Gantt1").ganttDuration = 8;`
		For Time Range scripting using Text control
		If you want to set the **Time Range's** start and end time properties in the **Gantt** (G1) control when you enter date and time in the **Entered Time** (T1) Text control, use the following script for the **On Data Change** event of the **Entered Time** (T1) Text control.
		`// assuming datetime string in "MM/dd/YYYY HH:mm:ss" format`
		`var str = currentValue;`
		`var d = str.split(" ")[0].split("/");`
		`var t = str.split(" ")[1].split(":");`
		`var dateTime = new Date(parseInt(d[2]), parseInt(d[0]), parseInt(d[1]), parseInt(t[0]), parseInt(t[1]), parseInt(t[2]));`
		`var dateTimeUtc = SFU.getDateTimeInServerUTCFormat(dateTime);`
		`control.findByXmlNode("Gantt1").ganttTimeRangeStartTime =SFU.getDateTimeInStringFormat(dateTimeUtc,"MM/dd/yyyy HH:mm:ss");`
		Note: The **Text** control date time input is a string with the format specified in the example script above and in UTC. This input is displayed in the Gantt Chart control in your local time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
		For Time Range scripting using Date Time control If you want to set the **Time Range's** start and end time properties in the **Gantt** (G1) control when you select date and time in the **Entered Time** (D1) Date Time control, use this script for the **On Data Change** event of the **Entered Time** (D1) Date Time control. `control.findByXmlNode("Gantt1").ganttTimeRangeStartTime = currentValue;`
		To get the value of the property *For Duration* `control.findByXmlNode("Gantt1").ganttDuration;`
Duration	Yes	See Time Period.
Show Details Pane	No	Not Applicable
Show Time Range	No	Not Applicable
Title Position	No	Not Applicable

Title	Yes	**To set the property**
		`control.findById("ID").ganttTitle="Gantt Chart";`
		To get the value of the property
		`control.findById("ID").ganttTitle;`

Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Click	Yes	For more information see scripting guidelines for On Click.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

ID	No	Not Applicable
Enable Refresh	No	Not Applicable
Refresh Rate	No	Not Applicable

Enable Action (1-3)	Yes	**To set the property**
		`control.findById("Gantt1").ganttEnableAction1=enable;`
		To get the value of the property
		`control.findById("Gantt1").ganttEnableAction1;`

Action (1-3)	No	Not Applicable

Scripts

Property	**Scripting Supported**	**Example**
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;`
Value	Yes	`return 90;`

Action (1-3) Script	Yes	**To show the Pop-Up control**
		`var popupCtr =control.findByXmlNode("Popupform1");`
		`SFU.showPopupControl(popupCtr);`
		To read a value of a Web API response field - "log_id"
		`var id = data.log_id;`

Gauge

Use scripts to set and get values for the properties of the **Gauge** control.

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable
Default Value	No	Not Applicable

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Retain Space	No	Not Applicable
Scale Minimum Value	Yes Writable Readable	**To set the property** control.findById("ID").scaleMin = 0; **To get the value of the property**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
		control.findById("ID").scaleMin;

Scale Maximum Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").scaleMax = 0;`
		To get the value of the property
		`control.findById("ID").scaleMax;`

Scale Start Angle	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").scaleAngleStart = 0;`
		To get the value of the property
		`control.findById("ID").scaleAngleStart;`

Scale End Angle	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").scaleAngleEnd = 0;`
		To get the value of the property
		`control.findById("ID").scaleAngleEnd;`

Scale Major Unit	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").majorUnit = 0;`
		To get the value of the property
		`control.findById("ID").majorUnit;`

Scale Minor Unit	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").minorUnit = 0;`
		To get the value of the property
		`control.findById("ID").minorUnit;`

Value Position	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID"). showTargetIndicator=true;`
		To get the value of the property
		`control.findById("ID"). showTargetIndicator;`

Show Target Indicator	No	**To set the property**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script

		`control.findById("ID"). showTargetIndicator=true;`
		To get the value of the property
		`control.findById("ID"). showTargetIndicator;`

Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Initialize	Yes	For more information, see scripting guidelines for On Initialize property.
Range Color	Yes	For more information, see scripting guidelines for Range Color property.
ID	No	Not Applicable

Target Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").target Value= 60;`
		To get the value of the property
		`control.findById("ID").target Value;`

Unit of Measurement	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID"). labelUnitOfMeasure="Sec";`
		To get the value of the property
		`control.findById("ID"). labelUnitOfMeasure;`

Decimal Places (Precision)	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID"). valuePrecision=2;`
		To get the value of the property
		`control.findById("ID"). valuePrecision;`

Scripts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Name	Yes	return "Value for Name";
Description	Yes	return "Value for Description";
Visible	Yes	return true;
Value	Yes	return 90;

Geolocation

Use scripts to set and get values for the properties of the **Geolocation** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Button Text	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").buttonText = "Value for Button Text";`
		To get the value of the property
		`control.findById("ID").buttonText;`

Appearance

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Alignment	No	Not Applicable
Display Mode	No	Not Applicable
Button Size	No	Not Applicable
Image Path	No	Not Applicable
Image Position	No	Not Applicable
Show as Link	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
ID	No	Not Applicable
Form Button	No	Not Applicable

Scripts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

On Geolocation Load	Yes	**To get the value of the property**
		`control.findById('T1').value = control.latitude;`
		`control.findById('T2').value = control.longitude;`

Hidden Field

Use scripts to set and get values for the properties of the **Hidden Field** control.

Basic

Property	Scripting Supported	Example
Name	Yes Readable	**To get the value of the property** `control.findById("ID").tagName;`
XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		property.
ID	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Default Value	Yes	`return "Default value for the control";` **Note:** The **Value** script takes precedence over the **Default Value** script.
Value	Yes	`return "Value for the control";`

Hierarchical List

Use scripts to set and get values for the properties of the **Hierarchical List** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Lookup Source	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
ID	No	Not Applicable
Default Number of Records	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
On Hierarchical List Load	Yes	For example, to apply a filter after the hierarchical list is loaded, code as follows: `control.findByXmlNode("Hierarchicallist1").applyFilter(function(recordValue){ if(recordValue.Progress > 50) { return true; } else { return false; } });`

On Level 1 List Load	Yes	**To set the Level 1 Indicator Color**
		`control.level1IndicatorColor = "#5482ab";`
		OR
		`control.level1IndicatorColor = "rgb(84,130,171)";`
		OR
		`control.level1IndicatorColor = currentRecord.IndiCol;`
		Note: Here, **IndiCol** is the column containing indicator color.
		To set the Level 1 Progress Color
		`control.level1ProgressColor = "#5482ab";`
		OR
		`control.level1ProgressColor = "rgb(84,130,171)";`
		OR
		`control.level1ProgressColor = currentRecord.ProCol;`
		Note: Here, **ProCol** is the column containing progress color.
		To set the Level 1 Progress Value

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

		`control.level1ProgressValue = 70;`
		OR
		`control.level1ProgressValue = currentRecord.ProVal;`
		Note:
		- Here, **ProVal** is the column containing progress value.
		- Progress value should be between 0 and 100.
		- Float values are rounded to the nearest integer.
		To set the Level 1 Progress Description
		`control.level1ProgressDescription = "Value for Description";`
		OR
		`control.level1ProgressDescription = currentRecord.ProDesc;`
		Note: Here, **ProDesc** is the column containing progress description.

On Level 2 List Load	Yes	**To set the Level 2 Indicator Color**
		`control.level2IndicatorColor = "#5482ab";`
		OR
		`control.level2IndicatorColor = "rgb(84,130,171)";`
		OR
		`control.level2IndicatorColor = currentRecord.IndiCol;`
		Note: Here, **IndiCol** is the column containing indicator color.
		To set the Level 2 Progress Color
		`control.level2ProgressColor = "#5482ab";`
		OR
		`control.level2ProgressColor = "rgb(84,130,171)";`
		OR
		`control.level2ProgressColor = currentRecord.ProCol;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

		Note: Here, **ProCol** is the column containing progress color.
		To set the Level 2 Progress Value
		`control.level2ProgressValue = 70;`
		OR
		`control.level2ProgressValue = currentRecord.ProVal;`
		Note:
		- Here, **ProVal** is the column containing progress value.
		- Progress value should be between 0 and 100.
		- Float values are rounded to the nearest integer.
		To set the Level 2 Progress Description
		`control.level2ProgressDescription = "Value for Description";`
		OR
		`control.level2ProgressDescription = currentRecord.ProDesc;`
		Note:
		- Here, **ProDesc** is the column containing progress description.
		- On some devices, progress description with more than 5 characters may get truncated.

On Level 3 List Load	Yes	**To set the Level 3 Indicator Color**
		`control.level3IndicatorColor = "#5482ab";`
		OR
		`control.level3IndicatorColor = "rgb(84,130,171)";`
		OR
		`control.level3IndicatorColor = currentRecord.IndiCol;`
		Note: Here, **IndiCol** is the column containing indicator color.
		To set the Level 3 Progress Color
		`control.level3ProgressColor = "#5482ab";`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

		OR
		`control.level3ProgressColor = "rgb(84,130,171)";`
		OR
		`control.level3ProgressColor = currentRecord.ProCol;`
		Note: Here, **ProCol** is the column containing progress color.
		To set the Level 3 Progress Value
		`control.level3ProgressValue = 70;`
		OR
		`control.level3ProgressValue = currentRecord.ProVal;`
		Note:
		- Here, **ProVal** is the column containing progress value.
		- Progress value should be between 0 and 100.
		- Float values are rounded to the nearest integer.
		To set the Level 3 Progress Description
		`control.level3ProgressDescription = "Value for Description";`
		OR
		`control.level3ProgressDescription = currentRecord.ProDesc;`
		Note:
		- Here, **ProDesc** is the column containing progress description.
		- On some devices, progress description with more than 5 characters may get truncated.

On Level 1 Progress Click	Yes	**To set the property** `SFU.showAlert(currentRecord.ProDesc);` **Note:** Here, **ProDesc** is the column containing progress description. On clicking the progress, the description is shown as an alert message.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

On Level 2 Progress Click	Yes	**To set the property**
		`SFU.showAlert(currentRecord.ProDesc);`
		`SFU.showAlert(parentRecord.ProDesc);`
		Note:
		- Here, **ProDesc** is the column containing progress description. On clicking the progress, the description is shown as an alert message.
		- On some devices, progress description with more than 5 characters may get truncated.

On Level 3 Progress Click	Yes	**To set the property**
		`SFU.showAlert(rootRecord.ProDesc);`
		`SFU.showAlert(currentRecord.ProDesc);`
		`SFU.showAlert(parentRecord.ProDesc);`
		Note:
		- Here, **ProDesc** is the column containing progress description. On clicking the progress, the description is shown as an alert message.
		- On some devices, progress description with more than 5 characters may get truncated.

On Level 2 Toggle	Yes	**To set the property**
		`if(toggleValue == true)`
		`{`
		`control.level1ProgressValue = 100;`
		`}`
		`else`
		`{`
		`control.level1ProgressValue = 50;`
		`}`
		`// Access above siblings`
		`selectItem.aboveSiblingValue = true;`
		`// Access all siblings`
		`selectItem.aboveSiblingValue = true;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

		`// Access below siblings`
		`selectItem.belowSiblingValue = true;`
		`// Select each of the sibling and enable slide-toggle and set the progress color to red`
		`for(var i=0; i<selectItem.siblings.length; i++)`
		`{`
		`selectItem.siblings.toggle[i].level2ToggleEnable = true;`
		`selectItem.siblings.progress[i].progressColor= "red";`
		`}`
		Note: Setting level 3 properties using On Level 2 Toggle script is not supported.

On Level 3 Toggle	Yes	**To set the property**
		`if(toggleValue == true)`
		`{`
		`control.level1ProgressValue = 100;`
		`}`
		`else`
		`{`
		`control.level1ProgressValue = 50;`
		`}`
		`// Access above siblings`
		`selectItem.aboveSiblingValue = true;`
		`// Access all siblings`
		`selectItem.aboveSiblingValue = true;`
		`// Access below siblings`
		`selectItem.belowSiblingValue = true;`
		`// Select each of the sibling and enable slide-toggle and set the progress color to red`
		`for(var i=0; i<selectItem.siblings.length; i++)`
		`{`
		`selectItem.siblings.toggle[i].level3ToggleEnable = true;`
		`selectItem.siblings.progress[i].progressColor= "red";`
		`}`
		Note: Setting level 2 properties

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		using On Level 3 Toggle script is only supported for Enable.

Note: You can specify search criteria on the hierarchical list. For more information, see control.applyFilter() and control.clearFilter() methods.

Hyperlink

Use scripts to set and get values for the properties of the **Hyperlink** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable

Hyperlink	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").displayText = "Value for Hyperlink";`
		To get the value of the property
		`control.findById("ID").displayText;`

URL	Yes	**To set the property**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

	Writable
	Readable
		`control.findById("ID").value = "http://www.corp.com";`
		To get the value of the property
		`control.findById("ID").value;`

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		For more information, see scripting guidelines Visible property.
Value	Yes	`return "http://www.corp.com";`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Image

Use scripts to set and get values for the properties of the **Image** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable
Image Path	Yes Writable	**To set the property** `control.findById("ID").value`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

	Readable	`= "https://www.image.com/image.png";`
		To get the value of the property
		`control.findById("ID").value;`

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Alignment	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Downscale Image	No	Not Applicable
Alternate Text	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Include in Summary	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Value	Yes	`return "https://www.image.com/image.png";`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Invoke Workflow

Use scripts to set and get values for the properties of the **Invoke Workflow** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`
Description	Yes Writable Readable	**To set the property** `control.findById("ID").description = "Value for Description";`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		To get the value of the property `control.findById("ID").description;`
Tooltip	No	Not Applicable
Button Text	Yes Writable Readable	**To set the property** control.findById("ID").buttonText = "Value for Button Text"; **To get the value of the property** control.findById("ID").buttonText;
Workflow Details	No	Not Applicable
XML Variable Name	No	Not Applicable
XML Schema	No	Not Applicable

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Alignment	No	Not Applicable
Retain Space	No	Not Applicable
Display Mode	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
Button Size	No	Not Applicable
Image Path	No	Not Applicable
Image Position	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example
Show Confirmation Message	No	Not Applicable
Confirmation Message Text	No	Not Applicable
Show Success Message	No	Not Applicable
Success Message Text	No	Not Applicable

Advanced

Property	Scripting Supported	Example
ID	No	Not Applicable
Execution Mode	No	Not Applicable
Close Form	No	Not Applicable
Form Button	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Value	Yes	`return "Value for the control";`
Pre-Workflow Execution	Yes	`return false;` This script stops the execution of the workflow. The *Pre-Workflow Execution* script can be used to validate, update values in columns, and so on. **Note:** The workflow is triggered after the *Pre-Workflow Execution* scripts are executed.
Post-Workflow Execution	Yes	`control.findById("ID").value = blockingOutput;` The *Post-Workflow Execution* script can be used to perform actions based on the workflow execution status. **Note:** - The script is executed only for synchronously executed workflows. - The script is executed even when the workflow fails but the corresponding workflow status is returned. - The script is not executed when the workflow engine is not running.

Enabling Read-Only mode for the form while the workflow is processed

When a workflow is triggered through the **Invoke Workflow** control, we recommend to deny the user to edit values of controls on the form while the workflow is being processed.

To enable **Read-Only** mode for the form, you can code the script for the **Pre-Workflow Execution** property of the **Invoke Workflow** control as follows:

```
control.topLevelForm.readEditAddBehaviour = readEditBehaviourEnum.readOnly;
```

To disable **Read-Only** mode for the form after the workflow is processed, you can code the script for the **Post-Workflow Execution** property of the **Invoke Workflow** control as follows:

```
control.topLevelForm.readEditAddBehaviour = readEditBehaviourEnum.editable;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using workflowStatus and blockingOutput Variables

Use the **workflowStatus** variable to accomplish a desired operation based on the workflow status, and the **blockingOutput** variable to accomplish a desired operation using the **Blocking Output** from **Blocking Data Activity** of the workflow. For more information on workflow status codes, see **List of Internal Status Codes** in the *Developer Guide*.

```
// Do something based on the workflow status.
if (workflowStatus === "FN")
{
//Workflow finished without errors.
/* Accomplish a desired operation.
* Read blockingOutput variable if the workflow is expected to return Blocking Output from
Blocking Data Activity.
* Refresh the form or a control if the workflow is supposed to modify data on the form.
*/
}
else if (workflowStatus === "FE")
{
//Workflow finished with errors.
SFU.showAlert("Workflow triggered with errors.");
}
```

Label

Use scripts to set and get values for the properties of the **Label** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description= "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Tooltip	No	Not Applicable

Display Text	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").value= "Value for Display Text";`
		To get the value of the property
		`control.findById("ID").value;`

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Alignment	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Show Line	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
ID	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Value	Yes	`return "Value for the control";`

List

Use scripts to set and get values for the properties of the **List** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable
Options	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Retain Space	No	Not Applicable
Rows	No	Not Applicable
Selection Mode	Yes Readable	**To get the value of the property** `control.findById("ID").selectedMode;`
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property. **Note:** If you use scripts for **On Data Change** property of a **List** control to add a record to a **Base Form**, then the focus will be on the first control of the **Base Form**.
ID	No	Not Applicable

Text Parameter	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").optionText = "OText";`
		To get the value of the property
		`control.findById("ID").optionText;`

Value Parameter	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").optionValue = "OValue";`
		To get the value of the property
		`control.findById("ID").optionValue;`

Forms 2011 Compatible	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.

Default Value	Yes	`return "ValueOne";`
		OR
		`return "ValueOne;#ValueTwo;#ValueThree";`
		Note:
		- If the value is not one of the values set for the **Value** parameter in the **Options** property, then that value will not be a part of the options but will be a part of the XML instance.
		- If multiple values are set when the **Selection Mode** property is set to **Single**, then only the first value will be a part of the options but all values will be a part of the XML instance.
		- Multiple values must be separated with ;#
		- The **Value** script takes precedence over the **Default Value** script.

Value	Yes	`return "ValueOne";`
		OR
		`return "ValueOne;#ValueTwo;#ValueThree";`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Options	Yes	`return "[{\"OValue\":\"1\",\"OText\":\"ValueOne\"},{\"OValue\":\"2\",\"OText\":\"ValueTwo\"},{\"OValue\":\"`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		`3\",\"OText\":\"ValueThree\"} ]";` Here **OText** is the **Text Parameter** and **OValue** is the **Value Parameter**. For more information, see scripting guidelines for Options property. **Note:** - **Text Parameter** and **Value Parameter** properties must be specified before using the Options property script. - **Fore Color** and **Back Color** properties cannot be set through scripts.

Note:
- Display value is readable.
- You can retrieve the display value via script as follows:

```
return control.findById("ID").displayValue;
```

Number

Use scripts to set and get values for the properties of the **Number** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`
Description	Yes Writable Readable	**To set the property** `control.findById("ID").description = "Value for Description";` **To get the value of the property**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		`control.findById("ID").description;`
Tooltip	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Read-Only	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").readOnly = true;`
		To get the value of the property
		`control.findById("ID").readOnly;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Watermark	No	Not Applicable
Width	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Auto-update Value	No	Not Applicable

Minimum Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").min = 100;`
		To get the value of the property
		`control.findById("ID").min;`

Maximum Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").max = 100;`
		To get the value of the property
		`control.findById("ID").max;`

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Number Format	No	Not Applicable
Decimal Places	No	Not Applicable
Decimal Separator	No	Not Applicable
Group Sizes	No	Not Applicable
Group Separator	No	Not Applicable
Show Group Separator	No	Not Applicable
Negative Pattern	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Read-Only	Yes	`return false;` For more information, see scripting guidelines for Read-Only property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.
Default Value	Yes	`return 100;` **Note:** The **Value** script takes precedence over the **Default Value** script.
Value	Yes	`return 100;`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Note:
- Display value is readable.

- You can retrieve the display value via script as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
return control.findById("ID").displayValue;
```

- Use parseInt/parseFloat while performing arithmetic operations on the Number control's value, as the Number control's value is of string type.

- Ensure the **Equality Operator (==)** is used instead of the **Identity Operator (===)** when type conversion is not required.

- For more information, see Best Practices for using scripts.

Password

Use scripts to set and get values for the properties of the **Password** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable

Appearance

Property	Scripting Supported	Script
Enable	Yes Writable Readable	**To set the property** `control.findById("ID").enable = true;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
		To get the value of the property `control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Read-Only	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").readOnly = true;`
		To get the value of the property
		`control.findById("ID").readOnly;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Watermark	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Text Pattern	No	Not Applicable
Minimum Length	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Maximum Length	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Security Mode	No	Not Applicable
Persist Password	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Read-Only	Yes	`return false;` For more information, see scripting guidelines for Read-Only property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		`ValidationOptions(true, "");` For more information, see scripting guidelines for Validate property.

Pop-Up

Use scripts to set and get values for the properties of the **Pop-Up** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable

Button Text	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").buttonText = "Value for Button Text";`
		To get the value of the property
		`control.findById("ID").buttonText;`

Persist Form Definition	No	Not Applicable
Form Details	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Dimension	No	Not Applicable
Retain Space	No	Not Applicable
Show as Link	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Open New Window	No	Not Applicable
Persist Value	No	Not Applicable
Form Button	No	Not Applicable

Scripts

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Example

If you want to populate the value of the **Name** (T1) control in the Pop-Up form with the value of the **Last Name** (T2) control in the parent form, then you can code the script for the **Value** property of the **Name** (T1) control as follows:

```
if(!SFU.isUndefined(window.parentViewModel))
{
return window.parentViewModel.topLevelForm.findById("T2").value;
}
```

If you want to populate the value of the **Project** (T3) control in the parent form with the value of the **Assigned Project** (T2) control in the Pop-Up form, then you can code the script for the **On Data Change** event of the **Details** (P1) control as follows:

```
if(!SFU.isUndefined(control.extendedInformation))
{
control.findById("T3").value = control.extendedInformation["Textinput2"];
}
```

If, **Assigned Project** (T2) control in the Pop-Up form is inside a container **Project Details** (B1), then you can code the script for the **On Data Change** event of the **Details** (P1) control as follows:

```
if(!SFU.isUndefined(control.extendedInformation))
{
control.findById("T3").value = control.extendedInformation["B1"]["T2"];
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Progress Bar

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Lookup Source	No	Not Applicable

Unit of Measurement	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").progressBarUom="%";`
		To get the value of the property
		`control.findById("ID").progressBarUom;`

Target Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").progressBarTargetValue=60;`
		To get the value of the property
		`control.findById("ID").progressBarTargetValue;`

Default Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").value=0;`
		To get the value of the property
		`control.findById("ID").value;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Retain Space	No	Not Applicable

Title	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").progressBarTitle = "Value for Title";`
		To get the value of the property
		`control.findById("ID").progressBarTitle;`

Decimal Places	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").progressBarDecimalPlaces=3;`
		To get the value of the property
		`control.findById("ID").progressBarDecimalPlaces;`

Scale	No	Not Applicable

Scale Minimum Value	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").progressBarScaleMinValue=0;`
		To get the value of the property
		`control.findById("ID").progressBarScaleMinValue;`

Scale Maximum Value	Yes Writable Readable	**To set the property** `control.findById("ID").progressBarScaleMaxValue=0;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

		To get the value of the property `control.findById("ID").progressBarScaleMaxValue;`
Control Background Color	No	Not Applicable
Background Color	No	Not Applicable
Bar Color	No	Not Applicable

Range	Yes	**To set the property**
		`control.findById(“ID”).options=[{optionvalue:20,foreColor:"red"},{optionvalue:60,foreColor:"green"}];`
		To get the value of optionvalue property of the first element
		`control.findById(“ID").options[0].optionvalue`
		To get the value of foreColor property of the first element
		`control.findById(“ID").options[0].foreColor`
		Note: The **Options** property exposed in scripting is used for the **Range** property.

Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Initialize	Yes	For more information, see scripting guidelines for **On Initialize** property.
Enable Refresh	No	Not Applicable
Refresh Rate	No	Not Applicable
ID	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;`
Value	Yes	`return 90;`

Radio Button

Use scripts to set and get values for the properties of the **Radio Button** control.

Basic

Property	**Scripting Supported**	**Example**

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable
Options	No	Not Applicable
Default Value	Yes	**To get the value of the property** `control.findById("ID").defaul`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
	Readable	`tValue;`

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Options Per Row	No	Not Applicable
Options Direction	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMand`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		`atory;`

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable

Text Parameter	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").optionText = "OText";`
		To get the value of the property
		`control.findById("ID").optionText;`

Value Parameter	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").optionValue = "OValue";`
		To get the value of the property
		`control.findById("ID").optionValue;`

Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.
Default Value	Yes	`return "ValueOne";` **Note:** -If the value is not one of the values set for the **Value** parameter in the **Options** property, then that value will not be a part of the options but will be a part of the XML instance. -The **Value** script takes precedence over the **Default Value** script.
Value	Yes	`return "ValueOne";`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Options	Yes	`return "[{\"OValue\":\"1\",\"OText\":\"ValueOne\"},{\"OValue\":\"2\",\"OText\":\"ValueTwo\"},{\"OValue\":\"3\",\"OText\":\"ValueThree\"}]";` Here **OText** is the **Text Parameter** and **OValue** is the **Value Parameter**. For more information, see scripting guidelines for Options property. **Note:** - **Text Parameter** and **Value Parameter** properties must be specified before using the Options property script. - **Fore Color** and **Back Color** properties cannot be set through scripts.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:
- Display value is readable.
- You can retrieve the display value via script as follows:

```
return control.findById("ID").displayValue;
```

Report Viewer

Use scripts to set and get values for the properties of the **Report Viewer** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Report Name	No	Not Applicable
Report Configuration	No	Not Applicable

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property. This property is currently not used.
ID	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.

Rich Text

Use scripts to set and get values for the properties of the **Rich Text** control.

Basic

Property	**Scripting Supported**	**Example**

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagNam`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		`e;`
XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Appearance

Property	Scripting Supported	Script
Show Toolbar	No	Not Applicable

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Read-Only	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").readOnly = true;`
		To get the value of the property
		`control.findById("ID").readOnly;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Advanced

Property	Scripting Supported	Example
ID	No	Not Applicable
Include in Summary	No	Not Applicable
Toolbar Options	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Read-Only	Yes	`return false;` For more information, see scripting guidelines for Read-Only property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.

Note: Entering or editing values in the **Rich Text** control is not supported in the **Mozilla Firefox** browser.

Role Lookup

Use scripts to set and get values for the properties of the **Role Lookup** control.

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
Label-Control Area	No	Not Applicable
Watermark	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.

Default Value	Yes	`return "639aa10e-dc02-446f-ac2e-d750550f7edb";`
		`Where "639aa10e-dc02-446f-ac2e-d750550f7edb" is a GUID.`
		Note: The **Value** script takes precedence over the **Default Value** script.

Value	Yes	`return "639aa10e-dc02-446f-ac2e-d750550f7edb";` `Where "639aa10e-dc02-446f-ac2e-d750550f7edb" is a GUID.`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Note:
- Display value is readable.
- You can retrieve the display value via script as follows:
`return control.findById("ID").displayValue;`

Signature

Use scripts to set and get values for the properties of the **Signature** control.

Basic

Property	Scripting Supported	Example
Name	Yes	**To set the property**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

	Writable
	Readable
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description= "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
Label-Control Area	No	Not Applicable
Width	Yes Readable	**To get the value of the property** `control.findById("ID").width;`
Height	Yes Readable	**To get the value of the property** `control.findById("ID").height;`
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example
Mandatory	No	Not Applicable

Advanced

Property	Scripting Supported	Example
ID	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		guidelines for Enable property
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.

Slider

Use scripts to set and get values for the properties of the **Slider** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Appearance

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
ID	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Default Value	Yes	`return 100;` **Note:** The **Value** script takes precedence over the **Default Value** script.

Text

Use scripts to set and get values for the properties of the **Text** control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description= "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script

Read-Only	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").readOnly = true;`
		To get the value of the property
		`control.findById("ID").readOnly;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Watermark	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Display Mode	No	Not Applicable
Rows	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Text Pattern	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").pattern = ["a-z"];`
		To get the value of the property
		`control.findById("ID").pattern;`

Minimum Length	Yes Writable Readable	**To set the property** `control.findById("ID").minLength = 100;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		To get the value of the property `control.findById("ID").minLength;`

Maximum Length	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").maxLength = 100;`
		To get the value of the property
		`control.findById("ID").maxLength;`

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Read-Only	Yes	`return false;` For more information, see scripting guidelines for Read-Only property.
Mandatory	Yes	`return true;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		For more information, see scripting guidelines for Mandatory property.
Default Value	Yes	`return "Default value for the control";` **Note:** The **Value** script takes precedence over the **Default Value** script.
Value	Yes	`return "Value for the control";`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Time

Use scripts to set and get values for the properties of the **Time** control.

Basic

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`
Description	Yes Writable Readable	**To set the property** `control.findById("ID").description = "Value for Description";` **To get the value of the property**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
		`control.findById("ID").description;`
Tooltip	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Read-Only	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").readOnly = true;`
		To get the value of the property
		`control.findById("ID").readOnly;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Width	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Minimum Value	No	Not Applicable
Maximum Value	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Time Format	Yes Readable	**To get the value of the property** `control.findById("ID").timeFormat;`
Time Interval	No	Not Applicable
Include in Summary	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Read-Only	Yes	`return false;` For more information, see scripting guidelines for Read-Only property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.
Default Value	Yes	`return "23:59:59";` **Note:** The **Value** script takes precedence over the **Default Value** script.
Value	Yes	`return "23:59:59";` **Note:** Time value is in 24-hour time format.

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Note:
- Display value is readable.
- You can retrieve the display value via script as follows:
`return control.findById("ID").displayValue;`

Tree

Use scripts to set and get values for the properties of the **Tree** control.

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`
Description	No	Not Applicable
Tooltip	No	Not Applicable
Data Source	No	Not Applicable
Node ID Column	No	Not Applicable
Node Parent Column	No	Not Applicable
Node Label Column	No	Not Applicable

Appearance

Property	Scripting Supported	Script

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable
Expand Nodes	No	Not Applicable

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
On Data Bound	Yes	For more information, see scripting guidelines for On Data Bound property.
On Selection Change	Yes	For more information, see scripting guidelines for On Selection Change property.
On Check Change	Yes	For more information, see scripting guidelines for On Check Change property.
ID	No	Not Applicable
Checkbox Behavior	No	Not Applicable
Node Checked Column	No	Not Applicable
Inherit Checked State	No	Not Applicable

Scripts

Property	Scripting Supported	Example
Name	No	Not Applicable
Description	No	Not Applicable
Visible	No	Not Applicable

User Lookup

Use scripts to set and get values for the properties of the **User Lookup** control.

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Tooltip	No	Not Applicable
Default Value	Yes Readable	**To get the value of the property** `control.findById("ID").defaultValue;`
Show Default User	No	Not Applicable

Appearance

Property	Scripting Supported	Script

Enable	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").enable = true;`
		To get the value of the property
		`control.findById("ID").enable;`

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Script
Label-Control Area	No	Not Applicable
Watermark	No	Not Applicable
Width	No	Not Applicable
Height	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Validation

Property	Scripting Supported	Example

Mandatory	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").isMandatory = true;`
		To get the value of the property
		`control.findById("ID").isMandatory;`

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable

Extended Information	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").requiresExtendedInformation = true;`
		To get the value of the property
		`control.findById("ID").requiresExtendedInformation;`
		For more information, see scripting guidelines for Extended Information.

Include in Summary	No	Not Applicable

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Enable	Yes	`return true;` For more information, see scripting guidelines for Enable property.
Visible	Yes	`return true;` For more information, see scripting guidelines for Visible property.
Mandatory	Yes	`return true;` For more information, see scripting guidelines for Mandatory property.

Default Value	Yes	`return "639aa10e-dc02-446f-ac2e-d750550f7edb";`
		`Where "639aa10e-dc02-446f-ac2e-d750550f7edb" is a GUID`
		Note: The **Value** script takes precedence over the **Default Value** script.

Value	Yes	`return "639aa10e-dc02-446f-ac2e-d750550f7edb";` `Where "639aa10e-dc02-446f-ac2e-d750550f7edb" is a GUID`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Note:
- Display value is readable.
- You can retrieve the display value via script as follows:
`return control.findById("ID").displayValue;`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Widget

Use scripts to set and get values for the properties of the **Data Grid** control.

Use **Widget** control to create reusable graphical components. You can maintain a library of widgets for the other users to use.

Properties

The properties for the control are grouped under the following tabs:

Basic

Property	Scripting	Example

Name	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").tagName = "Value for Name";`
		To get the value of the property
		`control.findById("ID").tagName;`

XML Node	Yes Readable	**To get the value of the property** `control.findById("ID").xmlNodeBoundTo;`

Description	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").description = "Value for Description";`
		To get the value of the property
		`control.findById("ID").description;`

Widget Type	No	Not Applicable
Widget	Yes Readable	**To get the value of the property** `control.findByXmlNode("Widget1").widgetName;`
Configuration	Yes Readable Writable	**To get the value of the property** `control.findByXmlNode("Widget1").confName;`
Form Details	No	Not Applicable

Appearance

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Scripting Supported	Example
Style	No	Not Applicable
In Designer	No	Not Applicable

Visible	Yes
	Writable
	Readable
		To set the property
		`control.findById("ID").visible = true;`
		To get the value of the property
		`control.findById("ID").visible;`

Label Position	No	Not Applicable
Label-Control Area	No	Not Applicable
Retain Space	No	Not Applicable
Custom Style Sheet Identifier	No	Not Applicable

Advanced

Property	Scripting Supported	Example
On Data Change	Yes	For more information, see scripting guidelines for On Data Change property.
ID	No	Not Applicable
Add Values as Url	No	Not Applicable
Widget Properties	Yes	**To get the value of the property** `control.findByXmlNode("Widget1").widgetProperties;` The properties are listed based on the Widget selected in the control. For more information, see the respective Widget documentation. • Grid Widget • Hierarchy Selector Widget **Note**: Form Parameters are considered as **Widget Properties** for the Widget control irrespective of the data

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

		type. The **Widget Properties** are always represented as String data type.

Scripts

Property	Scripting Supported	Example
Name	Yes	`return "Value for Name";`
Description	Yes	`return "Value for Description";`
Visible	Yes Writable Readable	`return true;`
Default Value	Yes	`return "Default value for the control";` **Note:** The **Value** script takes precedence over the **Default Value** script.
Value	Yes	`return "Value for the control";`

Validate	Yes	`return new ValidationOptions(false, "Error Message");`
		OR
		`return new ValidationOptions(true, "");`
		For more information, see scripting guidelines for Validate property.

Widget Scripts	Yes	The dynamic scripts are listed based on the Widget selected in the control. For more information, see the respective Widget documentation on the dynamic scripts. • Grid Widget • Hierarchy Selector Widget

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

		Note: This property is not applicable for Widget Type - Form.

Forms Parameters

Forms parameters are form-level variables used to store data in:

- Form Link
- Pop-Up Control
- Invoke Form Activity

The value of a forms parameter is used when the form is rendered.

Types of Forms Parameters

- String
- Array
- Number
- Date
- Object

Rules for Forms Parameter Names

Forms parameter names are case-sensitive and can start with a number. However, forms parameter names:

- Must be unique.
- Cannot contain non-English characters.
- Cannot contain spaces.
- Cannot contain special characters such as ' ! " # % & ( ) * , . / : ; ? @ [\ ] ^ _ ` { | } ~ + < = >

Add Forms Parameters

Add a forms parameter to use with a workflow at run-time.

To add a forms parameter to a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click **Design**). The **Forms Designer** page appears.
3. On the action bar, click **Properties**. The **Properties** dialog box appears.
4. Click the **Advanced** tab, and then click the **Add parameter** button under the **Forms Parameters**. A new row

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

appears.

5. Enter the parameter name in the **Parameter Name** box, and select the **Parameter Type**.
6. Click **Apply**. The forms parameter is saved and the **Properties** dialog box is closed.
7. Click the **Close** button. The form is saved and the **Forms Designer** page is closed.

Delete Forms Parameters

Delete a forms parameter when the forms parameter is no longer required.

To delete a forms parameters from a form

1. In the **Enterprise Console**, click the **Menu** button, and then click **Forms**. The **Forms** page appears.
2. Right-click on a form, and then click **Design Form** (or click on a form, and then click **Design**). The **Forms Designer** page appears.
3. On the action bar, click **Properties**. The **Properties** dialog box appears.
4. Click the **Advanced** tab, and then click the **Delete Parameter** button at the end of a forms parameter under the **Forms Parameters**.
5. Click **Apply**. The forms parameter is deleted and the **Properties** dialog box is closed.
6. Click the **Close** button. The form is saved and the **Forms Designer** page is closed.

Set Forms Parameters

Set forms parameters to manage controls on a form at run-time.

Setting Forms Parameters through Form Links

- When you create a **Form Link**, you can select and type values for the parameters. For more information,see Creating Forms Links.

Setting Forms Parameters through Pop-Up Control

- When you create a **Pop-Up** control, you can select parameters and map it to parent form controls. For more information, see *To add Form Details for the Control* under *Form Details* property in Pop-Up.

Setting Forms Parameters through External Systems

The forms parameter must be specified in JSON (JavaScript Object Notation) string format with attribute–value pairs. Multiple forms parameters can be specified by separating them with commas.

Syntax

```
// extForParam is the global variable used for setting Forms Parameters.
window.extFormParam = { "<formsParameterName>": { "id": <idValue>, "mappedParentControl":
"<mappedParentControlValue>", "name": "<formsParameterName>", "type":
"<formsParameterType>", "value": "<formsParameterValue>" } };
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Value	Description
<formsParameterName>	Name of the forms parameter.
<idValue>	Value of the forms parameter ID. This value is optional. Use null if no value is used.
<mappedParentControlValue>	Value of the forms parameter mapped to the parent control. This value is optional. Use "" if no value is used.
<formsParameterType>	Type of the forms parameter. For more information about, see Types of Forms Parameters.
<formsParameterValue>	Value of the Forms Parameter.

Example

Setting Forms Parameters through an HTML page

If you can want to populate the **First Name** and the **Last Name** text controls on the **Employee** form with values from the **FName** and the **LName** forms parameters, then you can code the form and the HTML page as follows:

Script the **On Form Load** property of the **Employee** form to populate the text controls with the values from the forms parameters.

```
// T1and T2 are the Text controls for the First Name and the Last Name respectively.
control.findById("T1").value = control.topLevelForm.formParameters["FName"].value;
control.findById("T2").value = control.topLevelForm.formParameters["LName"].value;
```

Code the HTML page to set the values for the forms parameters.

```
// John and Doe are values set to FName and LName respectively.
// extForParam is the variable used for setting Forms Parameters.
// This code must appear before the loadViewAndViewModelFromServer() function.
window.extFormParam = { "FName": { "name": "FName", "type": "STRING", "value": "John" },
"LName": { "name": "LName", "type": "STRING", "value": "Doe" } };
```

Forms Parameters for Invoke Form Activity

Forms parameters are form-level variables used in workflows at run-time to store data from a workflow and display the data in a form. The value of a forms parameter is used when the form is rendered.

Forms parameters are specifically used in **Invoke Form** scenarios where a form is populated as a **Work Item** and the parameters work within the boundaries of the **Invoke Form** activity.

Default Forms Parameters

The default parameters that can be obtained from the forms parameters in **Invoke Form** scenario using JavaScript are:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Forms Parameters	Script
Execution ID	control.topLevelForm.formParameters["variable.executionId"].value
Execution Details ID	control.topLevelForm.formParameters["variable.executionDetailsId"].value
Submitted User ID	control.toplevelForm.formParameters["submittedById"].value
Work Item ID	control.topLevelForm.formParameters["variable.workItemId"].value
Workflow Name	control.topLevelForm.formParameters["variable.workflowName"].value

Accessing Forms Parameters through scripts

```
// Script for the On Data Change event.
// GetWFName is a forms parameter set from an external system (Workflow).
// Gets the workflow name on data change of the control.
var getwfname = control.topLevelForm.formParameters["GetWFName"];
if(getwfname !== undefined && getwfname !== null){
 return (getwfname.value);
}
// Script for the Visible property.
// GetWFName is forms parameter set from an external system (Workflow).
// Hides the control based on the forms parameter value.
var getwfname = control.topLevelForm.formParameters["GetWFName"];
if(getwfname !== undefined && getwfname !== null){
 return (getwfname.value !== null?false:true);
}
```

Accessing Default Forms Parameters through scripts

```
// Accesses the default forms parameter - Workflow name.
var getwfname = control.topLevelForm.formParameters["variable.workflowName"];
if(getwfname !== undefined && getwfname !== null){
 if(getwfname.value !== null){
 return getwfname.value;
 }
}
```

Note: When a forms parameter of type date is set through a workflow variable of type date in the **Invoke Form** activity, the forms parameter will contain a "T" separator between date and time.

Best Practices

We recommend the following best practices when working with scripts in your forms.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Ensure you do not use keywords for variables.
  - Do not use **control** and **currentValue** for creating variables.
- Ensure you use a string for the **Value** property of the **Boolean, Date, Time,** and **DateTime** controls.
  - **Enable**, **Visible**, **Read-Only**, and **Mandatory** properties hold their value as true or false. However, the **Value** property for the **Boolean** control has a string value, which can be either "**True**" or "**False**".
  - **Value** property for the **Date**, **Time**, and **DateTime** should be given in the specific string format. Refer the scripting section on these controls for more details.
- Ensure you refer to the same control.
  - Refer to the same control when scripting for the **Enable**, **Visible**, **Read-Only**, **Value**, or **Validate** property of a control.
  - Use the **currentValue** method in the script to refer to the value of the property of a control that is scripted.
  - Do not validate the **Value** property of a control that is set as invisible.
- Ensure the script returns a value.
  - The script must return a value even when the entire script is commented except when scripting for the **On Data Change**, **On Form Load**, **On Click**, and so on events.
  - Most of the scripts present in the Scripts Tab of which the common ones are **Name**, **Description**, **Enable**, **Visible**, **Read-Only**, **Mandatory**, **Validate**, and **Default Value**, require a return value.
  - Refer the scripts documentation for the control to know which ones require a return value and accordingly have the scripts written.
- Ensure the script returns the required value.
  - The script should return a Boolean value for the **Enable**, **Visible**, **Read-Only**, or **Mandatory** property of a control.
  - The default value of the property for a control is returned when the script does not have a return statement.
  - The previous value for the **Enable**, **Visible**, **Read-Only**, or **Mandatory** property of a control is retained when the script does not have a return statement or the return value is invalid.
- Ensure invalid values are not set through the scripts.
  - The invalid values are cleared on submission of a form, and other rules such as validation, minimum value, and maximum value are not triggered when an invalid value is set through the scripts for the **Value** property of a **Number**, **Date**, or **Time** control.
- Ensure commented code is handled in the script.
  - The script should not contain only commented code as in such cases the code is executed.
  - The script should not contain unnecessary comments, and the code that is no longer required should be removed.
- Scripts like **Name**, **Description**, **Enable**, **Visible**, etc. in the **Scripts** tab, except for the **On Form Load** script, create dependency injections based on the Script written. Hence, we recommend to write code that would perform the required checks and return a value. Also, these scripts should not set values to other controls or properties..
- Ensure the rules are triggered.
  - Validation rules for a **Number**, **Date**, or **Time** control are not triggered at form load when scripting for the **Value** property of the control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Ensure the **Equality Operator (==)** is used instead of the **Identity Operator (===)** when type conversion is not required.
  - The **Equality Operator (==)** is used for an abstract comparison and converts the operands to the same type before making the comparison, whereas the **Identity Operator (===)** is used for a strict comparison and is true only if the operands are of the same type.

    Example

    If the **Number (N1)** control is set to 123.00, then the following comparison is false as the **Number (N1)** control returns 123.

    ```
    if(control.findById("N1").value === "123.00")
    {
       alert("True");
    }
    else
    {
       alert("False");
    }
    ```

    The comparison will be true when the operator or operand is changed as follows:

    ```
    if(control.findById("N1").value == "123")
    ```

    OR

    ```
    if(control.findById("N1").value === 123.00)
    ```

    Note: If the **Number** (N1) control is set to 123.45, then the **Number** (N1) control returns 123.45, and you can continue to compare as follows:

    ```
    if(control.findById("N1").value === "123.45")
    ```

    as the decimal value other than 0 is not discarded.

- Ensure you create a variable outside the AJAX call and assign that variable inside the success function. Then return the variable after the AJAX call, when you code a synchronous AJAX call which returns a result, as shown in below example.

  ```
  var returnData;
  $.ajax({
     type: …,
     async: false,
     url: …,
     contentType: …,
     dataType: …,
     data: …,
     success: function(result)
     {
     returnData = result;//true or false
     },
     error: …
  });
  return returnData;
  ```

- Ensure you check the syntax of the script before you save the script.
- In case of scripts that return value, when you use control.findByXmlNode, control.findById, and/or control.findByName APIs to refer to the properties like **Enable**, **Visible**, **Value**, etc., the dependency would be created on these properties. So if any value change occurs to the dependent properties, the script would re-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

trigger.

- Ensure you write the scripts for functions as shown below:

  Assign function to Variable:

  ```
  var count = function(value) {
     return value.length;
  };
  ```

  IIFE functions:

  ```
  function () {
     //write your js code here
  });
  ```

  Global functions:

  ```
  window.calculateLength = function(value) {
     return value.length;
  };
  ```

- Avoid writing huge scripts in the scripts editor in Forms. The script editor has an auto save functionality, which would save the scripts without having the script been completely verified on the syntax. This is due to the fact that the scripts are not lost in case of a system crash. But there is a negative effect for this functionality as well. After the system crash (let's assume), when you open the form and preview, it might not show the preview and give exceptions. So you would need to update the scripts where you had left off before the system crash, and make appropriate corrections. Hence, it is better to avoid large scripts written in the script editor. Instead, use custom js files to write down the huge scripts and have a call to those custom script functions from the Form scripts.

- The **On Data Change** script for a control would not execute when value is set in the **On Form Load** script. This is done to prevent inadvertent triggering of **On Data Change** event when instance xml is set while invoking the form.

  Example:

  A Form is having only 1 Textinput control. The **On Form Load** script is defined as below:

  ```
  console.log("**********************************************************Textinput1 value
  set from On Form Load");
  control.findByXmlNode("Textinput1").value = "Value set from On Form Load";
  ```

  And have the **On Data Change** script for the Textinput control as below:

  ```
  console.log("**********************************************************Textinput1 - On
  Data Change script - value: " + currentValue);
  ```

  When you preview the form, it would not execute the **On Data Change** script of the Textinput control, as you have set the value of the TextInput control in the **On Form Load** script. Below is the console log result:

  ```
  **********************************************************Textinput1 value set from On
  Form Load
  ```

- Property based scripts takes precedence – **Name**, **Description**, **Enable**, **Visible**, **Mandatory**, **Value**, **Default Value** type of scripts

  - Example with **On Form Load** script and **Textinput Value** script:

    A Form is having only 1 Textinput control. The **On Form Load** script is defined as below:

    ```
    console.log("**********************************************************Textinput
    1 value set from On Form Load");
    control.findByXmlNode("Textinput1").value = "Value set from On Form Load";
    ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

And have the **Value** script for the Textinput control as below:

```
console.log("**********************************************************Textinput
1 value set via Value script");
return "Value set from Textinput control's Value script";
```

When you preview the form, it would execute the **On Form Load** event of the Textinput control, but as you have the **Value** script defined for the TextInput control, the **Value** script would take precedence. Below is the console log result:

```
**********************************************************Textinput1 value set
from On Form Load
**********************************************************Textinput1 value set
via Value script
**********************************************************Textinput1 value set
via Value script
**********************************************************Textinput1 value set
via Value script
```

Note: The last 2 logs related to Value script execution happens due to internal handling of scenario around value script for Validation and On Form Load completion events.

- Example with **Button On Click** script and **Textinput Enable** script:

  A Form is having only 1 Textinput control. In the Textinput control properties, set **Enable** property as **No** and define the Enable script as below:

  ```
  console.log("**********************************************************Textinput
  Enable property set to True via Textinput Enable script");
  return true;
  ```

  Let us add a button control and give the **On Click** script as below:

  ```
  console.log("**********************************************************Textinput
  Enable property set to False via button click");
  control.findByXmlNode("Textinput1").enable = false;
  ```

  When you preview the form, it shows the Textinput control in enabled state, as the **Enable** script takes precedence.

  When you click on the button, you would see that the Textinput control is still in the enabled state, as the **Enable** script takes precedence. Below is the console log result:

  ```
  **********************************************************Textinput Enable
  property set to True via Textinput Enable script
  **********************************************************Textinput Enable
  property set to False via button click
  **********************************************************Textinput Enable
  property set to True via Textinput Enable script
  ```

- The control's value set last via a script would have it's On Data Change script executed last.

  Example:

  A Form having controls - 2 Dateinput, 1 Textinput, and 1 Button.

  The Dateinput1 control has the **On Data Change** script as below:

  ```
  console.log("**********************************************************Dateinput1 On
  Data Change script - Update Textinput1 with value");
  control.findByXmlNode("Textinput1").value = currentValue;
  ```

  The Dateinput2 control has the **On Data Change** script as below:

  ```
  console.log("**********************************************************Dateinput2 On
  Data Change script - Update Textinput1 with value");
  ```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.findByXmlNode("Textinput1").value = currentValue;
```

The button control has the **On Click** script as below:

```
console.log("*****************************************************Set Dateinput1
value as 03/20/2022");
control.findByXmlNode("Dateinput1").value = "03/20/2022 00:00:00";
console.log("*****************************************************Set Dateinput2
value as 03/30/2022");
control.findByXmlNode("Dateinput2").value = "03/30/2022 00:00:00";
```

When you preview the form, it would show the Textinput control's value as the Dateinput2 control's value, as the button **On Click** script sets the Dateinput2 control's value at the end. Below is the console log result:

```
*****************************************************Set Dateinput1 value as 03/
20/2022
*****************************************************Dateinput1 On Data Change
script - Update Textinput1 with value
*****************************************************Set Dateinput2 value as 03/
30/2022
*****************************************************Dateinput2 On Data Change
script - Update Textinput1 with value
```

If you change the order in which the Dateinput control values are set via the button **On Click** script, you will have to change the button **On Click** script as below:

```
console.log("*****************************************************Set Dateinput2
value as 03/30/2022");
control.findByXmlNode("Dateinput2").value = "03/30/2022 00:00:00";
console.log("*****************************************************Set Dateinput1
value as 03/20/2022");
control.findByXmlNode("Dateinput1").value = "03/20/2022 00:00:00";
```

When you preview the form, it would show the Textinput control's value as the Dateinput1 control's value, as the button **On Click** script sets the Dateinput1 control's value at the end. Below is the console log result:

```
*****************************************************Set Dateinput2 value as 03/
30/2022
*****************************************************Dateinput2 On Data Change
script - Update Textinput1 with value
*****************************************************Set Dateinput1 value as 03/
20/2022
*****************************************************Dateinput1 On Data Change
script - Update Textinput1 with value
```

- Referring to properties of the same control or other control in scripts like **Name**, **Description**, **Enable**, **Visible**, **Read-Only**, **Mandatory**, **Default Value**, **Value**, and **Validate**, would create the dependencies on those properties. If any change happens to those dependent control properties, the scripts would get executed.

  Example:

  Consider a Textinput control's Scripts Tab containing scripts for **Name, Description, Enable, Visible, Read-Only, Mandatory, Default Value, Value**, and **Validate**. These scripts are also based on the control's respective property except in the case of **Validate** script, like **Enable** script of Textinput control would relate to the **Enable** property of the Textinput control.

  So, when you define these kind of scripts, which require return values, then those return values are targeted to the respective property's value. When writing scripts, you would refer to other controls and their properties in the scripts, which create dependencies for these kinds of scripts. For example, consider a BooleanInput control and a Textinput control in the Form. The Textinput control's **Enable** script is given as below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
if (control.findByXmlNode("Booleaninput1").value == "True")
{
   return true;
}
else
{
   return false;
}
```

When you preview the form, the Textinput control is disabled, as the BooleanInput control is set to **No**. If you change the BooleanInput control to **Yes**, then the Textinput control is enabled. This is how dependency gets created for BooleanInput control's **Value** property when used in the Textinput control's **Enable** script.

Note: You should not set any other control properties or value in these scripts, only reading the values can be done. For example, you cannot set the **Read-Only** property's value for the same control or other control, in the control's enable script. This is to avoid unwanted behavior and/or recursive loops in the form.

- When all the property based scripts like **Name, Description, Enable, Visible,** etc. are given for a control, the properties are defined for the control initially. Once the **On Form Load** script completes execution, these scripts would get executed.

  Example:

  Lets define a form with a Textinput control. Have the **On Form Load** script as below:

  ```
  console.log("**********************************************************On Form Load
  script executed");
  ```

  Have the Textinput control's **Name** script as below:

  ```
  console.log("**********************************************************Textinput1 Name
  script");
     return currentValue;
  ```

  Similarly, have the Textinput control's **Description** script as below:

  ```
  console.log("**********************************************************Textinput1
  Description script");
     return currentValue;
  ```

  Similarly, define the scripts for **Enable, Visible, Read-Only, Mandatory, Default Value,** and **Value** in the **Scripts** tab for the Textinput control.

  When you preview the form, the console log result is as below:

  ```
  ******************************************************On Form Load script executed
  ******************************************************Textinput1 Mandatory script
  ******************************************************Textinput1 Name script
  ******************************************************Textinput1 Description script
  ******************************************************Textinput1 Visible script
  ******************************************************Textinput1 Read-Only script
  ******************************************************Textinput1 Enable script
  ******************************************************Textinput1 Value script
  ```

  The order of these scripts execution is not important, they might get triggered again based on the dependencies they have. The important point is that we should never set any other control's value in these scripts. We can only refer or read other control's property values or the same control's property values in these scripts, which would create the dependencies and accordingly the execution of the scripts would happen based on the dependencies, as and when these dependent values are changed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

APIs

Application programming interface (API) is a set of functions to accomplish a specific task. APIs provide speed, reliability, and extensibility.

Forms APIs

Application programming interface (API) is a set of functions to accomplish a specific task. APIs provide speed, reliability, and extensibility.

Forms APIs make use of the following namespaces, classes, and objects.

Namespace	Class	Object
skelta skelta.forms	utilities	control

Use forms APIs to customize your business applications.

SFU.closeForm()

Use `closeForm()` method to close the window in which the form is rendered.

Syntax

```
SFU.closeForm();
```

Parameters

None.

Return Value

None.

Example

```
// Close the window in which the form is rendered
SFU.closeForm();
```

SFU.decodeXmlValue()

Use `decodeXmlValue()` method to decode all characters in the encoded XML value.

The following encoded characters are decoded as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Encoded Value	Decoded Value
&	&
'	'
"	"
<	<
>	>

Syntax

```
sRetVal = SFU.decodeXmlValue(string);
```

Parameters

Parameter	Description
string	String to be decoded.

Return Value

This method returns a string value.

Example

```
// Decode encoded XML value.
SFU.decodeXmlValue("&amp;quot;Developer&amp;apos;s&amp;quot; Guide");
//&amp;quot;Developer&amp;apos;s&amp;quot; Guide is decoded as "Developer's" Guide.
```

SFU.encodeXmlValue()

Use `encodeXmlValue()` method to encode all characters in the XML value.

The following characters in the XML value are encoded as follows:

XML Value	Encoded Value
&	&
'	'
"	"
<	<

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

XML Value	Encoded Value
>	>

Syntax

```
sRetVal = SFU.encodeXmlValue(string);
```

Parameters

Parameter	Description
string	String to be encoded.

Return Value

This method returns a string value.

Example

```
// Encode XML value.
SFU.encodeXmlValue("\"Developer's\" Guide");
//"Developer's" Guide is encoded as &amp;quot;Developer&amp;apos;s&amp;quot; Guide.
```

SFU.getClientCredentialToken()

Use `getClientCredentialToken()` method to get the client credential token with custom scope.

Syntax

```
var accessToken = SFU.getClientCredentialToken();
```

Return Value

This method returns token value of string datatype.

Example

```
//Get the client credential token with custom scope.
var accessToken = SFU.getClientCredentialToken();
```

Note:
For Azure Active Directory provider, this method returns the token with the scope - `api://<clientid>/worktasks/`
For AVEVA Connect provider, this method returns the token with the scope - `ac_worktasks`
For AVEVA Identity Manager provider, this method returns the token with the scope - `system`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SFU.getDateTimeInStringFormat()

Use `getDateTimeInStringFormat()` method to get the date time from the string format.

Syntax

```
sRetVal = SFU.getDateTimeInStringFormat(dateTime,
skelta.forms.constants.dateFormats.dateTimeFormatForCoreValue);
```

Parameters

Parameter	Description
dateTime	Date and Time object. **Note**: The time zone from the JavaScript Date object is not considered.
skelta.forms.constants.dateFormats.dateTimeFormatForCoreValue	Date and Time Formate Constant.

Return Value

This method returns a date time string.

Example

```
// Show date and time in the Date Time control from the value of the Forms Parameter.
var myDateTime = control.topLevelForm.formParameters["DateTimeParam"].value;
var OutDateTime = SFU.getDateTimeInStringFormat(myDateTime,
skelta.forms.constants.dateFormats.dateTimeFormatForCoreValue);
control.findById("D1").value = OutDateTime;
```

SFU.getDateTimeInServerUTCFormat()

Use `getDateTimeInServerUTCFormat()` method to get the date and time in Server or Local culture.

Syntax

```
sRetVal = SFU.getDateTimeInServerUTCFormat(dateTime, forUIValue);
```

Parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
dateTime	Date and Time object. **Note**: The time zone from the JavaScript Date object is not considered.
forUIValue	A Boolean flag which determines whether the dateTime parameter value has to be converted to a local/display value based on the server UTC offset. This parameter is optional. The default value is **false**.

Return Value

This method returns a date and time object.

Example

```
// Get the date and time in Server culture.
var dateTime = new Date();
var dateTimeUtc = SFU.getDateTimeInServerUTCFormat(dateTime);
// Get the date and time in Local culture.
var dateTimeUtc = new Date(); // UTC Date value
var dateTime = SFU.getDateTimeInServerUTCFormat(dateTimeUtc, true);
```

SFU.getDateTimeInUTCFormat()

Use `getDateTimeInUTCFormat()` method to get the date and time in UTC.

Syntax

```
sRetVal = SFU.getDateTimeInUTCFormat(dateTime);
```

Parameters

Parameter	Description
dateTime	Date and Time object. **Note**: The time zone from the JavaScript Date object is not considered.

Return Value

This method returns a date and time object.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example

```
// Get the date and time in UTC.
var dateTime = new Date();
var dateTimeUtc = SFU.getDateTimeInUTCFormat(dateTime);
```

SFU.getFormUrl()

Use `getFormUrl()` method to get the URL, which can be used to navigate to a different form.

Note: This API is not supported in AVEVA Work Tasks Pro App.

Syntax

```
SFU.getFormUrl(formName,formVersion);
```

Parameters

Parameter	Description
formName	Name of the form that you want to navigate to. The value must be a string.
formVersion	Version of the form. The value should be string. If the value is set to blank, then the latest published version of the form is considered.

Note:
- Ensure that the form with the specified title and the version exists and is in the "Published" state.

- Folders are not supported for this API.

Return Value

This method returns the URL.

Example

```
// Get the form that has to be embedded in the current form.
var formsURL=SFU.getFormUrl("ApplicationForm1","1");
control.findByXmlNode("Embedpage1").url = formsURL;
```

SFU.getGridValue()

Use `getGridValue()` method to get the persisted column values when the check box is selected for the record.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Syntax

```
aRetVal = SFU.getGridValue(gridControl);
```

Parameters

Parameter	Description
gridControl	Grid control.

Return Value

This method returns an array of objects.

Example

```
// Get the persisted column values.
var gridControl = control.findById("G1");
var rowData = SFU.getGridValue(gridControl);
control.findById("T3").value = rowData[0].<lookupname>.AddressID;
control.findById("T4").value = rowData[0].<lookupname>.City;
control.findById("T5").value = rowData[0].<lookupname>.PostalCode;
```

Note: The DateTime column types would contain DateTime values (not String values), when the **Use Culture Specific Values** property is set to **No**.

SFU.getImagePath()

Use `getImagePath()` method to include image from the image path for the **Image** control.

Syntax

```
SFU.getImagePath(imagePath);
```

Parameters

Parameter	Description
imagePath	Path to the image for the Image control.

Return Value

None

Example

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// Include MyLogo.png image for the Image control through the Value script.
return SFU.getImagePath("<BPMUITemplates>/Images/MyLogo.png");
```

SFU.getJSONObjectFromXml()

Use `getJSONObjectFromXml()` method to convert an XML string to a JSON object.

Syntax

```
sRetVal = SFU.getJSONObjectFromXml(xmlString);
```

Parameters

Parameter	Description
xmlString	XML string to convert to JSON object.

Return Value

This method returns a JSON object.

Example

```
// Define an XML string.
var xmlString = "<MyRoot><FirstName>John</FirstName><LastName>Smith</LastName></MyRoot>";
// Convert the XML string to a JSON object.
var jsonObj = SFU.getJSONObjectFromXml(xmlString);
// The JSON object will be returned as:
// {
// "MyRoot" : {
//  "FirstName":"John",
//  "LastName":"Smith",
// }
```

SFU.getLookupSchemaAndData()

Use `getLookupSchemaAndData()` method to get Data Lookup schema and data from Lookup name.

Syntax

```
SFU.getLookupSchemaAndData(lookupName, lookupParameters, getAllColumnData);
```

Parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
lookupName	Name of the Data Lookup.
lookupParameters	Collection of parameters for parameterized Data Lookup This parameter is optional. Pass this only when lookup is Parameterised
getAllColumnData	Boolean parameter. Default value is false. Pass this as true when all the columns from the Data Lookup query are required. This parameter is optional. Pass this only when all the columns from the Lookup query are required.

Return Value

This method returns a JSON object containing Display column, Value column, Data.

Example

For examples, see Data Binding Drop-Down Controls Using a Lookup Name (Parameterized) and Data Binding Drop-Down Controls Using Lookup Name (Non-parameterized)

SFU.getServerUtcOffset()

Use `getServerUtcOffset()` method to get the difference in minutes from the UTC and the server time.

Syntax

```
nRetVal = SFU.getServerUtcOffset();
```

Parameters

None.

Return Value

This method returns a numeric value.

Example

```
// Get the difference in minutes from the UTC and the server time.
var offsetMin = SFU.getServerUtcOffset();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SFU.getStorageItem()

Use `getStorageItem()` method to get value for the specified key in the web storage.

Syntax

```
aRetVal = SFU.getStorageItem(key);
```

Parameters

Parameter	Description
key	Key in the web storage for which value is required.

Return Value

This method returns an array.

Example

```
// Get value for the specified key in the web storage.
var workItems = SFU.getStorageItem("WorkItemData");
```

SFU.getUserAccessToken()

Use `getUserAccessToken()` method to get the logged in user access token.

Syntax

```
var UserAccessToken = SFU.getUserAccessToken();
```

Return Value

This method returns user access token.

Example

```
//Get the user access token.
var UserAccessToken = SFU.getUserAccessToken();
```

SFU.getUserLookupExtendedInformation()

Use `getUserLookupExtendedInformation()` method to get the extended information for the user using the virtual actor ID.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Syntax

```
sRetVal = SFU.getUserLookupExtendedInformation(virtualActorId);
```

Parameters

Parameter	Description
virtualActorId	Virtual Actor ID.

Return Value

This method returns a string value.

Example

```
// Get the Email (which is the extended information) for the user using the virtual actor
ID.
var extendedInformation = SFU.getUserLookupExtendedInformation(currentValue,false);
control.findById("T6").value = extendedInformation["UserEmail"];
//Here, currentValue refers to virtualActorId
```

SFU.getUserMemberOf()

Use `SFU.getUserMemberOf()` method to get the list of groups name to which the logged-in user is the direct member. This API works only with the Azure Active Directory repository.

Syntax

```
SFU.getUserMemberOf);
```

Return Value

List of groups name to which the logged-in user is the direct member. If the user is part of multiple groups, comma separated string of the groups is returned.

Example 1

```
// Gets the Groups for the logged-in user being the direct member
var groups = SFU.getUserMemberOf();
// groups variable would get a value like- "Everyone,skeltalist::Admin"
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SFU.getUserRoles()

Use `SFU.getUserRoles()` method to get the roles of a user.

Syntax

```
SFU.getUserRoles(userIdWithProvider)
```

Parameters

Parameter	Description
userIdWithProvider	User Id of the user for whom the roles are fetched. This parameter is optional. If the parameter is not passed, it fetches the roles for the current logged-in user.

Return Value

Roles of the User. If the user has multiple roles, comma separated string of the roles is returned.

Example 1

```
// Gets the Roles for the logged-in user
var roles = SFU.getUserRoles();
// roles variable would get a value like- "Everyone,skeltalist::Admin"
```

Example 2

```
// Gets the Roles for the User with Id "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"
var roles = SFU.getUserRoles("skeltalist::XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX");
// roles variable would get a value like- "Everyone,skeltalist::Contributor"
```

The `userIdWithProvider` value varies based on the authentication provider configured. This value should have the Entity instance mapped for the repository to be prefixed along with real id of the user.

- If the repository is mapped to **Repository User Provider**, the `userIdWithProvider` will be skeltalist::[GUID]. Here the GUID is of the user john, taken from SKEUsers table present in the Repository DB.

  For example: skeltalist:: C99B4110-C8CC-410B-BF70-93F3A7E34EDF
- If the repository is mapped to the **Active Directory** user provider, the `userIdWithProvider` will be activedirectory::[samAccountName]. Here, samAccountName is an active directory property value.

  For example: activedirectory::john

Note: The `SFU.getUserRoles()` method does not work in AVEVA Work Tasks Pro app.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SFU.getUserSecurityGroup()

Use `SFU.getUserSecurityGroup()` method to get the effective security group of a user.

Syntax

```
SFU.getUserSecurityGroup(userIdWithProvider)
```

Parameters

Parameter	Description
userIdWithProvider	User Id of the user for whom the security group is fetched. This parameter is optional. If the parameter is not passed, it fetches the roles for the current logged-in user.

Return Value

Effective Security Group of the User.

Example 1

```
// Gets the Effective Security Group for the logged-in user
var securityGroup = SFU.getUserSecurityGroup();
// securityGroup variable would get a value like- "Administrator"
```

Example 2

```
// Gets the Effective Security Group for the User with Id "XXXXXXXX-XXXX-XXXX-XXXX-
XXXXXXXXXXXX".
var securityGroup = SFU. getUserSecurityGroup("skeltalist::XXXXXXXX-XXXX-XXXX-XXXX-
XXXXXXXXXXXX");
// securityGroup variable would get a value like- "Contributor"
```

The `userIdWithProvider` value varies based on the authentication provider configured. This value should have the Entity instance mapped for the repository to be prefixed along with real id of the user.

- If the repository is mapped to **Repository User Provider**, the `userIdWithProvider` will be skeltalist::[GUID]. Here the GUID is of the user john, taken from SKEUsers table present in the Repository DB.

  For example: skeltalist:: C99B4110-C8CC-410B-BF70-93F3A7E34EDF
- If the repository is mapped to the **Active Directory** user provider, the `userIdWithProvider` will be activedirectory::[samAccountName]. Here, samAccountName is an active directory property value.

  For example: activedirectory::john

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SFU.getValueForRightCode()

Use `SFU.getValueForRightCode()` method to get the values for security rights of a user.

Syntax

```
SFU.getValueForRightCode(rights, userIdWithProvider)
```

Parameters

Parameter	Description
rights	Array of Right codes
userIdWithProvider	User Id of the user for whom the roles are fetched. This parameter is optional. If the parameter is not passed, it fetches the roles for the current logged-in user.

Return Value

Values for security right codes.

Example 1

```
// Gets the values for Security Rights for the logged-in user
var rightsValue = SFU.getValueForRightCode(["CanView", "CanEdit"]);
// rightsValue variable would get values of the rights as a key-value pair
var valueForCanViewRight = rightsValue["CanView"];
var valueForCanEditRight = rightsValue["CanEdit"];
```

Example 2

```
// Gets the values for Security Rights for the User with Id "XXXXXXXX-XXXX-XXXX-XXXX-
XXXXXXXXXXXX".
var rightsValue = SFU.getValueForRightCode(["CanView", "CanEdit"], "skeltalist::XXXXXXXX-
XXXX-XXXX-XXXX-XXXXXXXXXXXX");
// rightsValue variable would get values of the rights as a key-value pair
var valueForCanViewRight = rightsValue["CanView"];
var valueForCanEditRight = rightsValue["CanEdit"];
```

The `userIdWithProvider` value varies based on the authentication provider configured. This value should have the Entity instance mapped for the repository to be prefixed along with real id of the user.

- If the repository is mapped to **Repository User Provider**, the `userIdWithProvider` will be skeltalist::[GUID].

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Here the GUID is of the user john, taken from SKEUsers table present in the Repository DB.

For example: skeltalist:: C99B4110-C8CC-410B-BF70-93F3A7E34EDF

- If the repository is mapped to the **Active Directory** user provider, the `userIdWithProvider` will be activedirectory::[samAccountName]. Here, samAccountName is an active directory property value.

  For example: activedirectory::john

Note: The API `SFU.getValueForRightCode()` is applicable only for “Repository” or “Global” types of “Scope of Right” values.

SFU.getVirtualActorId()

Use `getVirtualActorId()`method to get the logged on user ID.

Syntax

```
sRetVal = SFU.getVirtualActorId();
```

Parameters

None.

Return Value

This method returns a string value.

Example

```
// Get the logged on user ID.
SFU.getVirtualActorId();
```

SFU.getXmlFromJSONObject()

Use `getXmlFromJSONObect()` method to convert a JSON object to an XML string.

Syntax

```
sRetVal = SFU.getXmlFromJSONObject(JSONobj);
```

Parameters

Parameter	Description
JSONobj	JSON object to convert to XML string.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Return Value

This method returns an XML string.

Example

```
// Define a JSON object.
var jsonObj = {
 "MyRoot" : {
"FirstName":"John",
"LastName":"Smith",
}
};
// Convert the JSON object to an XML string.
var xmlString = SFU.getXmlFromJSONObject(jsonObj);
// The XML String will be returned as:
// "<MyRoot><FirstName>John</FirstName><LastName>Smith</LastName></MyRoot>"
```

SFU.hideLoader()

Use `hideLoader()` method to hide the loader.

Note: The showLoader and hideLoader APIs have a limitation wherein, the loader is immediately shown or hidden and the number of calls made to show or hide the loader is not considered. Therefore, the number of calls made to the loader cannot be tracked and the show or hide action cannot be performed accordingly.

Syntax

```
SFU.hideLoader();
```

Parameters

None.

Return Value

None.

Example

```
// Hide the loader.
SFU.hideLoader();
```

SFU.includeCustomCssFiles()

Use `includeCustomCssFiles()` method to include custom CSS files in the form.

In the AVEVA Work Tasks Pro app, the API loads the content asynchronously.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Syntax

```
SFU.includeCustomCssFiles(filesArray);
```

Parameters

Parameter	Description
filesArray	Path array of custom CSS Files to include in the form.

Return Value

None

Example

```
// In Form On Load event, include your CSS file as a part of the rendering form.
var files=["custom/css/MyCustomFile.css"];
OR
var files=["custom/css/myCustomFile1.css", "custom/css/myCustomFile2.css"];
```

```
SFU.includeCustomCssFiles(files);
```

Note:
- Update the CSS files in the path: **../<BPMUITemplates>/Default/NextGenForms/custom/css**.
- If Theme is applied, the css file must be tagged with **FormCustomThemePath**, for example, var cssFiles=["FormCustomThemePath:MyCustomStyleSheet.css"]; and must be placed in **../<BPMUITemplates>/Default/NextGenForms/custom/Themes/<ThemeName>/css**.
- If Theme is not applied, the css file can be placed in **../<BPMUITemplates>/Default/NextGenForms/custom/css/MyCustomStyleSheet.css**. You can create folder structure in **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/css/** to place the css files, for example, **../<BPMUITemplates>/Default/NextGenForms/custom/css/Mycssfolder/MyCustomStyleSheet.css**
- For support in the AVEVA Work Tasks Pro app, update the CSS files in the path **../<BPMUITemplates>/Default/NextGenForms/custom/css**.
- For AVEVA Work Tasks Pro app, mapping of themes is not considered.

SFU.includeCustomJsFiles()

Use `includeCustomJsFiles()` method to include custom JS files in the form.

In the AVEVA Work Tasks Pro app, the API loads the content asynchronously.

Syntax

```
SFU.includeCustomJsFiles(filesArray);
```

Parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
filesArray	Path array of custom JS Files to include in the form.

Return Value

None

Example

```
// In Form On Load event, include your JS file as a part of the rendering form.
var files=["custom/js/MyCustomFile.js"];
OR
var files=["custom/js/myCustomFile1.js", "custom/js/myCustomFile2.js"];
SFU.includeCustomJsFiles(files);
```

Note:
- Update the js files in the path: **../<BPMUITemplates>/Default/NextGenForms/custom/js**
- The js file must be tagged with **FormCustomPath** or **FormCustomThemePath.** For example, var jsFiles=["FormCustomThemePath:myCustomFile.js"]; and must be placed in **../<BPMUITemplates>/Default/NextGenForms/custom/js**.
- For support in the AVEVA Work Tasks Pro app, update the js files in the path **../<BPMUITemplates>/Default/NextGenForms/custom/js**.
- For AVEVA Work Tasks Pro app, mapping of themes is not considered.

SFU.initializeWidgetFormCommunication()

Use `SFU.initializeWidgetFormCommunication()` method to establish communication between Widget control and a Form.

Syntax

```
SFU.initializeWidgetFormCommunication(control)
```

Parameters

For information on establishing communication and defining parameters, see Configure Communication between Widget Control and Form.

SFU.invokeWorkflow()

Use `invokeWorkflow()` method to trigger the **Click** event of the **Invoke Workflow** control.

Syntax

```
SFU.invokeWorkflow(control);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameters

Parameter	Description
control	Control which represents the Invoke Workflow control.

Return Value

None.

Example

```
// Trigger the Click event of the Invoke Workflow control.
var invokeWorkflowControl = control.findById("I1");
SFU.invokeWorkflow(invokeWorkflowControl);
```

SFU.isArray()

Use `isArray()` method to determine if an object is an array or not.

Syntax

```
bRetVal = SFU.isArray(object);
```

Parameters

Parameter	Description
object	Object to determine.

Return Value

This method returns a boolean value.

Example

```
// Check if the object is an array.
var rowData = ["a","b","c","d"];
if (SFU.isArray(rowData))
{
// Perform required operations.
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SFU.isControlInNewMode()

Use `isControlInNewMode()` method to determine if a record of a **Base Form** is in new mode where the control exists.

Syntax

```
bRetVal = SFU.isControlInNewMode(control);
```

Parameters

Parameter	Description
control	Control whose mode needs to determined.

Return Value

This method returns a boolean value.

Example

```
// Check whether the control is in new mode.
var chkCtrl = control.findById("T1");
var isNewMode = isControlInNewMode(chkCtrl);
if (isNewMode)
{
// Perform required operations.
}
```

SFU.isEmpty()

Use `isEmpty()` method to determine if the value specified is empty.

Syntax

```
bRetVal = SFU.isEmpty(value);
```

Parameters

Parameter	Description
value	Value to check.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Return Value

This method returns a boolean value.

Example

```
// Define a variable and check if the specified value is empty.
var employeeNameVal = control.findByXmlNode("EmployeeName").value;
var valEmpty = SFU.isEmpty(employeeNameVal);
```

SFU.isFormInEditMode()

Use `isFormInEditMode()` method to determine if a form is in edit mode or not.

Syntax

```
SFU.isFormInEditMode();
```

Parameters

None.

Return Value

This method returns a boolean value.

Example

```
// Check whether the form is in edit mode.
var editableForm = SFU.isFormInEditMode();
if (editableForm){
// Perform required operations.
}
```

SFU.isMobileDisplayMode()

Use `isMobileDisplayMode()` method to determine if a form is rendered in mobile display mode or not.

Syntax

```
bRetVal = SFU.isMobileDisplayMode();
```

Parameters

None.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Return Value

This method returns a boolean value.

Example

```
// Check the display mode of the form.
var isMobileMode = SFU.isMobileDisplayMode();
if (isMobileMode)
{
// Perform required operations.
}
```

SFU.isNumber()

Use `isNumber()` method to determine if a value from a string or number is number.

Syntax

```
bRetVal = SFU.isNumber(value, acceptEmptyString);
```

Parameters

Parameter	Description
value	Value to check.
acceptEmptyString	Pass as **true** to consider an empty string as number. This parameter is optional.

Return Value

This method returns a boolean value.

Example

```
// Check if the value from the string is number.
var valControl = control.findById("T1").value;
var valNumber = SFU.isNumber(valControl);
// Return true if value contains "123" or 123.
// Returns false if value contains "12A3" or ""
// Check if a value from the string is empty and consider it as number.
var valControl = control.findById("T1").value;
var valNumber = SFU.isNumber(valControl,true);
// Returns true if value contains "123", 123,or "".
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SFU.isUndefined()

Use `isUndefined()` method to determine if an object is defined or not or if an object is null.

Syntax

```
bRetVal = SFU.isUndefined(object);
```

Parameters

Parameter	Description
object	Object to determine.

Return Value

This method returns a boolean value.

Example

```
// Get the Country control by XML node.
var countryControlObject = control.findByXmlNode("Country");
// Check whether the Country control exists.
var countryControlExists = SFU.isUndefined(countryControlObject);
if (countryControlExists){
// Perform required operations.
}
```

SFU.isWorkTasksPro()

Use `isWorkTasksPro()` method to determine if the Form is accessed from AVEVA Work Tasks Pro (Mobile application).

Syntax

```
SFU.isWorkTasksPro();
```

Parameters

No Parameters

Return Value

This method returns a boolean value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example

```
if (SFU.isWorkTasksPro())
{
//// Add your script when Form is accessed from AVEVA Work Tasks Pro
}
```

SFU.isWorkTasksProConnectedToInternet()

Use `isWorkTasksProConnectedToInternet()` method to determine if the Form accessed from AVEVA Work Tasks Pro (Mobile application) is connected to the server.

Syntax

```
SFU.isWorkTasksProConnectedToInternet();
```

Parameters

No parameters

Return Value

This method returns a boolean value.

Example

```
if (SFU.isWorkTasksProConnectedToInternet())
{
//// Add your script when Form is accessed from AVEVA Work Tasks Pro and connected to the
server.
}
```

SFU.logMessageToWMLogger()

Use `SFU.logMessageToWMLogger()` method to show log messages in the AVEVA Work Tasks Logger to the user.

Syntax

```
SFU.logMessageToWMLogger(message, logType);
```

Parameters

Parameter	Description
message	Log message to be display to the user.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

logType	Determines the type of log message, whether it is an information, an error, or a warning. For example, - if logType is specified as 1, it is an error message - if logType is specified as 2, it is a warning message - if logType is specified as 3, it is an information message

Return Value

None.

Example

```
// Show the welcome message to the user.
SFU.logMessageToWMLogger("Welcome to AVEVA Work Tasks", 3);
OR
SFU.logMessageToWMLogger("Welcome to AVEVA Work Tasks", "3")
```

Note: This API can be used to send log messages from AVEVA Work Tasks Pro app to AVEVA Work Tasks Logger.

SFU.parseNumberValue()

Use `parseNumberValue()` method to convert the value to number.

If the value is not number, then the method returns the number 0.

Syntax

```
nRetVal = SFU.parseNumberValue(value);
```

Parameters

Parameter	Description
value	Value to be converted to number.

Return Value

This method returns a numeric value.

Example

```
// Convert the value to number.
var txtVal = control.findById("T1").value;
var numVal = SFU.parseNumberValue(txtVal);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// Returns the number for valid number. For example, returns 123 for 123.
// Returns 0 for invalid number. For example, returns 0 for "12A3" or "".
```

SFU.reloadForm()

Use `reloadForm()` method to reload a form to the initial stage.

Syntax

```
SFU.reloadForm();
```

Parameters

None.

Return Value

None.

Example

```
// Reload the form to the initial stage.
SFU.reloadForm();
```

SFU.setStorageItem()

Use `setStorageItem()` method to set key and value for the key in the web storage.

Syntax

```
SFU.setStorageItem(key,value);
```

Parameters

Parameter	Description
key	Key in the web storage.
value	Value for the key in the web storage.

Return Value

None.

Example

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// Set key and value for the key in the web storage.
SFU.setStorageItem("WorkItemData","WO2015");
```

SFU.soaMethod()

Use soaMethod() method to call the SOA method.

Syntax

```
SFU.soaMethod("TitleName", "TitleName.ClassName", "MethodName", [Parameters], true);
```

Parameters

Parameter	Description
TitleName	Name of the Assembly.
ClassName	Name of the Class.
MethodName	Name of the Method.
Parameters	Parameters. Use **[ ]** if there are no parameters.
true/false	Use **true** to show error an alert message in case of any exception. Otherwise, use **false**.

Return Value

This method returns an object.

Example

```
// Call the SOA method to get the age of the employee.
var empName=control.findById("T1").value;
var empAge=control.findById("N1").value;
var empSal=control.findById("N2").value;
var empGender=control.findById("T2").value;
if(empName !== '')
{
var empDetails= SFU.soaMethod("EmpDetails", "EmployeeDetails.Employee",
"GetEmployeeDetailsByName", ['"+empName+"'], true);
return empDetails.Age;
}
else
{
return currentValue;
}
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: You must parse the parameters to match the SOA method parameter types.

SFU.showAlert()

Use `showAlert()` method to show a notification to the user.

Syntax

```
SFU.showAlert(title, message [, callbackHandlerFn]);
```

Parameters

Parameter	Description
title	Title of the message
message	Message to display to the user.
callbackHandlerFn	A function that is triggered after the alert message is closed.

Note: You can include HTML content in the title and message.

Return Value

None.

Example

```
// Show the welcome message to the user.
SFU.showAlert("Hi", "Welcome to AVEVA Work Tasks",
function()
{
//enter code to perform actions after the alert is closed.
});
```

SFU.showConfirmation()

Use `showConfirmation()` method to show a confirmation message to the user.

Note: The API `SFU.showConfirmationAlert()`is deprecated and is replaced by `SFU.showConfirmation()`.

Syntax

```
SFU.showConfirmation(title, message [, callbackHandlerFn]);
```

Parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
title	Title of the message.
message	Message to display to the user.
callbackHandlerFn(val)	A function that is triggered after the confirmation action is performed. Here val is a boolean value that represents the confirmation result.

Note: You can include HTML content in the title and message.

Return Value

None.

Example

```
// Show the confirmation message to the user.
SFU.showConfirmation("Save File", "Do you want to save this file?",
function(val)
{
if (val)
{
//enter code to save the file
}
});
```

SFU.showError()

Use `showError()` method to show an error message to the user.

Syntax

```
SFU.showError(title, message [, callbackHandlerFn [, details]]);
```

Parameters

Parameter	Description
title	Title of the message
message	Message to display to the user.
callbackHandlerFn	A function that is triggered after the error message is closed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
details	More details about the message.

Note: You can include HTML content in the title, message, and details.

Return Value

None.

Example

```
// Show the error message to the user.
SFU.showError("Validation Error", "Please verify the values before you save the form.",
function()
{
//enter code to perform actions after the error message is displayed.
},
"Age is not specified.<br>Date of joining should be earlier than today's date.");
```

SFU.showLoader()

Use `showLoader()` method to show the loader.

Note: The showLoader and hideLoader APIs have a limitation wherein, the loader is immediately shown or hidden and the number of calls made to show or hide the loader is not considered. Therefore, the number of calls made to the loader cannot be tracked and the show or hide action cannot be performed accordingly.

Syntax

```
SFU.showLoader();
```

Parameters

None.

Return Value

None.

Example

```
// Show the loader.
SFU.showLoader();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SFU.showPopupControl()

Use `showPopupControl()` method to show the Pop-Up control.

Syntax

```
SFU.showPopupControl(control);
```

Parameters

Parameter	Description
control	Control which represents the Pop-Up control.

Return Value

None.

Example

```
// Show the Pop-Up control.
var popupControl = control.findByXmlNode("Popupform1");
SFU.showPopupControl(popupControl);
```

SFU.showWarning()

Use `showWarning()` method to show a warning message to the user.

Syntax

```
SFU.showWarning(title, message [, callbackHandlerFn]);
```

Parameters

Parameter	Description
title	Title of the message.
message	Message to display to the user.
callbackHandlerFn(val)	A function that is triggered after the confirmation action is performed. Here val is a boolean value that represents the confirmation result.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: You can include HTML content in the title and message.

Return Value

None.

Example

```
// Show the warning message to the user.
SFU.showWarning("Clear Form", "Clear Form will clear all the contents in the Form. Do you
want to continue?",
function(val)
{
if (val)
{
control.topLevelForm.clearContent();
}
});
```

SFU.subscribeToControlParameters()

Use `subscribeToControlParameters()` method to the subscribe to the parameters of a control so that the callBack Function is called when a value of the parameter control changes.

Syntax

```
SFU.subscribeToControlParameters(control, parameterCollection, functionToBeCalled)
```

Parameters

Parameter	Description
control	Control
parameterCollection	Collection of parameters for parameterized Data Lookup
functionToBeCalled	Function to be called when the value of the parameter control changes

Return Value

None

Example

If a custom control uses Lookup Binding, use this method for subscribing to changes in value of the Parameter

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

control to call the callback function.

see the TreeView JS file corp.customControls.treeView.js for reference.

SFU.updateFormHeaderForWorkTasksPro()

Use `SFU.updateFormHeaderForWorkTasksPro()` method to update the form header.

Syntax

```
SFU.updateFormHeaderForWorkTasksPro(formHeader)
```

Parameters

Parameter	Description
formHeader	The value to be passed to update the form header.

Return Value

None.

Example

```
// Get the value of a key from the locale.
var localizedHeader = skelta.localize.getString("headerKey");
// Set the localized value as the Form header
SFU.updateFormHeaderForWorkTasksPro(localizedHeader);
```

Note: This API is applicable only for AVEVA Work Tasks Pro app.

control.addRecord()

Use `addRecord()` method to add a record to the **Base Form**.

Syntax

```
// B1 is a Base Form object.
B1.addRecord(object);
or
B1.addRecord();
```

Parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
object	Record to add. This parameter is optional.

Return Value

This method returns the new record added.

Example

```
// Define a record to add.
var dataObj = { Name:"Kevin",Address:"Park Street 1" };
// Define a Base Form object.
var B1 = control.findById("B1");
// Add the record to the Base Form.
B1.addRecord(dataObj);
// or B1.addRecord();
```

Note: If you use this method to add a record using scripts for the **On Record Add** property of the **Base Form**, then at run-time a recursive loop will cause the browser to stop responding.

control.addRecordToBottom()

Use **addRecordToBottom()** method to add a record at the end of the existing records on the **Base Form**.

Syntax

```
// B1 is a Base Form object.
B1.addRecordToBottom();
or
B1.addRecordToBottom(object);
```

Parameters

Parameter	Description
object	Record to add. This parameter is optional.

Return Value

This method returns the new record added.

Example

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// Define a record to add.
var dataObj = { Name:"Kevin",Address:"Park Street 1" };
// Define a Base Form object.
var B1 = control.findById("B1");
// Add the record to the Base Form.
B1.addRecordToBottom(dataObj);
// or B1.addRecordToBottom();
```

Note: If you use this method to add a record using scripts for the **On Record Add** property of the **Base Form**, then at run-time a recursive loop will cause the browser to stop responding.

control.addRecordToPosition()

The `addRecordToPosition()` method to add a record at a specific position on the **Base Form**.

Syntax

```
// B1 is a Base Form object.
B1.addRecordToPosition(index);
or
B1.addRecordToPosition(index,object);
```

Parameters

Parameter	Description
index	Specifies the position of the object to be added.
object	Record to add. This parameter is optional.

Return Value

This method returns the new record added.

Example

```
// Define a record to add.
var dataObj = { Name:"Kevin",Address:"Park Street 1" };
// Define a Base Form object.
var B1 = control.findById("B1");
// Add the record to a position on the Base Form.
B1.addRecordToPosition(2,dataObj);
// or B1.addRecordToPosition(2);
```

Note: If you use this method to add a record using scripts for the **On Record Add** property of the **Base Form**, then at run-time a recursive loop will cause the browser to stop responding.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

control.addRecordToTop()

Use `addRecordToTop()` method to add a record at the top of the existing records on the **Base Form**.

Syntax

```
// B1 is a Base Form object.
B1.addRecordToTop(object);
or
B1.addRecordToTop();
```

Parameters

Parameter	Description
object	Record to add. This parameter is optional.

Return Value

This method returns the new record added.

Example

```
// Define a record to add.
var dataObj = { Name:"Kevin",Address:"Park Street 1" };
// Define a Base Form object.
var B1 = control.findById("B1");
// Add the record to the Base Form.
B1.addRecordToTop(dataObj);
// or B1.addRecordToTop();
```

Note: If you use this method to add a record using scripts for the **On Record Add** property of the **Base Form**, then at run-time a recursive loop will cause the browser to stop responding.

control.applyFilter()

Use `control.applyFilter()` to specify search criteria on the hierarchical list at the first level.

Syntax

```
control.applyFilter(functionToBeCalled);
```

Parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
functionToBeCalled	Function to be called when search criteria is set.

Return Value

This method returns a boolean value.

Example

```
// Filter records where the ProcessStatus column is set to 1.
control.findByID("H1").applyFilter(
function(currentVal)
{
if(currentVal.ProcessStatus == 1)
{
return true;
}
else
{
return false;
}
}
);
// Shows records in the hierarchical list matching the search criteria.
```

control.associatedForm

Use `associatedForm` method to get the immediate record form object.

Syntax

```
control.associatedForm;
```

Parameters

None.

Return Value

This method returns a control object.

Example

```
// Consider a Base Form with Text controls for
// Employee ID, First Name, and Last Name, and a
// Button control for displaying the employee names.
// Define On Click script for the Button control.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
baseFormObj.items()[0].findByXmlNode("EmployeeID");
// Get the associated record to which the Button control belongs to.
var associatedForm = control.associatedForm;
// Display the employee names.
skeltaUtils.showAlert("Employee Name is " + associatedForm.findByXmlNode("FirstName").value
+ " " );
```

For more examples, refer to Delete Record in a Base Form Control.

control.clearContent()

Use **clearContent(event)** method to clear values of controls on a record or all records on a **Base Form**, clear values of controls on a **Panel Form**, or clear values of controls on a form.

Syntax

```
control.clearContent();
```

Parameters

None.

Return Value

None.

Example

```
// Clear the values of controls on the record of the Base Form.
control.findById("B1").records()[0].clearContent();
// Clear the values of controls on the Base Form.
control.findById("B1").clearContent();
// Clear the values of controls on the Panel Form.
control.findById("P1").clearContent();
// Clear the values of controls on the form.
control.topLevelForm.clearContent();
```

control.clearFilter()

Use `control.clearFilter()` to clear search criteria on the hierarchical list.

Syntax

```
control.clearFilter();
```

Parameters

None.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Return Value

None.

Example

```
// Clears search criteria on the hierarchical list.
control.clearFilter()
```

control.controls()

Use `control.controls()` property to get the array of controls in a record of a container.

Here controls is applicable only to containers and its records.

Syntax

```
control.controls();
```

Parameters

None.

Return Value

This property returns an array of control object.

Example

```
// Get the array of controls in the associated record of the control in the Base Form.
var ctrlArr = control.associatedForm.controls;
// which is similar to
// var ctrlArr = control.findById("B1").records()[1].controls;
for (var i = 0; i < ctrlArr.length ; i++)
{
// Perform required operation on each control.
}
```

control.dataContainerForm

Use `dataContainerForm` method to get the immediate container form object.

Syntax

```
control.dataContainerForm;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameters

None.

Return Value

This method returns a control object.

Example

```
// Consider a Base Form with Text controls for
// Employee ID, First Name, and Last Name, and a
// Button control for looking up duplicate names in the form.
// Define On Click script for the Button control.
baseFormObj.items()[0].findByXmlNode("EmployeeID");
// Get the associated record to which the Button control belongs to.
var associatedForm = control.associatedForm;
// Get the First Name for comparison.
var firstName = associatedForm.findByXmlNode("FirstName").value;
// Get the Last Name for comparison.
var lastName = associatedForm.findByXmlNode("LastName").value;
// Get the main container form which contains all the records.
var dataContainerForm = control.dataContainerForm;
var records = dataContainerForm.records();
var totalRecords = records.length;
// Define record index which needs to be skipped for comparison.
var currentRecordIndex = records.indexOf(associatedForm);
// Define duplicates found flag.
var duplicatesFound = false;
// Traverse through the records.
for (var i = 0; i < totalRecords; i++)
{
  // Skip the record index, which needs to be skipped for comparison.
  if (i == currentRecordIndex)
{
  continue;
}
  var recordObj = records[i];
  // Compare the first and the last names.
  if ((recordObj.findByXmlNode("FirstName").value == firstName) &&
(recordObj.findByXmlNode("LastName").value == lastName))
{
// Set the duplicate flag.
duplicatesFound = true;
  break;
}
}
// Check the duplicate flags and show appropriate messages.
if (duplicatesFound)
{
skeltaUtils.showAlert("Duplicates Found!");
}
else
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
skeltaUtils.showAlert("No Duplicates Present!");
}
```

Refer to Delete Record in a Base Form Control for more example(s).

control.findAllById()

Use `findAllById()` method to find an array of controls with a specific ID on the form.

Syntax

```
control.findAllById(controlID);
```

Parameters

Parameter	Description
controlID	ID of the control to search on the form.

Return Value

This method returns an array of control objects.

Example

```
// Find all Text controls in the same and lower hierarchy on the form.
control.findAllById("T1");
// Find all Text controls anywhere on the form.
control.findAllById("T1",1);
// Find all Text controls in the same hierarchy on the form.
control.findAllById("T1",2);
```

control.findAllByName()

Use `findAllByName()` method to find an array of controls with a specific name on the form.

Syntax

```
control.findAllByName(controlname);
```

Parameters

Parameter	Description
Control Name	Name of the control to search on the form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Return Value

This method returns an array of control objects.

Example

```
// Find all controls in the same and lower hierarchy on the form.
control.findById("B1").findAllByName("Name");
// Find all controls anywhere on the form.
control.findById("B1").findAllByName("Name",1);
// Find all controls in the same hierarchy on the form.
control.findById("B1").findAllByName("Name",2);
```

control.findAllByXmlNode()

Use `findAllByXmlNode()` method to find an array of controls with a specific XML node on the form.

Syntax

```
control.findAllByXmlNode(mappedXMLnode);
```

Parameters

Parameter	Description
mappedXMLnode	XML node of the control to search on the form.

Return Value

This method returns an array of control objects.

Example

```
// Find all controls in the same and lower hierarchy on the form.
control.findById("B1").findAllByXmlNode("EmployeeName");
// Find all controls anywhere on the form.
control.findById("B1").findAllByXmlNode("EmployeeName",1);
// Find all controls in the same hierarchy on the form.
control.findById("B1").findAllByXmlNode("EmployeeName",2);
```

control.findById()

Use `findById()` method to find a control with a specific ID on the form.

Syntax

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.findById(controlID);
```

Parameters

Parameter	Description
controlID	ID of the control to search on the form.

Return Value

This method returns a control object.

Example

```
// Set a value for the Text control
control.findById("T1").value = "Hello World";
```

control.findByName()

Use `findByName()` method to find a control with a specific name on the form.

Syntax

```
control.findByName(controlname);
```

Parameters

Parameter	Description
controlname	Name of the control to search on the form.

Return Value

This method returns a control object.

Example

```
// Set a value for the Text control.
control.findByName("Employee Name").value = "John Dow";
```

control.findByXmlNode()

Use `findByXmlNode()` method to find a control with a specific XML node on the form

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Syntax

```
control.findByXmlNode(mappedXMLnode);
```

Parameters

Parameter	Description
mappedXMLnode	XML node of the control to search on the form.

Return Value

This method returns a control object.

Example

```
// Set a value for the Text control.
control.findByXmlNode("EmployeeName").value = "John Dow";
```

control.findChildRecordControlById()

Use `findChildRecordControlById()` method to find a control with a specific ID in the child record and its children.

Syntax

```
control.findChildRecordControlById(controlID);
```

Parameters

Parameter	Description
controlID	ID of the control to search in the child record.

Return Value

This method returns a control object.

Example

```
// Set a value for the Text control.
control.findChildRecordControlById("T1").value = "Hello World";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

control.findChildRecordControlByName()

Use `findChildRecordControlByName()` method to find a control with a specific name in the child record and its children.

Syntax

```
control.findChildRecordControlByName(controlname);
```

Parameters

Parameter	Description
controlname	Name of the control to search in the child record.

Return Value

This method returns a control object.

Example

```
// Set a value for the Text control.
control.findChildRecordControlByName("Employee Name").value = "John Doe";
```

control.findChildRecordControlByXmlNode()

Use `findChildRecordControlByXmlNode()` method to find a control with a specific XML node in the child record and its children.

Syntax

```
control.findChildRecordControlByXmlNode(mappedXMLnode);
```

Parameters

Parameter	Description
mappedXMLnode	XML node of the control to search in the child record.

Return Value

This method returns a control object.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example

```
// Set a value for the Text control.
control.findChildRecordControlByXmlNode("EmployeeName").value = "John Doe";
```

control.findImmediateChildRecordControlById()

Use `findImmediateChildRecordControlById()` method to find a control with a specific ID in the child record (but not its children) of a container.

Here controls is applicable only to containers and its records.

Syntax

```
control.findImmediateChildRecordControlById(controlID);
```

Parameters

Parameter	Description
controlID	ID of the control to search in the immediate child record but not its children.

Return Value

This method returns a control object.

Example

```
// Set a value for the Text control.
control.findImmediateChildRecordControlById("T1").value = "Hello World";
```

control.findImmediateChildRecordControlByName()

Use `findImmediateChildRecordControlByName()` method to find a control with a specific name in the child record (but not its children) of a container.

Here controls is applicable only to containers and its records.

Syntax

```
control.findImmediateChildRecordControlByName(controlname);
```

Parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
controlname	Name of the control to search in the immediate child record but not its children.

Return Value

This method returns a control object.

Example

```
// Set a value for the Text control.
control.findImmediateChildRecordControlByName("Employee Name").value = "John Doe";
```

control.findImmediateChildRecordControlByXmlNode()

Use `findImmediateChildRecordControlByXmlNode()` method to find a control with a specific XML node in the child record (but not its children) of a container.

Here controls is applicable only to containers and its records.

Syntax

```
control.findImmediateChildRecordControlByXmlNode(mappedXMLnode);
```

Parameters

Parameter	Description
mappedXMLnode	XML node of the control to search in the immediate child record but not its children.

Return Value

This method returns a control object.

Example

```
// Set a value for the Text control.
control.findImmediateChildRecordControlByXmlNode("EmployeeName").value = "John Doe";
```

control.getFormSubmissionData()

Use `getFormSubmissionData()` method to get form data in JSON format from a form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Here form is applicable only to top level form.

from a top level form.

Syntax

```
sRetVal = control.getFormSubmissionData();
```

Parameters

None.

Return Value

This method returns a JSON object.

Example

```
// Get form data in JSON format from the top level form.
var instanceJSON = control.topLevelForm.getFormSubmissionData();
// convert JSON data to XML string
var instanceXml = SFU.getXmlFromJSONObject(instanceJSON);
// Assign the form instance XML string to a multi-line Text control with XML node set as
SubmittedFormXml.
control.findByXmlNode("SubmittedFormXml").value = instanceXml;
```

control.getType()

Use `getType()` method to determine the type of a control on form.

Syntax

```
// B1 is a Base Form object.
tRetVal = control.findById("B1").getType();
```

Parameters

None.

Return Value

This method returns the type of control.

Example

```
// Define a variable and get the type of control.
var controlType = control.findById("B1").getType();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

control.refreshCard()

Use `refreshCard()` method to reload data in a card.

Syntax

```
control.refreshCard();
```

Parameters

None.

Return Value

None.

Example

```
// Reload data in a Card.
control.findByXmlNode("Card1").refreshCard();
```

control.refreshData()

Use `refreshData()` method to reload data in a gantt chart.

Syntax

```
control.refreshData();
```

Parameters

None.

Return Value

None.

Example

```
// Reload data in a Gantt Chart.
control.findByXmlNode("Gantt3").refreshData();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

control.refreshGrid()

Use `refreshGrid()` method to reload records in a data grid.

Syntax

```
control.refreshGrid();
```

Parameters

None.

Return Value

None.

Example

```
// Reload records in a data grid.
control.findById("G1").refreshGrid();
```

control.refreshTree()

Use `refreshTree()` method to reload a tree.

Syntax

```
control.refreshTree();
```

Parameters

None.

Return Value

None.

Example

```
// Reload records in a data grid.
control.findById("T1").refreshTree();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

control.remove()

Use `remove()` method to delete a control from a record on the Base Form.

Syntax

```
// B1 is a Base Form object.
var recordToBeDeleted = B1.records()[1];
recordToBeDeleted.remove();
```

Parameters

None.

Return Value

None.

Example

```
// Define a Base Form object.
var B1 = control.findById("B1");
// Set the control to be deleted from the record of the Base Form.
var recordToBeDeleted = B1.records()[1];
// Delete the control from the record of the Base Form.
recordToBeDeleted.remove();
```

Refer to Delete Record in a Base Form Control for more example(s).

control.removeAll()

Use `removeAll()` method to delete all controls from a record on the Base Form.

Syntax

```
// B1 is a Base Form object.
control.findById("B1").removeAll();
```

Parameters

None.

Return Value

None.

Example

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// Define a Base Form object.
var B1 = control.findById("B1");
// Delete all controls from the record of the Base Form.
B1.removeAll();
```

control.removeAt()

Use `removeAt()` method to delete a record from a specific position on the Base Form.

Syntax

```
// B1 is a Base Form object.
control.findById("B1").removeAt(position);
```

Parameters

Parameter	Description
position	Specifies the position of the object to be removed.

Return Value

None.

Example

```
// Define a Base Form object.
var B1 = control.findById("B1");
// Delete a record from a position on the Base Form.
B1.removeAt(1);
```

Refer to Delete Record in a Base Form Control for more example(s).

control.removeRecord()

Use `removeRecord()` method to delete a record from the Base Form.

Syntax

```
// B1 is a Base Form object.
B1.removeRecord(object);
```

Parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
object	Record to delete. This parameter is optional.

Return Value

None.

Example

```
// Define a Base Form object.
var B1 = control.findById("B1");
// Remove the record from the Base Form.
B1.removeRecord(Object);
```

For more examples, refer to Delete Record in a Base Form Control.

control.submitClick()

Use `submitClick()` method to submit a record on a **Base Form** or **submit a Base Form**.

Syntax

```
control.submitClick(event);
```

Parameters

Parameter	Description
event	Specifies the event handler.

Return Value

None.

Example

```
// Submit the record on the Base Form.
control.associatedForm.submitClick(event);
// Submit the Base Form.
control.dataContainerForm.submitClick(event);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

control.selectedTabIndex()

Use `control.selectedTabIndex()` to get the index of the selected tab in the **Tabs** container.

Syntax

```
nRetVal = control.selectedTabIndex();
```

Parameters

None.

Return Value

This method returns a numeric value.

Example

```
// Define a variable and get the index of the selected tab in the Tabs container.
var tabIndex = control.findById("T1").selectedTabIndex;
```

control.selectRecord()

Use `selectRecord()` method to select a specific record from the **On Data Bound** event of the **Data Grid**.

Syntax

```
control.selectRecord(e,recordNumber);
```

Parameters

Parameter	Description
e	Constant for events triggered in the data grid such as paging, filtering, sorting, and so on.
recordNumber	Specifies the record number of the record in the data grid to be selected.

Return Value

None.

Example

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// Select the second record in the Data Grid.
control.selectRecord(e,2);
```

Note:
- In the **Data Grid** control, the **Select First Record** property takes precedence over the script to select a record in the **On Data Bound** event.
- For example, in a **Data Grid** control when the **Select First Record** property is set to **Yes** and a script in the **On Data Bound** event is set to select the second record, the first record is selected.
- Therefore, `selectRecord()` method and the **Select First Record** property must not be used together.

control.selectTab()

Use `control.selectTab()` method to select a specific tab, that is switch to a specific tab, in the **Tabs** container.

Syntax

```
control.selectTab(index);
```

Parameters

Parameter	Description
index	Specifies the index of the tab to be selected. **Note:** Index starts from 0.

Return Value

None.

Example

```
// Select the second tab in the Tabs container.
control.findById("T1").selectTab(1);
```

control.submitForm()

Use `submitForm()` method to submit a form.

Syntax

```
control.submitForm(event);
```

Parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
event	Specifies the event handler.

Return Value

None.

Example

```
// Submit the form.
control.submitForm(event);
```

control.sumById()

Use `sumById()` method to get the sum of the value of all controls with a specific ID on the form.

Syntax

```
// N1 is a Number object.
nRetVal = control.findById("N1").sumById();
```

Parameters

None.

Return Value

This method returns a numeric value.

Example

```
// Define a variable and get the sum of the value of all controls with a specific ID.
var sum = control.findById("N1").sumById();
// Add all Number controls in the same and lower hierarchy on the form.
control.findById("N1").sumById("N1");
// Add all Number controls anywhere on the form.
control.findById("N1").sumById("N1",1);
// Add all Number controls in the same hierarchy on the form.
control.findById("N1").sumById("N1",2);
```

control.sumByName()

Use `sumByName()` method to get the sum of the value of all controls with a specific name on the form.

Syntax

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// N1 is a Number object.
nRetVal = control.findById("N1").sumByName();
```

Parameters

None.

Return Value

This method returns a numeric value.

Example

```
// Define a variable and get the sum of the value of all controls with a specific name.
var sum = control.findByName("Amount").sumByName();
// Add all Number controls in the same and lower hierarchy on the form.
control.findById("N1").sumByName("Amount");
// Add all Number controls anywhere on the form.
control.findById("N1").sumByName("Amount",1);
// Add all Number controls in the same hierarchy on the form.
control.findById("N1").sumByName("Amount",2);
```

control.sumByXmlNode()

Use `sumByXmlNode()` method to get the sum of the value of all controls with a specific XML node on the form.

Syntax

```
// N1 is a Number object.
nRetVal = control.findById("N1").sumByXmlNode();
```

Parameters

None.

Return Value

This method returns a numeric value.

Example

```
// Define a variable and get the sum of the value of all controls with a specific XML node.
var sum = control.findByXmlNode("Amount").sumByXmlNode();
// Add all Number controls in the same and lower hierarchy on the form.
control.findById("N1").sumByXmlNode("Amount");
// Add all Number controls anywhere on the form.
control.findById("N1").sumByXmlNode("Amount",1);
// Add all Number controls in the same hierarchy on the form.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.findById("N1").sumByXmlNode("Amount",2);
```

control.topLevelForm()

Use `topLevelForm()` property to find the parent or top level form.

Syntax

```
control.topLevelForm();
```

Parameters

None.

Return Value

This property returns a control object.

Example

```
// Get parameters of the form.
control.topLevelForm.formParameters["<parameter_name>"].value
// The parameters of the form will be returned as:
// <value_of_the_parameter>
```

control.topLevelForm.actionbar.getActionItemByName()

Use `control.topLevelForm.actionbar.getActionItemByName()` method to get the Action Item by design-time name. Using this API, you can change the **Visible** and/or **Enable** properties of the Action Item as shown in the example below.

Syntax

```
var actionItem = control.topLevelForm.actionbar.getActionItemByName("actionItemName");
```

Parameters

Parameter	Description
actionItemName	Name of the action item.

Return Value

None.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Examples

Toggle the Visible property of the Action Item using a Button Click script

```
//New Action Item1 represents the name of the Action Item.
var actionItem = control.topLevelForm.actionbar.getActionItemByName("New Action Item1");
if(!SFU.isUndefined(actionItem))
{
actionItem.visible= !actionItem.visible;
}
```

Toggle the Enable property of the Action Item using a Button Click script

```
//New Action Item1 represents the name of the Action Item.
var actionItem = control.topLevelForm.actionbar.getActionItemByName("New Action Item1");
if(!SFU.isUndefined(actionItem))
{
actionItem.enable= !actionItem.enable;
}
```

control.updateLevel1ProgressDisplay()

Use `control.updateLevel1ProgressDisplay()` to to update a hierarchical list at the first level.

Syntax

```
control.updateLevel1ProgressDisplay(indicatorColor, progressValue, progressDescription,
progressColor);
```

Parameters

Parameter	Description
`indicatorColor`	Indicator color for the control
`progressValue`	Progress value for the control
`progressDescription`	Progress description for the control
`progressColor`	Progress color for the control

Return Value

None.

Example

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
//Set indicator color, progress value, description, and color for the hierarchical list at
the first level.
control.findByName("Hierarchical List HL").updateLevel1ProgressDisplay("#5482AB", 25,
"Warning", "Red");
```

control.updateLevel2()

Use `control.updateLevel2()` method to reload records in a hierarchical list at the second level.

Syntax

```
control.updateLevel2();
```

Parameters

None.

Return Value

None.

Example

```
// Reload records in a hierarchical list at the second level.
control.findByXmlNode("Hierarchicallist1").updateLevel2();
```

```
//Update children for a hierarchical list at the first level.
//Note: You cannot update children for a hierarchical list at the second level.
for (var i = 0; i < control.findByXmlNode("Hierarchicallist1").children.length; i++)
{
control.findByXmlNode("Hierarchicallist1").children.progress[i].indicatorColor = "Green";
control.findByXmlNode("Hierarchicallist1").children.progress[i].progressValue = value;
control.findByXmlNode("Hierarchicallist1").children.toggle[i].level2ToggleEnable = true;
}
control.findByXmlNode("Hierarchicallist1").updateLevel2();
```

```
//Update siblings for the hierarchical list at the second level.
for (var i = 0; i < control.findByXmlNode("Hierarchicallist1").sibling.length; i++)
{
control.findByXmlNode("Hierarchicallist1").sibling.progress[i].indicatorColor = "#252ABA";
control.findByXmlNode("Hierarchicallist1").sibling.progress[i].progressValue = 98;
control.findByXmlNode("Hierarchicallist1").sibling.toggle[i].level2ToggleEnable = true;
}
control.findByXmlNode("Hierarchicallist1").updateLevel2();
```

control.updateLevel2ProgressDisplay()

Use `control.updateLevel2ProgressDisplay()` to to update a hierarchical list at the second level.

Syntax

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.updateLevel2ProgressDisplay(indicatorColor, progressValue, progressDescription,
progressColor);
```

Parameters

Parameter	Description
indicatorColor	Indicator color for the control
progressValue	Progress value for the control
progressDescription	Progress description for the control
progressColor	Progress color for the control

Return Value

None.

Example

```
//Set indicator color, progress value, description, and color for the hierarchical list at
the second level.
control.findByName("Hierarchical List HL").updateLevel2ProgressDisplay("#5482AB", 25,
"Warning", "Red");
```

control.updateLevel2SelectItem()

Use `control.updateLevel2SelectItem()` method to select records in a hierarchical list at the second level.

Syntax

```
control.updateLevel2SelectItem(object);
```

Parameters

Parameter	Description
object	Item to update.

Return Value

None.

Example

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// Select each of the sibling and enable slide-toggle.
for (var i=0; i<selectItem.siblings.toggle.length; i++)
{
selectItem.siblings.toggle[i].level2ToggleEnable = true;
}
control.updateLevel2SelectItem(object);
```

control.updateLevel2ToggleDisplay()

Use `control.updateLevel2ToggleDisplay()` to update toggle values for a hierarchical list at the second level.

Syntax

```
control.updateLevel2ToggleDisplay(enable, visible);
```

Parameters

Parameter	Description
`enable`	Enable or disable the slide-toggle. • Set **true** to enable the slide toggle. • Set **false** to disable the slide-toggle.
`visible`	Show or hide the slide-toggle. • Set **true** to show the slide toggle. • Set **false** to hide the slide-toggle.

Return Value

None.

Example

```
//Enable and show the slide-toggle for hierarchical list at the second level.
control.findByXmlNode("Hierarchicallist1").updateLevel2ToggleDisplay(true,true);
```

control.updateLevel3()

Use `control.updateLevel3()` method to reload records in a hierarchical list at the third level.

Syntax

```
control.updateLevel3();
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameters

None.

Return Value

None.

Example

```
// Reload records in a hierarchical list at the third level.
control.findByXmlNode("Hierarchicallist1").updateLevel3();
```

```
//Update children for a hierarchical list at the first level.
//Note: You cannot update children for a hierarchical list at the second level.
for (var i = 0; i < control.findByXmlNode("Hierarchicallist1").children.length; i++)
{
control.findByXmlNode("Hierarchicallist1").children.progress[i].indicatorColor = "Green";
control.findByXmlNode("Hierarchicallist1").children.progress[i].progressValue = value;
control.findByXmlNode("Hierarchicallist1").children.toggle[i].level2ToggleEnable = true;
}
control.findByXmlNode("Hierarchicallist1").updateLevel2();
```

```
//Update siblings for the hierarchical list at the third level.
for (var i = 0; i < control.findByXmlNode("Hierarchicallist1").sibling.length; i++)
{
control.findByXmlNode("Hierarchicallist1").sibling.progress[i].indicatorColor = "#252ABA";
control.findByXmlNode("Hierarchicallist1").sibling.progress[i].progressValue = 98;
control.findByXmlNode("Hierarchicallist1").sibling.toggle[i].level3ToggleEnable = true;
}
control.findByXmlNode("Hierarchicallist1").updateLevel3();
```

control.updateLevel3ProgressDisplay()

Use `control.updateLevel3ProgressDisplay()` to to update a hierarchical list at the third level.

Syntax

```
control.updateLevel3ProgressDisplay(indicatorColor, progressValue, progressDescription,
progressColor);
```

Parameters

Parameter	Description
`indicatorColor`	Indicator color for the control
`progressValue`	Progress value for the control
`progressDescription`	Progress description for the control

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
progressColor	Progress color for the control

Return Value

None.

Example

```
//Set indicator color, progress value, description, and color for the hierarchical list at
the third level.
control.findByName("Hierarchical List HL").updateLevel3ProgressDisplay("#5482AB", 25,
"Warning", "Red");
```

control.updateLevel3SelectItem()

Use `control.updateLevel3SelectItem()` method to select records in a hierarchical list at the third level.

Syntax

```
control.updateLevel3SelectItem(object);
```

Parameters

Parameter	Description
object	Item to update.

Return Value

None.

Example

```
// Select each of the sibling and enable slide-toggle.
for (var i=0; i<selectItem.siblings.toggle.length; i++)
{
selectItem.siblings.toggle[i].level3ToggleEnable = true;
}
control.updateLevel3SelectItem(object);
```

control.updateLevel3ToggleDisplay()

Use `control.updateLevel3ToggleDisplay()` to update toggle values for a hierarchical list at the third level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Syntax

```
control.updateLevel3ToggleDisplay(enable, visible);
```

Parameters

Parameter	Description
`enable`	Enable or disable the slide-toggle. • Set **true** to enable the slide toggle. • Set **false** to disable the slide-toggle.
`visible`	Show or hide the slide-toggle. • Set **true** to show the slide toggle. • Set **false** to hide the slide-toggle.

Return Value

None.

Example

```
//Enable and show the slide-toggle for hierarchical list at the third level.
control.findByXmlNode("Hierarchicallist1").updateLevel3ToggleDisplay(true,true);
```

control.validateControl()

Use `validateControl()` method to validate a control, and all controls in a container control.

Syntax

```
bRetVal = control.validateControl();
```

Parameters

None.

Return Value

This method returns a boolean value.

Example

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
// Validate the Text control.
control.findById("T1 ").validateControl();
```

control.validateForm()

Use `validateForm()` method to method to validate a form.

Syntax

```
bRetVal = control.validateForm();
```

Parameters

None.

Return Value

This method returns a boolean value.

Example

```
// Validate the Base Form.
control.dataContainerForm.validateForm();
// Validate all the forms.
control.topLevelForm.validateForm();
```

control.validateRecord()

Use `validateRecord()` method to validate a record on a **Base Form**.

Syntax

```
bRetVal = control.validateRecord();
```

Parameters

None.

Return Value

This method returns a boolean value.

Example

```
// Validate the record on the Base Form.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.associatedForm.validateRecord();
```

isControlVisible()

Use `isControlVisible()` method to determine if the control is visible.

Syntax

```
bRetVal = isControlVisible(control);
```

Parameters

Parameter	Description
control	Control object.

Return Value

This method returns a boolean value.

Example

```
// Define a variable and check if the specified control is visible.
var ctr = control.findById("T1");
var isVisible = isControlVisible(ctr);
if(isVisible)
{
control.findById("T1").value = "Some Value";
}
```

skelta.localize.getString()

Use `getString()` method to get the value of a key from the locale.

Syntax

```
sRetVal = skelta.localize.getString("key");
```

Parameters

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Parameter	Description
key	Key of the locale string.

Return Value

This method returns a string value.

Example

```
// Get the value of a key from the locale.
var localeVal = skelta.localize.getString("key ");
```

Note: To get the value of the key from the locale, update the local key value in the path: **..Locales\en-US\Forms2\Standard**
localization_key:Localized Name - Here, localization_key is the 'key' and Localized Name is the value

skelta.userContext.getUserContextFor()

Use `getUserContextFor()` method to get the form information.

Syntax

```
sRetVal = skelta.userContext.getUserContextFor(formConstants);
```

Parameters

Parameter	Description
formConstants	Form Constants.
	"appN" – Application Name "uId" – User ID "vId" – Virtual Actor ID "fName" – Form Name "vStamp" – Form Version "itemId" – Item ID "listName" – List Name "themePath" – Theme Path "culture" – Culture

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

	"deviceId" – Device ID (applicable only for AVEVA Work Tasks Pro app) "accessToken" – Access Token of the logged-in user

Return Value

This method returns a string value.

Example

```
// Get the application name of the form.
var applicationName = skelta.userContext.getUserContextFor("appN");
```

Note: For this API, the Access token constant is not supported in System Platform.

skelta.grid.getDataSource()

Use `getDataSource()` method in **On Data Bound** event of a Data Grid to get the name of the data source. Here the data source is the lookup consumed in the grid configuration.

Syntax

```
var dataSource = skelta.grid.getDataSource(e.sender);
```

Parameters

Parameter	Description
sender	sender from the event parameter (e) in On Data Bound script.

Return Value

This method returns a string value.

Example

// This prints the lookup name in the browser console.

```
var dataSource = skelta.grid.getDataSource(e.sender);
console.log(dataSource);
```

Web APIs

Web APIs are frameworks to build HTTP services for browsers and mobile devices. Use Web APIs to customize

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

your business applications.

To add a Web API

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Web API**.

   The **Web API List** page appears.
2. Click **New**.

   The **Web API List Item** wizard appears.
3. In the **Name** box, type a name for the API.
4. In the **Description** box, type a description for the API.
5. In the **Source** box, select the Web API.
   - If you select the Web API, you need to provide the **Base Address**, **Request URI**, and **Authentication Mode**. For more information, see Configure Web API
   - If you select the OpenAPI, the **Advanced Option** is available. Select **Yes** to provide the **Base Address**, **Request URI**, and **Authentication Mode**.

     The SwaggerUrl is displayed when the Web API is configured from Swagger. The Web API schema is fetched from Swagger file. For more information, see Web API Type files for Swagger/OpenAPI based Web APIs
6. In the **Base Address** box, type a URL in the format: **http://localhost/MySite/**.

   You can also enter an IPv6 URL for the **Base Address**. The IPv6 address must be enclosed within square brackets.
7. In the **Request URI** box, type a URI. This is an optional parameter.

   For example, **api/product/1** for **http://localhost/MySite/api/product/1** (for Request URI without parameters) or **api/{product}/{ID}** for **http://localhost/MySite/api** (for Request URI with parameters).
8. Select one of the following options for **Authentication Mode**:
   - **No Authentication**: If the Web API does not require authentication.
   - **Windows**: If the Web API is Windows authenticated.
   - **User Access Token**: If the Web API requires the currently logged-in user's access token.

     Note: If User Access Token is configured for the Web API and associated with Scheduler Event Association, the workflow fails.
   - **Basic Authentication**: If the Web API requires basic authentication. Provide the user name with domain and the password.
   - **Certificate**: If the Web API requires certificate authentication. Provide the certificate store location and the thumbprint. For more information on certificate-based authentication, see Web API Certificate-based Authentication.

     Note: The thumbprint should not contain any spaces. In some machines, the characters in thumbprint should be in upper case.
   - **Service to Service Access Token**: If the Web API requires Service to Service Access Token. To get the Access Token, provide **Authority URL**, **Audience**, **Client ID**, **Client Secret**, and **Scope**.

     In the **Authority URL**, provide the token end-point of the respective repository provider. For example,

     AVEVA Connect: `{authority url}/oauth/token`

     Azure Active Directory: `{authority url}/oauth2/v2.0/token`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

AVEVA Identity Manager: `{authority url}/connect/token`

9. Do one of the following:
   - Test the Web API server connection.
     a. Click **Test Server**. An information message appears.

     Note:
     - The **Request URI** and **Authentication Mode** details are not required to test the Web API connection.
     - The Web API connection is successful if the site is pointing to the default IIS port, the Server is running but the Application Pool is stopped.
     - The Web API connection is unsuccessful if the Server, Site, or IIS is stopped.
   - Preview the data fetched using the Web API.
     a. Click **Preview**. The Preview dialog box appears.

     Preview is available only for **.json** file format.

     Note: When you click **Preview**,
     - If the Web API is configured with **Headers**, the option **Show Headers** is available, and you can select **Yes** to view the desired Keys and Values. But, if the Web API does not have **Headers** configured, the data fetched from the Web API gets listed directly.
     - For both Web API and OpenAPI, you need to provide the Key and Value for the Headers. These are the mandatory properties.

     b. Check the data fetched using the Web API.
     c. Click **Close** (or click the **Close** button). The Preview dialog box is closed.
   - Save the Web API.
     a. **Click Save & Close**. The API is added, and an information message appears.
     b. Click **OK** (or click the **Close** button). The information message is closed.

Note:
- You can pass parameters in the **Request URI** using braces **{ }**.
- For example, **api/{product}/{ID}** for **http://localhost/MySite/api.**
- If the Web API is Windows authenticated, then the status of the **Windows Authentication** must be set to **Enabled** for the **EnterpriseConsole** site under **AVEVAWorkTasksSites** site in the **IIS Manager**.
- If a Web API method (action) is modified, then you must create the web API List Item, the Web API Lookup, or the Web API activity again.
- Web APIs can recognize the response received in the ODATA (Open Data Protocol) format.
- The **Preview** option is available only for Web APIs, which support JSON format.
- The **Preview** button is enabled only when Request URI is provided.
- The data fetched using Web API would not consider the column names containing characters **: /** and ****.
- The password should not contain **<** and **>** characters.

Web API Certificate-based Authentication

You can authenticate the Web API using following configurations:

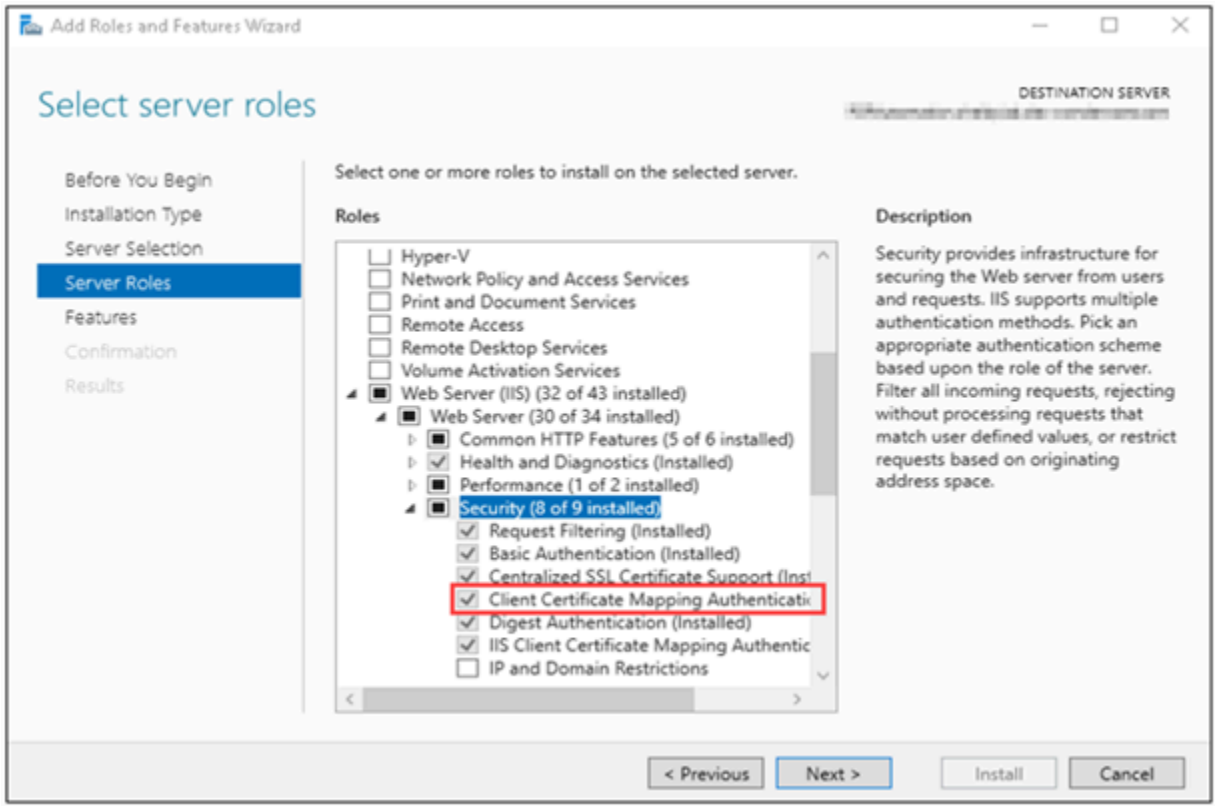
- Web API Server Configuration
- Web API Client Configuration

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Web API Server Configuration

Enable Client Certificate Mapping Authentication

- In Windows features, select **Server Roles**, and enable **Client Certificate Mapping Authentication** option.



Generate and Install Root Certificate

1. To generate the root certificate,
   a. Run the Command prompt and navigate to the *makecert* tool.
   b. Execute the below command:

      makecert -n "CN=<FQDN>" -r -sv ClientRootCA.pvk ClientRootCA.cer

      Replace *<FQDN>* with the fully qualified domain name of the machine (i.e. the computer name with the domain).

      The certificate is generated in the same path.

Note: The *makecert* utility is part of the Windows SDK.

2. To install the certificate,
   a. Run MMC on your machine. Then click **File** > **Add/Remove Snap-in.**
   b. Select **Certificate** and click **Add>**.
   c. Select **Computer account** option in the **Certificates snap-in** dialog box and click **Next**.
   d. Select **Local Computer** option in the **Select Computer** dialog box and click **Finish**.

   The generated certificate will be available at *Certificates (Local Computer) > Trusted Root Certification*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Authorities > Certificates

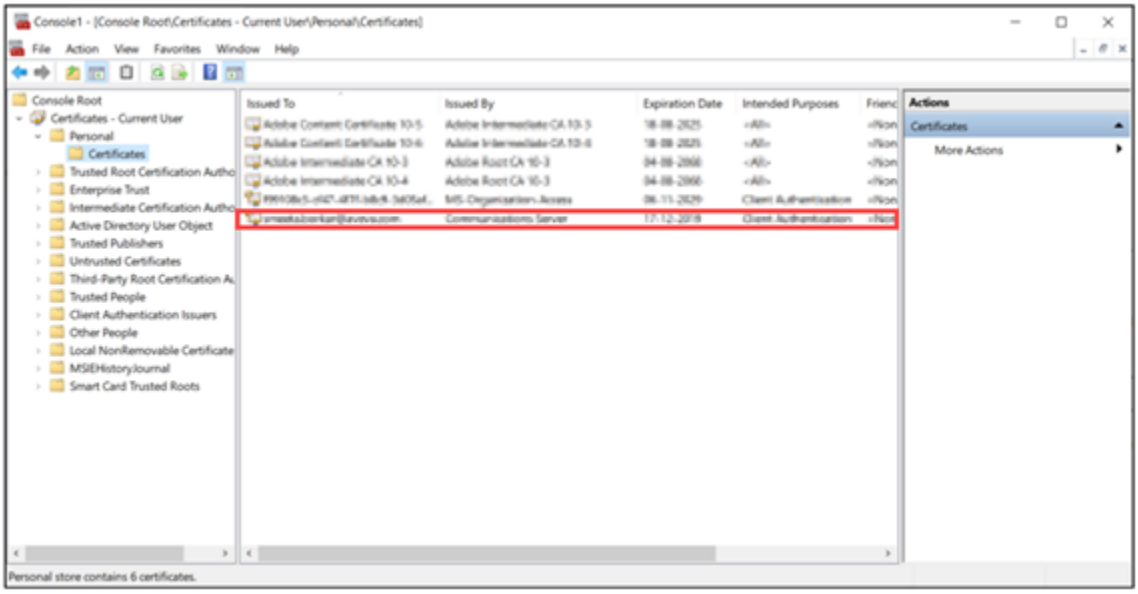
Generate and Install Client Certificate

1. To generate the client certificate,
   a. Run the Command prompt and navigate to the *makecert* tool
   b. Execute the below command:

      makecert -sk MyKeyName -iv ClientRootCA.pvk -n "CN=<username>" -ic ClientRootCA.cer -sr currentuser -ss my -sky signature -pe

      where, *<username>* is the current user name you have logged with.
2. To install the certificate,
   a. Run MMC on your machine. Then click **File** > **Add/Remove Snap-in.**
   b. Select **Certificate** and click **Add>**.
   c. Select **My user account** option in the **Certificates snap-in** dialog box.
   d. Click **Finish**.

   The generated certificate will be available at *Certificates (Current User) > Personal > Certificates*.



Export the Generated Certificate

To export the certificates for client authentication:

1. Navigate to Console Root\Certificates - Current User\Personal\Certificates.
2. Right-click on the generated certificate, click **All Tasks** and then click **Export**.
3. In the Certificate Export Wizard,
   - Select **Yes, export the private ke**y option to export certificate with the private key.
   - Select **No, do not export the private key** option if you do not want the private key.
4. To export the certificates with private key (.pfx certificate) for client authentication,
   a. In the Certificate Export Wizard, select **Yes, export the private ke**y option to export certificate with the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

private key.

b. Select the **Personal Information Exchange PKCS #12 (.PFX)** option. Select the **Include all certificates in the certification path if possible** and the **Export all extended properties** check boxes and then click **Next**.

c. Provide the password and click **Next**.

d. Specify the location and name for the file to be exported. Click **Next** and then click **Finish**.

The certificate with private key is available for client authentication.

5. To export the certificates with public key (.CER certificate) for client authentication,

a. In the Certificate Export Wizard, select **No, do not export the private key** option.

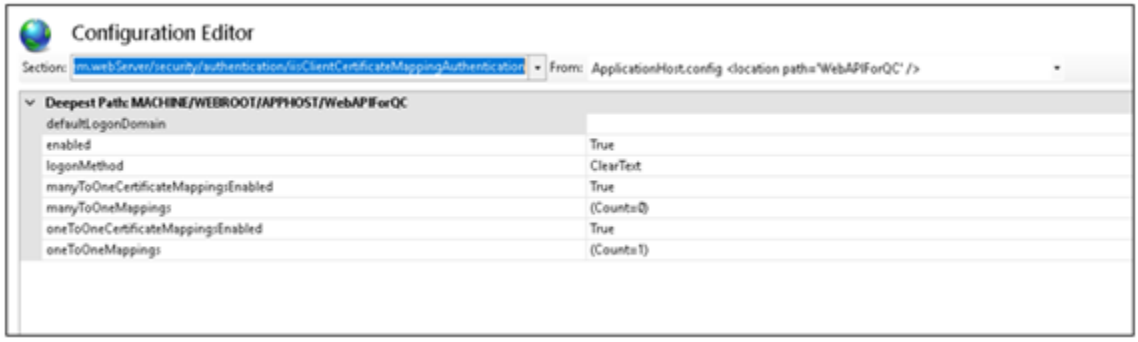
b. Select the **Base-64 encoded X.509 (.CER)** option and click **Next**.

c. Specify the location and name for the file to be exported. Click **Next** and then click **Finish**.

This certificate is used for IIS client certificate mapping authentication

Configuring IIS Mapping

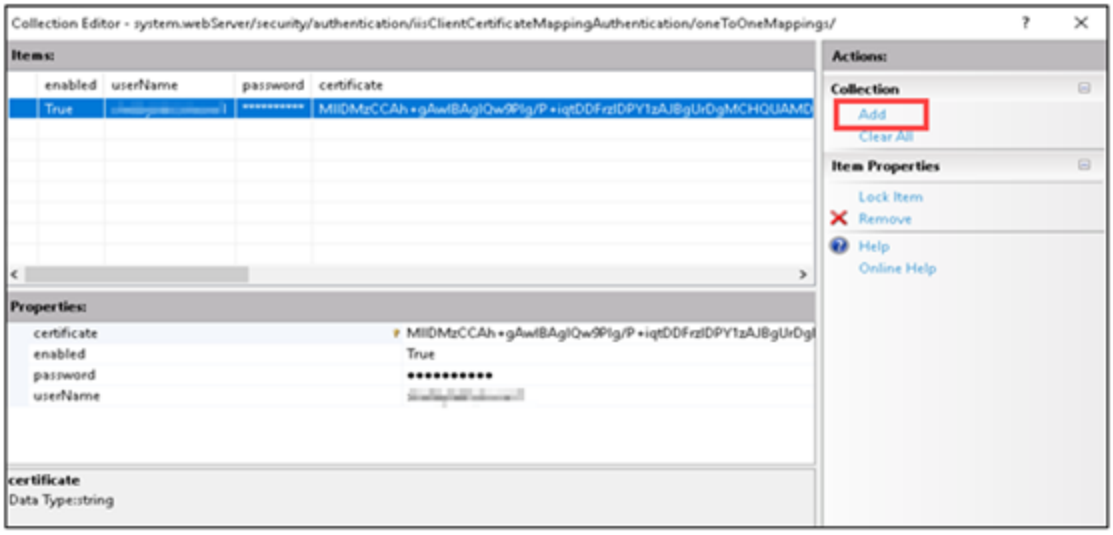
1. Open IIS to configure your site to map website to client certificate.
2. Click on your website and double click on Authentication.
3. Disable all authentication types.
4. Open Configuration Editor in Management Section.



- In the **Section**, enter *system.webServer/security/authentication/iisClientCertificateMappingAuthentication.*
- For the **enabled** property, enter **True**.
- For the **oneToOneMappingsEnabled** property, enter **True**.

5. Click the **oneToOneMappings** property.
6. In the Collection Editor dialog box, click **Add**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



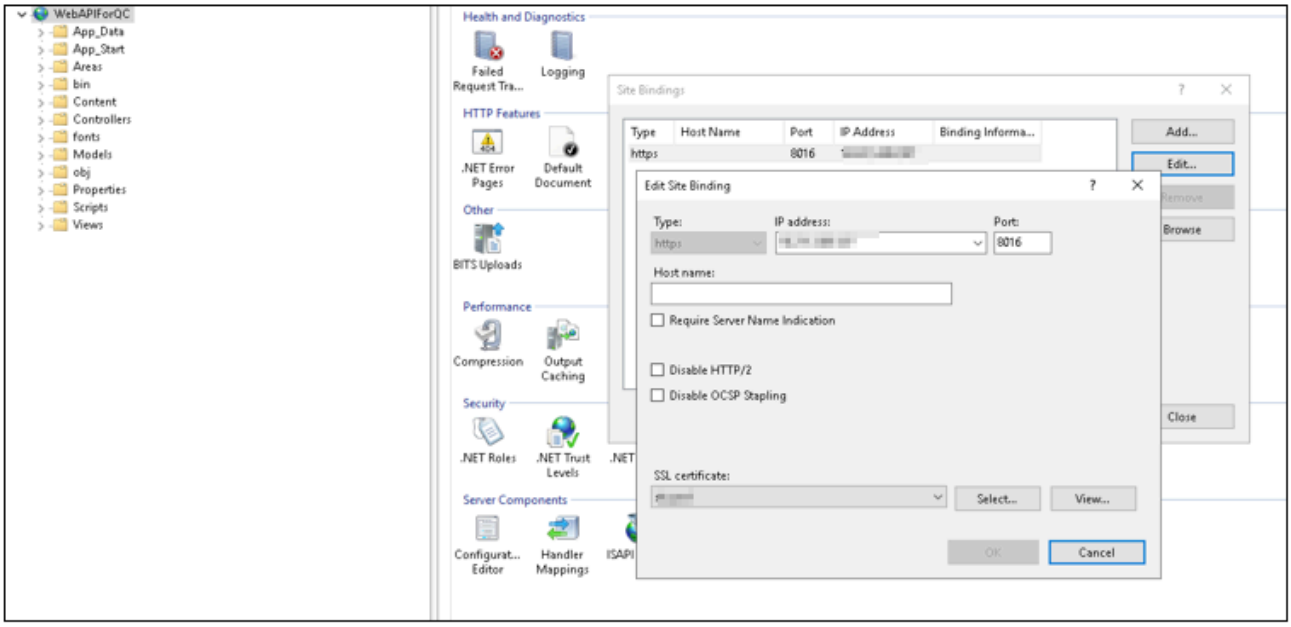
7. For the **certificate** property, copy and paste the public certificate that was exported (in step 5 of the section *Export the generated certificate*).

   Note: Open the Public certificate in the notepad and copy the encrypted certificate value in a single line.

8. For the **enabled** property, enter **True**.
9. Enter the valid Windows username and password which is used to login to the server machine.
10. Close the dialog box and click **Apply**.

    Mapping to your site is completed.

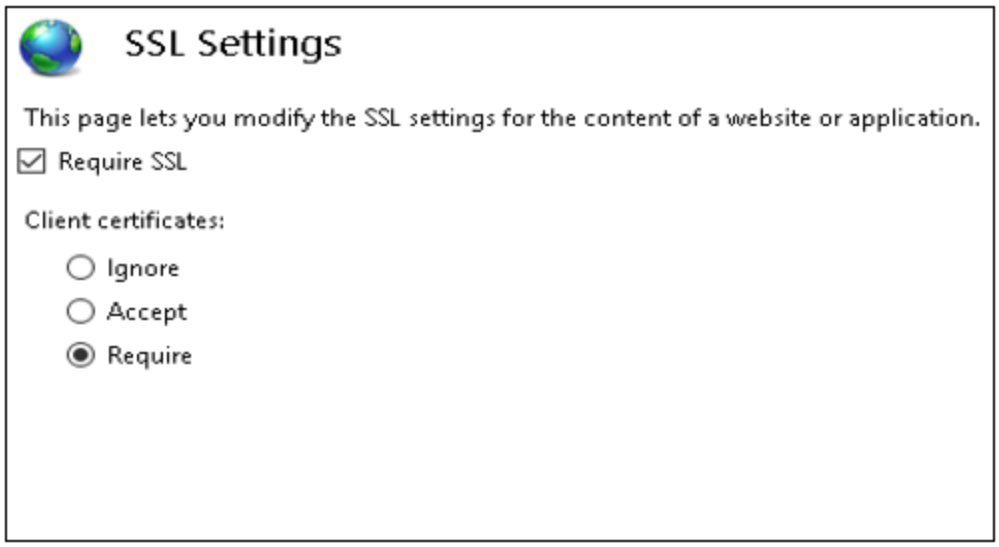
11. Import the certificate in IIS and edit **Site Bindings**.



- Under **Type** property, select **https**.
- Under **SSL certificate** property, select the certificate.

12. Click the **SSL Settings** and select **Require SSL** check box.

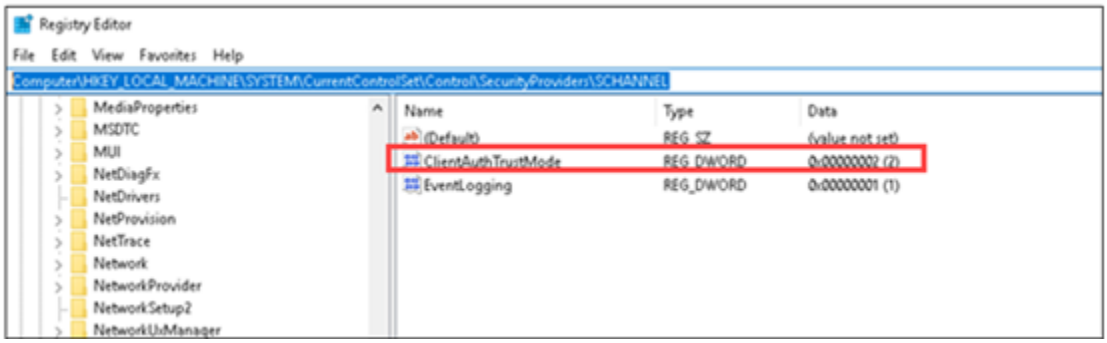
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Select the **Require SSL** checkbox and under the **Client certificates** section, select the **Require** option to make any client connection require a certificate to the site.

Modify Registry

- Modify/Add 'ClientAuthTrustMode' key in the Registry Editor under *Computer**HKEY_LOCAL_MACHINE**SYSTEM**CurrentControlSet**Control**SecurityProviders**SCHANNEL.*



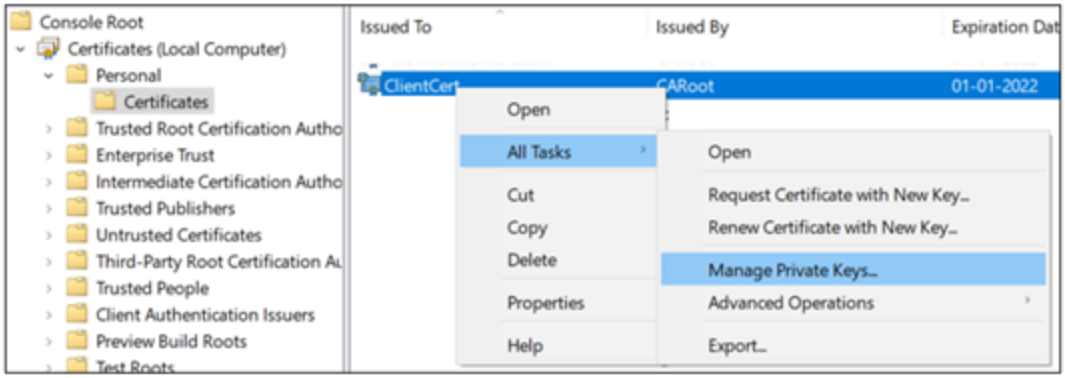
Add value as 2 for ClientAuthTrustMode Key.

Web API Client Configuration

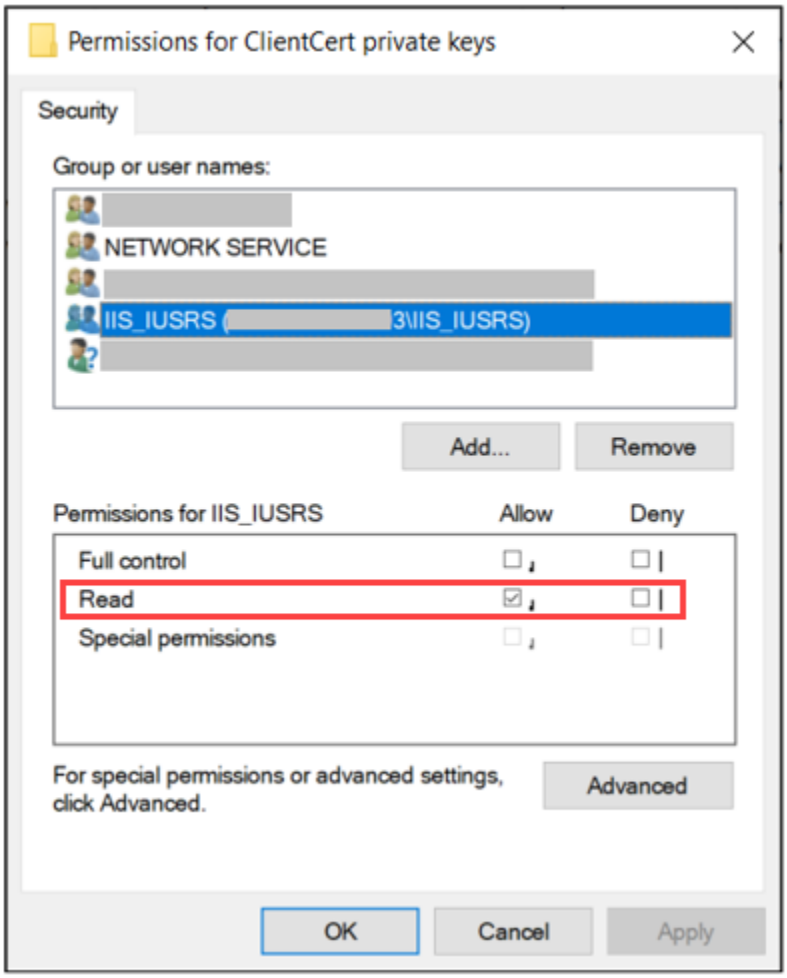
Install Certificate at Client Site

1. Install the exported certificate with private key under *Certificates(Local Computer)**Personal**Certificate*.
2. Right click on the installed certificate and select **All Tasks**>**Manage Private Keys** option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

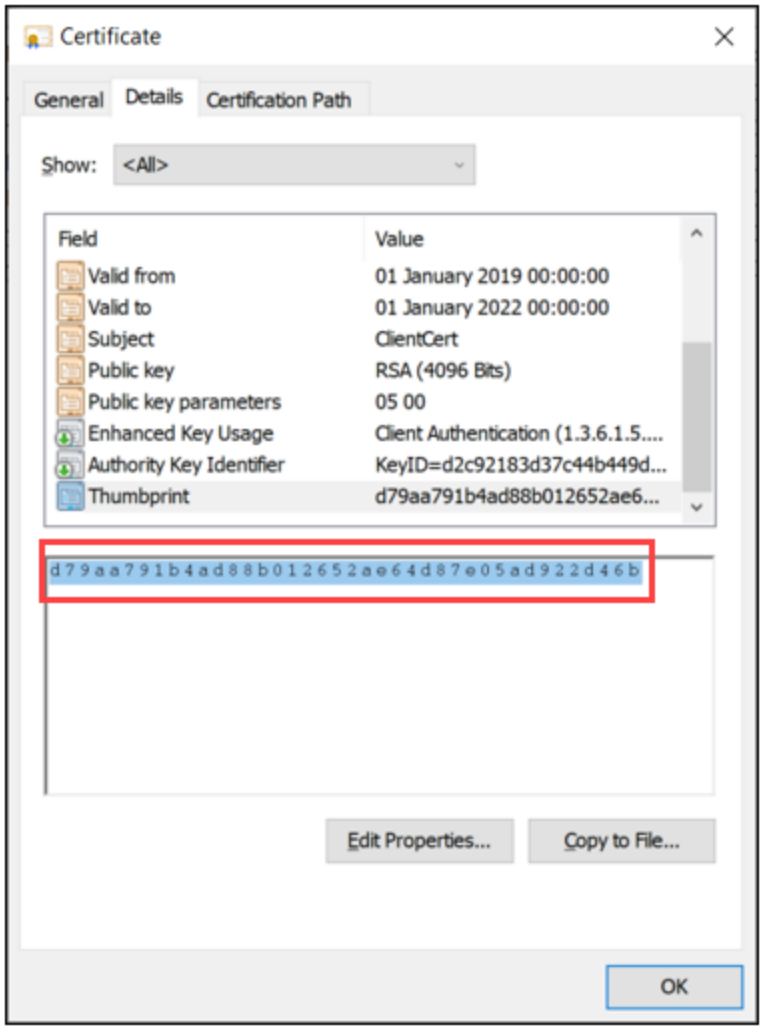


3. Add Read permissions to *NETWORK SERVICE* and <*MachineName*>*IIS_IUSERS*.



4. Open the certificate and copy the **Thumbprint** from the **Details** tab.
   This thumbnail is used to configure the Web API list item.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Web API Type files for Swagger/OpenAPI based Web APIs

You can add a Web API resource file in the *<Install Folder>\Resources\WebAPI*

For example: *C:\Program Files\AVEVA\Work Tasks\Resources\WebAPI*

There can be many files or a single file with multiple Web API types. These Web API types will get loaded in Web API List source dropdown. The existing Web APIs and the APIs which are not Swagger/OpenAPI based Web APIs are considered as Generic types.

The sample schema to add the Web API types is as given below:

```
<?xml version="1.0" encoding="utf-8"?>
<WebAPITypes>
<WebAPIType Name="TestOpenAPINoAuth">
<SwaggerConfiguration><![CDATA[http://localhost:63452/swagger/DotNetCoreWebAPIForTestV1/
swagger.json]]></SwaggerConfiguration>
<BaseAddress><![CDATA[http://localhost:63452/]]></BaseAddress>
<Authentication Mode="No Authentication"/>
</WebAPIType>
<WebAPIType Name="TestOpenAPI301Spec">
<SwaggerConfiguration><![CDATA[http://localhost:8040/swagger/DotNetCoreWebAPIForTestV1/
swagger.json]]></SwaggerConfiguration>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<BaseAddress><![CDATA[http://localhost:8040/]]></BaseAddress>
<Authentication Mode="Windows" >
<ImpersonationUser><![CDATA[domain\username]]></ImpersonationUser>
</Authentication>
</WebAPIType>
<WebAPIType Name="TokenAccessAPI">
<SwaggerConfiguration><![CDATA[http://localhost:8032/swagger/V1/
swagger.json]]></SwaggerConfiguration>
<BaseAddress><![CDATA[http://localhost:8032]]></BaseAddress>
<Authentication Mode="Access Token"/>
</WebAPIType>
<WebAPIType Name="TestOpenAPI20Spec">
<SwaggerConfiguration><![CDATA[http://localhost:8010/swagger/DotNetCoreWebAPIForTestV1/
swagger.json]]></SwaggerConfiguration>
<BaseAddress><![CDATA[http://localhost:8010/]]></BaseAddress>
<Authentication Mode="Basic Authentication">
<UserName><![CDATA[domain\username]]></UserName>
</Authentication>
</WebAPIType>
<WebAPIType Name="Certificate Auth">
<SwaggerConfiguration><![CDATA[http://localhost:8010/swagger/DotNetCoreWebAPIForTestV1/
swagger.json]]></SwaggerConfiguration>
<BaseAddress><![CDATA[http://localhost:8010]]></BaseAddress>
<Authentication Mode="Certificate" >
<Location><![CDATA[LocalMachine]]></Location>
<Thumbprint><![CDATA[]]></Thumbprint>
</Authentication>
</WebAPIType>
</WebAPITypes>
```

Web API Source

Web API source is used to define the details of the OpenAPI which can be then fetched through the **Source** dropdown of Web API List page. Use Web API Source to customize your business applications.

To add a Web API Source

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Web API Source**.

   The **Web API Source** page appears.
2. Click **New**.

   The **Web API Source List Item** wizard appears.
3. In the **Name** box, type a name for the Web API source.
4. In the **Description** box, type a description for the Web API source.
5. In the **Swagger Url** box, type the swagger url of the Web API that was created and configured using swagger.
6. In the **Base Address** box, type a URL in the format: **http://localhost/MySite/**.

   You can also enter an IPv6 URL for the **Base Address**. The IPv6 address must be enclosed within square brackets.
7. Select one of the following options for **Authentication Mode**:
   - **No Authentication**: If the Web API source does not require authentication.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Windows**: If the Web API source is Windows authenticated. Provide the impersonation user.
- **Access Token**: If the Web API source requires ArchestrA Security Token Service (ASTS) access token authentication.
- **Basic Authentication**: If the Web API source requires basic authentication. Provide the user name with domain and the password.
- **Certificate**: If the Web API source requires certificate authentication. Provide the certificate store location and the thumbprint. For more information on certificate-based authentication, see Web API Certificate-based Authentication.

  Note: The thumbprint should not contain any spaces. In some machines, the characters in thumbprint should be in upper case.

8. Click **Save**. The Web API source is added.

Note: Once the Web API Source gets consumed by Web API, you can change the authentication mode and other values and save it. But when you again select the Web API Source from the drop-down, then the values from the Web API Source are populated, and the previously saved values are lost.

To sync the OpenAPI details

You can sync the OpenAPI details in the XML file located in the Resource folder in the installation path and save it in the database. Any modifications made in the XML file will be reflected in the Web API Source page when you sync.

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Web API Source**.

   The **Web API Source** page appears.
2. Click **Sync**.

   The OpenAPI details in the XML file located in the **Resource** folder in the installation path is synced and will get displayed in the **Web API Source** page.

Import a New Form using an API

Use the following API to import a new Form :

```
                try
    {
// Specify the form Id
        var formId = "C0084A77-0BAE-4237-9D75-1FA09B59E82B";
        // Specify the form name
        var formName = "New";
        // specify the form version
        var formVersion = "1";
        // Specify the application/repository name
        var applicationName = "PSI12Sprint05Repo";
        // Forms List Name
        var formListName = "Forms List";

        BaseForm newDefinition;
        BaseForm currentFormDefinition;
        Skelta.Entity.UserContext userContext = new Skelta.Entity.UserContext(new
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Guid("B6FD1625-D203-4DEC-933D-516789E90E45"), new
Skelta.Core.ApplicationObject(applicationName), "skeltalist");

            Skelta.Forms.Web.CommonFunctions.ValidateCachedForm(formId, formVersion, out
currentFormDefinition, formListName);
            XmlDocument xmlDoc = new XmlDocument();
    // Specify form definition xml file path
            xmlDoc.Load(@"C:\PSI12Sprint05Repo_New_Form.xml");
            var newFormDefinitionString = xmlDoc.InnerXml;
            newDefinition = BaseForm.LoadDefinitionXml(newFormDefinitionString);
            newDefinition =
Workflow.NET.FormsUtility.UpdateNextGenFormDefinition(newDefinition);
            newDefinition.TopLevelForm.FormTitle = formName;
            Skelta.Forms.Web.CommonFunctions.UpdateDefinitionToDBAndValidateCache(formId,
formVersion, newDefinition, formListName);
         }
         catch(Exception ex)
         {
         }
```

Note:
- The code imports the specified form definition without performing validations, for example, checking product or build versions.
- If the the form definition contains an invoke workflow control, then the API imports the form definition but does not create workflow association entries. You have to create them manually.
- If a form definition contains an invoke workflow control, and if you import the invoke workflow control from either the context menu or the Forms Designer, then the workflow association entries are created.

Scenarios

This section demonstrates some scenarios and examples on using Forms:

- Scenario 1: Purchase Order Form
- Scenario 2: Vehicle Registration Form
- Scenario 3: EMI Form
- Examples

Scenario 1: Purchase Order Form

Purchase order form scenario shows how you can use a Base Form container to create and repeat line items, code scripts to automatically calculate values, and update a value on data change event.

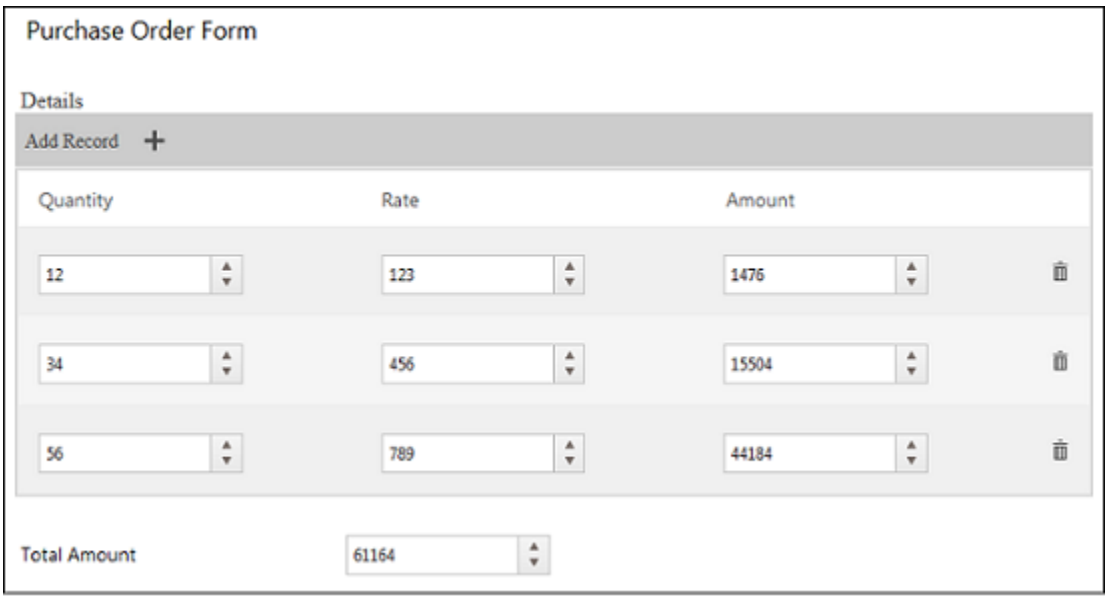
Requirements

A purchase order form which:

- captures information about quantity and rate as a line item.

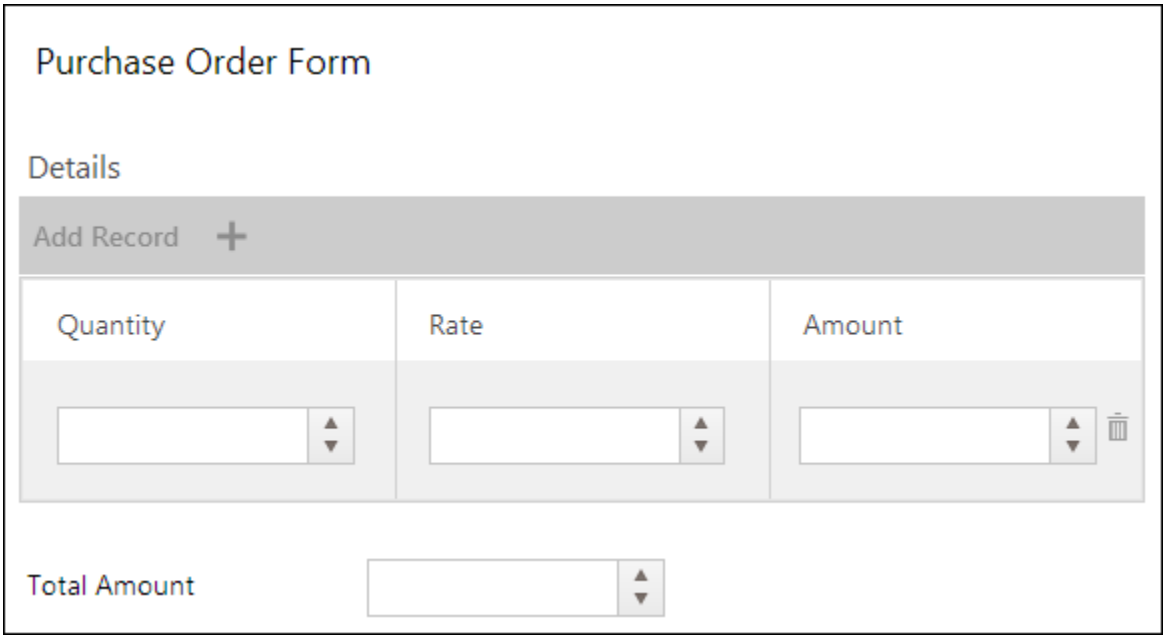
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- automatically calculates the amount by multiplying quantity and rate and displays it as a line item.
- starts with one line item and provides an option to add as many line items as required.
- automatically calculates the total amount by adding each line item amount.



Design

1. Create a form.
2. Add a **Base Form** container on the form and set the **Display Mode** to **Grid**, to create line items.
3. Add three **Number** controls on the **Base Form** container to capture quantity and rate and display the amount, as a line item.
4. Add a **Number** control on the form to display the total amount.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scripts

1. Code the script as follows for the **Value** property of the **Amount** control to automatically calculate the amount:

   ```
   qty=control.findByXmlNode("Quantity").value;
   rate=control.findByXmlNode("Rate").value;
   return qty*rate;
   ```

2. Code the script as follows for the On Data Change property of the Amount control to automatically calculate the total amount:

   ```
   control.findByXmlNode("TotalAmount").value= control.sumByXmlNode("Amount");
   ```

Note: The **sumByXmlNode()** method adds the value of each instance of the specified control.

Scenario 2: Vehicle Registration Form

Vehicle registration form scenario shows how you can show or hide Panel Form containers to display relevant information based on a selection.

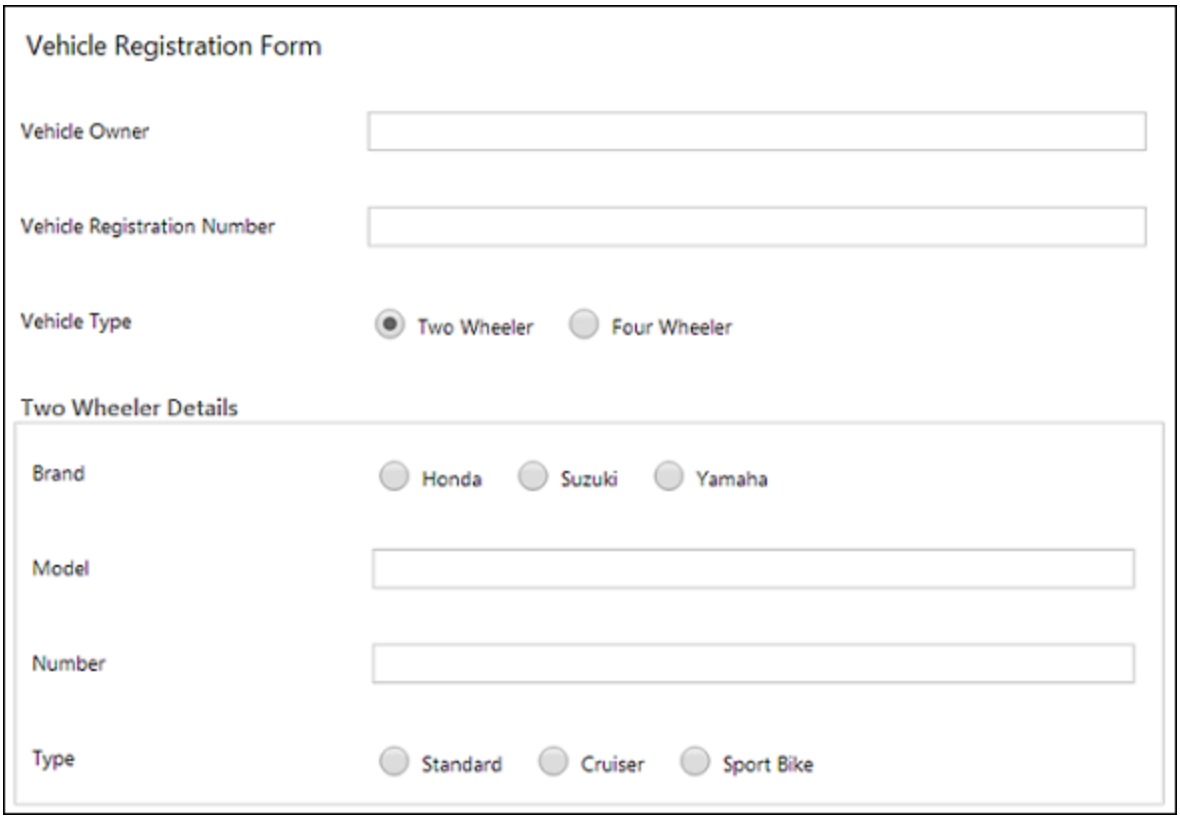
Requirements

A vehicle registration form which:

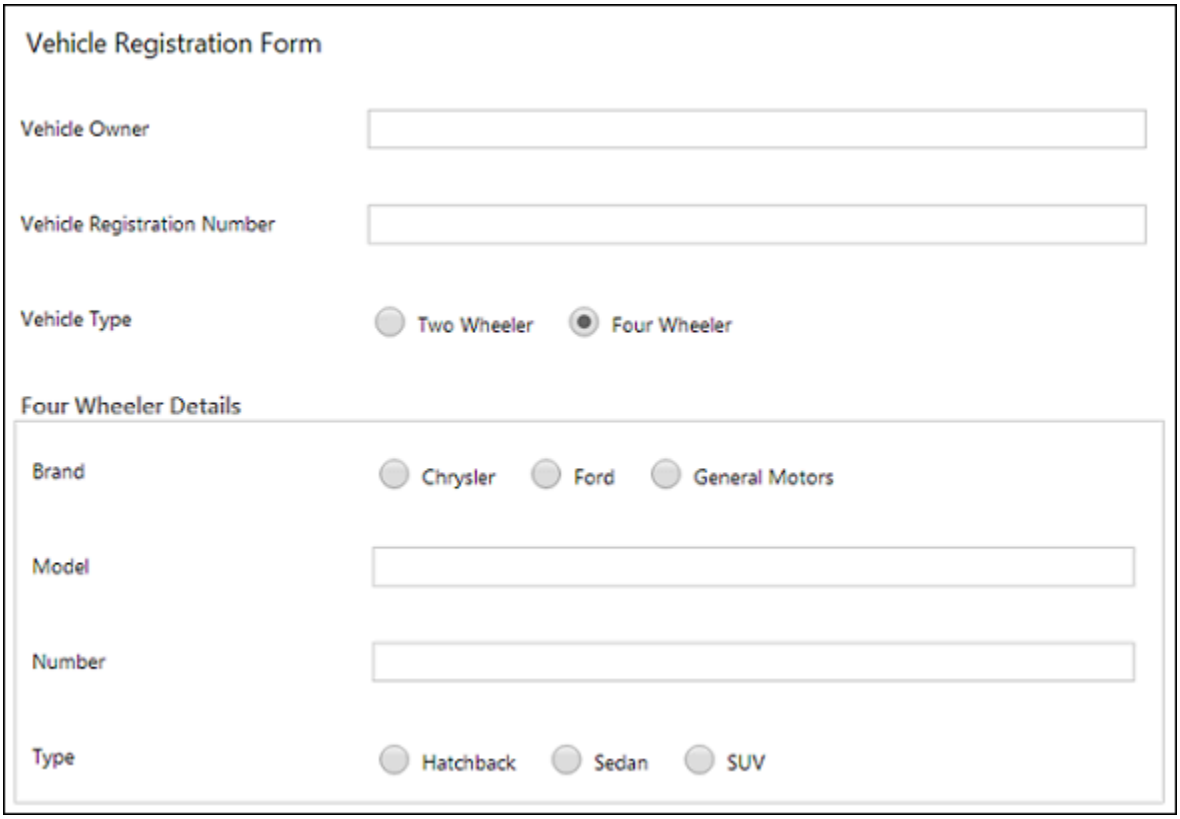
- captures information about a vehicle owner, vehicle registration number, and vehicle type.
- displays the two wheeler additional information by default.
- automatically displays the relevant additional information to capture based on the selected vehicle type.
- captures additional information about the selected vehicle type such as brand, model, number, and type.

The following figure represents the Vehicle Registration Form for entering information about Two Wheelers.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



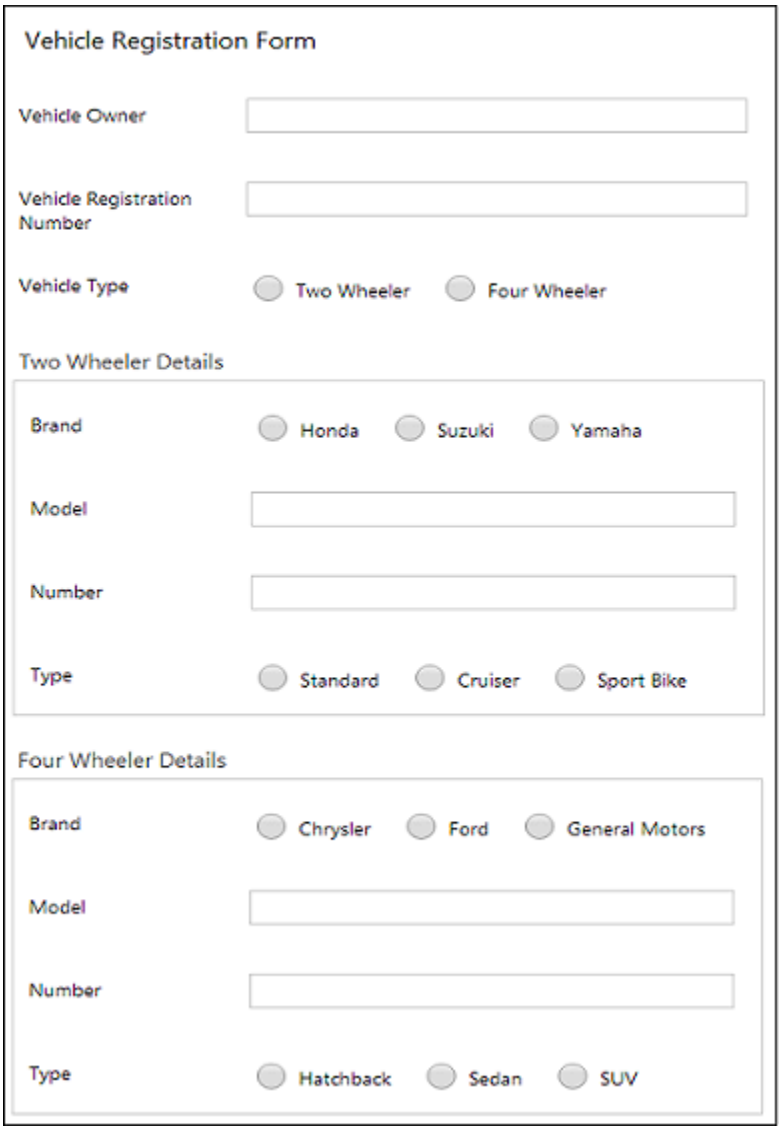
The following figure represents the Vehicle Registration Form for entering information about Four Wheelers.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Design

1. Create a form.
2. Add two **Text** controls on the form to capture vehicle owner and vehicle registration number.
3. Add a **Radio Button** control on the form to select the vehicle type and set the **Default Value** to the value of the two wheeler option.
4. Add two **Panel Form** containers on the form to capture additional information about the selected vehicle type.
5. Add two **Text** controls on each of the **Panel Form** to capture model and number.
6. Add two **Radio Button** controls on each of the **Panel Form** to capture brand and type.
7. To hide the Four Wheeler Panel Form, set the value of the **Visible** property to **No** in the **Appearance** tab.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scripts

Code the script as follows for the **On Data Change** property of the **Vehicle Type** control to automatically displays the relevant additional information based on the selected vehicle type:

```
if( currentValue == "2")
 {
 control.findById('P1').visible = true;
 control.findById('P2').visible = false;
 }
else
 {
 control.findById('P1').visible = false;
 control.findById('P2').visible = true;
 }
```

Vehicle Type control has the value 2 and 4 set for the **Two Wheeler** and **Four Wheeler** options respectively. **P1** and **P2** are the IDs for the **Two Wheeler Details** and **Four Wheeler Details** containers respectively.

Scenario 3: EMI Form

EMI (Equated Monthly Installment) form scenario shows how you can use scripts to automatically add line items on a Base Form container on data change event.

Requirements

An EMI form which:

- captures information about principal, rate, and tenure.
- automatically calculates the EMI amount, interest amount, principal reduction, and balance due using the EMI formula and displays it as a line item for each installment number.

EMI Formula

EMI = (P*r) (1+r)^n / (1+r)^n - 1

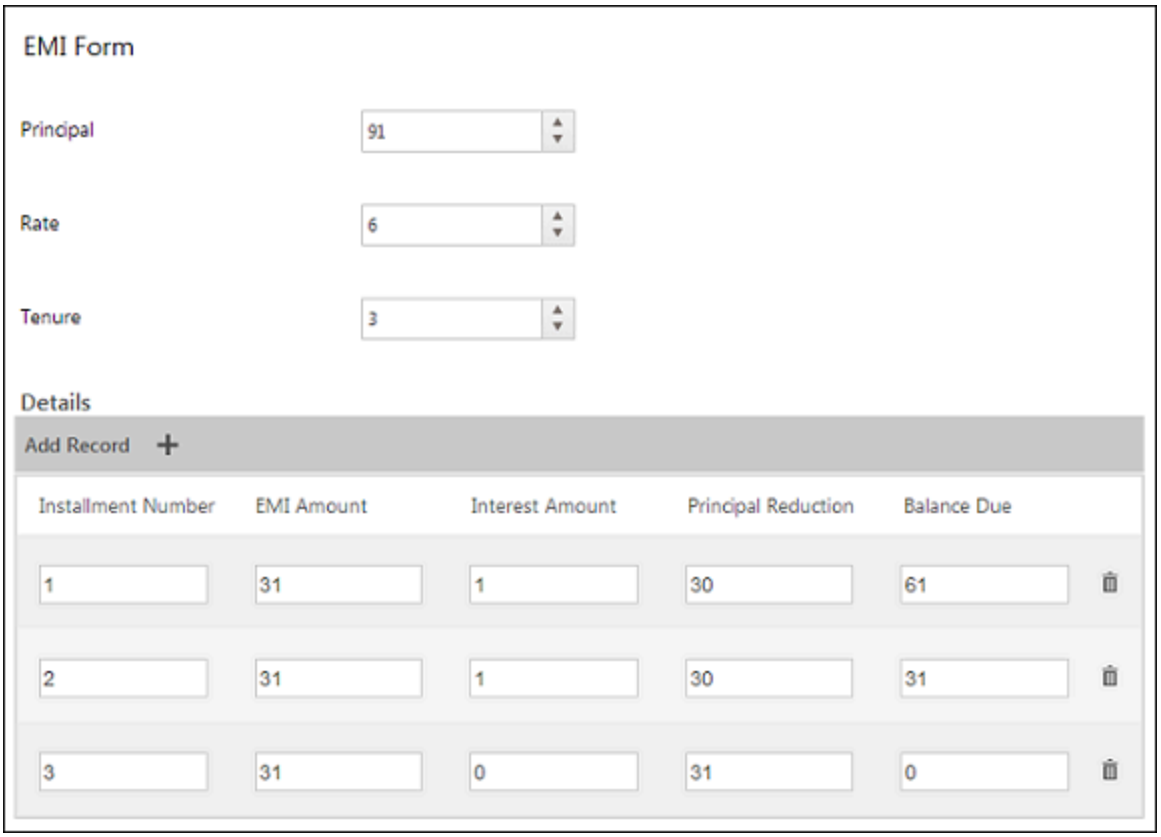
where:

P = Principal (amount of loan).

r = Rate of interest per installment period.

n = Number of installments in the tenure.

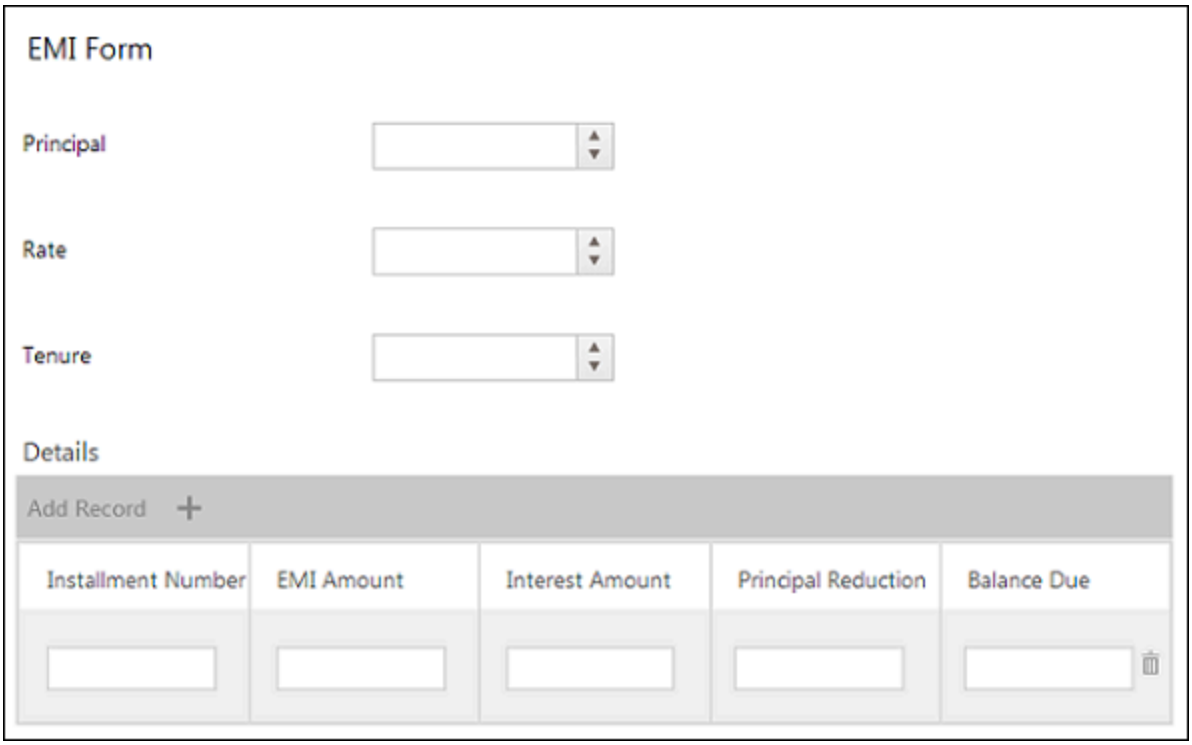
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Design

1. Create a form.
2. Add three **Number** controls on the form to capture principal, rate, and tenure.
3. Add a **Base Form** container on the form and set the **Display Mode** to **Grid**, to create line items.
4. Add five **Text** controls on the **Base Form** container to display the installment number, EMI amount, interest amount, principal reduction, and balance due, as a line item.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Scripts

Code the script as follows for the **On Data Change** property of the **Tenure** control to automatically calculate the **EMI** and display the line items:

```
var tenure =control.findByXmlNode("Tenure").value;
var interest = control.findByXmlNode("Rate").value;
var principal =control.findByXmlNode("Principal").value;
if (principal === 0 || interest === 0 || tenure === 0)
{
return;
}
var grid=control.findByXmlNode("Details");
var principal1= parseFloat(principal);
var interest1 = parseFloat(interest );
var tenure1 = parseFloat(tenure );
var ipm =parseFloat(parseFloat(interest1) / parseFloat("1200.0"));
var EMId =((principal1 * ipm) * parseFloat(Math.pow(parseFloat(ipm + 1), tenure1))) /
parseFloat(parseFloat((Math.pow(parseFloat(ipm+1), tenure1))) - 1);
var EMI =Math.round(EMId);
var count=grid.records().length;
for (i=count; i>0; i--)
{
grid.removeAt(i);
}
for (j=0; j<tenure; j++)
{
var newRow=grid.addRecordToBottom();
newRow.findByXmlNode("InstallmentNumber").value = j+1;
if (j===0)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
newRow.findByXmlNode("PrincipalReduction").value = Math.round((EMI-(principal/100)));
newRow.findByXmlNode("EMIAmount").value =EMI;
newRow.findByXmlNode("InterestAmount").value = Math.round((principal)/100);
newRow.findByXmlNode("BalanceDue").value = Math.round(principal - (EMI-(principal/100)));
}
else
{
var newprincipal=grid.records()[j-1].findByXmlNode("BalanceDue").value;
newRow.findByXmlNode("EMIAmount").value =EMI;
newRow.findByXmlNode("PrincipalReduction").value =Math.round((EMI-(newprincipal/100)));
newRow.findByXmlNode("InterestAmount").value =Math.round((newprincipal)/100);
newRow.findByXmlNode("BalanceDue").value = Math.round(newprincipal - (EMI-(newprincipal/
100)));
}
}
```

Examples

This section demonstrates the examples for using Forms features.

Accessing Values of a Selected Row Through Other Controls

You can access the column values for a selected row of a data grid through other controls.

Data Grid

In the following example, we create a lookup named **Employee**, with a column named **Salary** that holds the salaries of all the employees. We create a form with **Data Grid** control and configure the lookup to the **Data Grid** control. We also add a **Button** and **Text** control to the form. Let us now consider that we want to fetch the salary of a selected employee .We can access this value through the **Text** control in the form as follows:

1. Create a lookup and name it as **Employee**.
2. Create a G**rid Configuration** using the **Employee** lookup.
3. Create a form and add a **Data Grid** control.
4. Modify properties of the **Data Grid** control as follows:
   a. In the **Basic** tab, set the **Configuration Name** property to the grid configuration created in Step 2.
5. Add a **Button** and **Text** control to the form.
6. Set the **Button Type** property of the **Button** control to **Custom**.
7. Code the script for the **On Click** event of the **Button** control as follows:

   ```
   var gridSelecedValue=control.findByXmlNode("Grid1").selectedValue;
   control.findByXmlNode("Textinput1").value=gridSelecedValue.<ColumnName>;
   ```

The script uses the `selectedValue` property to fetch the column value of selected row. When you click the button, the **Text** control gets updated with the Salary value (column), of the selected Employee.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Hierarchical Grid

In the following example, we create a hierarchical data grid with lookups named **Work Order** (parent lookup), and **Jobs** (child lookup). We create a form and consume this grid configuration in the form. We also add a **Button** and **Text** control to the form. Let us now consider that we want to fetch the **Job_Id** of a selected Job. We can access this value through the **Text** control in the form as follows:

1. Create a Hierarchical Data Grid using the lookups **Work Order** (parent lookup) and **Jobs** (child lookup). For more information on creating a Hierarchical Data Grid, see the topic Create a Hierarchical Data Grid.
2. Create a form and consume the grid configuration created in step 1.
3. Add a **Button** and **Text** control to the form.
4. Set the **Button Type** property of the **Button** control to **Custom**.
5. Code the script for the **On Click** event of the **Button** control as follows:

```
var gridSelecedValue =control.findByXmlNode("Grid1").selectedValue;
control.findByXmlNode("Textinput1").value=gridSelecedValue.<ParentLookupName>.<ChildLookupName>
```

The script uses the `selectedValue` property to fetch the column value of selected row. When you click the button, the **Text** control gets updated with the Job_Id (column), of the selected Job.

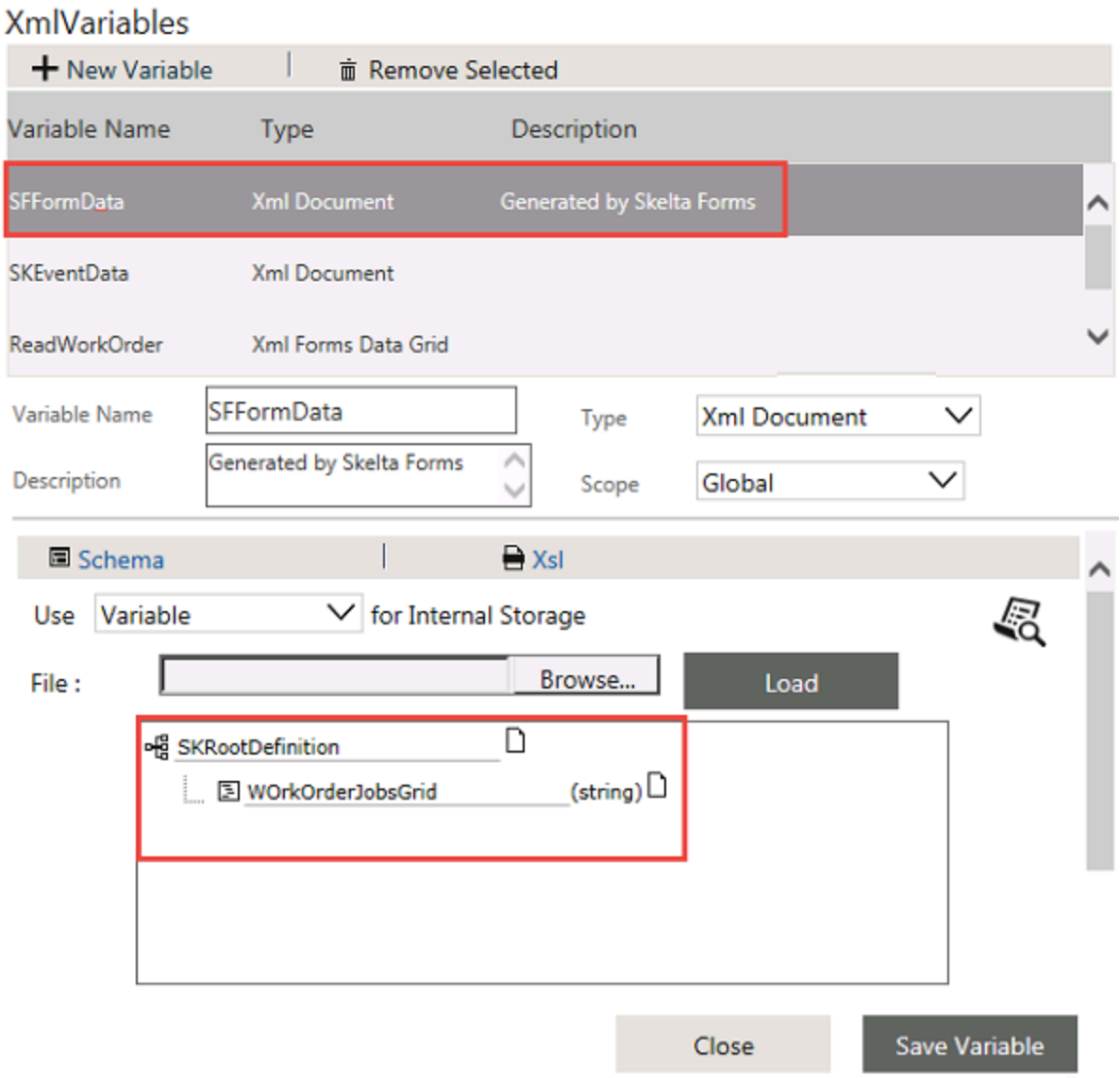
Add Data Grid to Workflows

You can add Data Grid to workflows. The Data Grid values can be read and modified within workflows.

In this example, we create a workflow with a For-Each loop activity. The user selects a Job in the **WorkOrderJobsGrid** and clicks **Finish**. The For-Each Loop activity iterates through the **WorkOrderJobsGrid**. An information mail is sent to the actor with details of Work Order Id, Operation ID and the Sequence number of the selected Job.

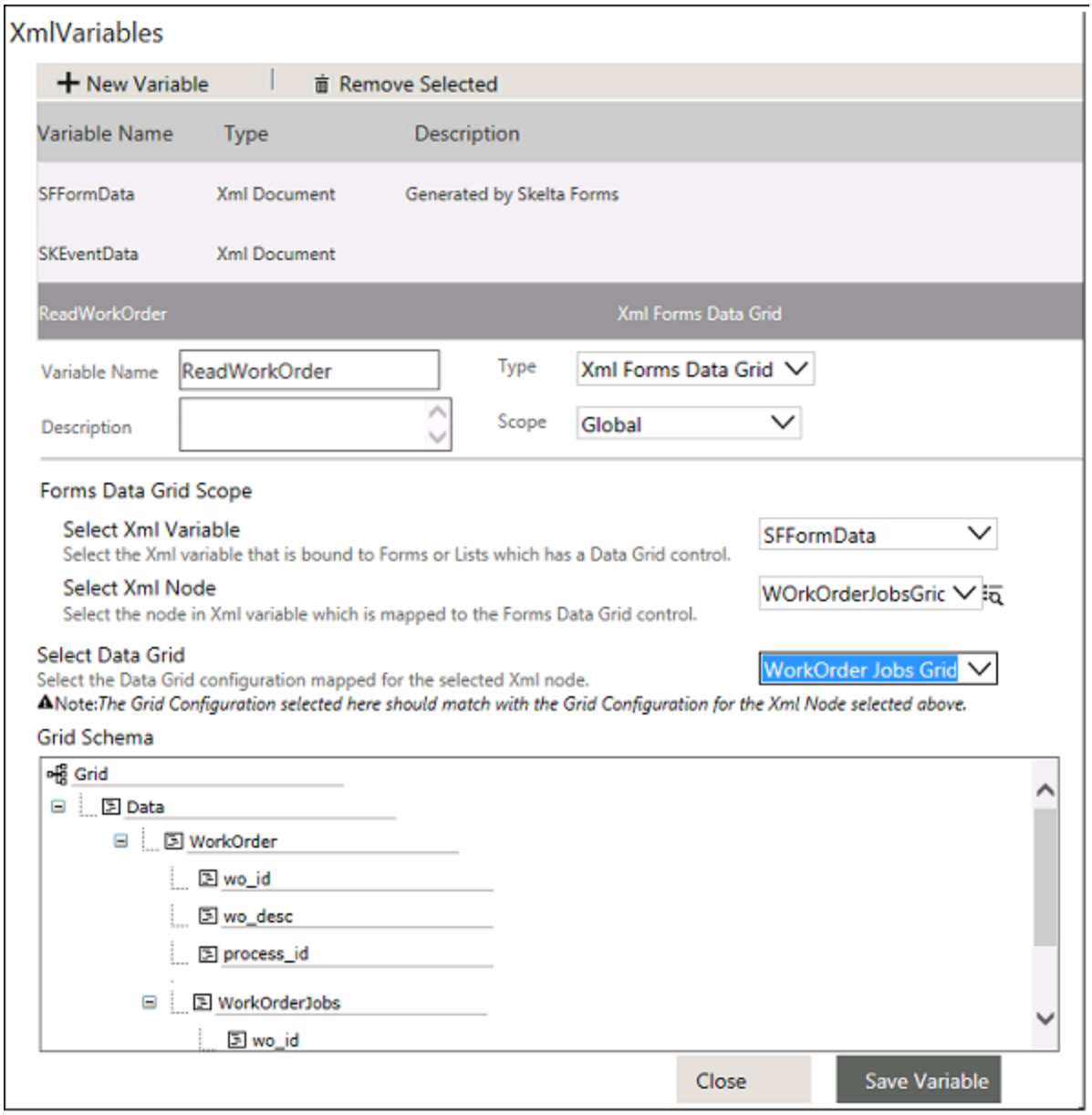
1. Create a form with a two level grid configuration, containing two lookups, Work Order and Jobs. Name the form as **WorkOrderJobs** scenario.
2. Associate the form with a workflow and name the workflow as **WorkOrder Jobs WF**.
3. Design the workflow. The **SFFormData** variable is created for the associated **WorkOrderJobs** scenario form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



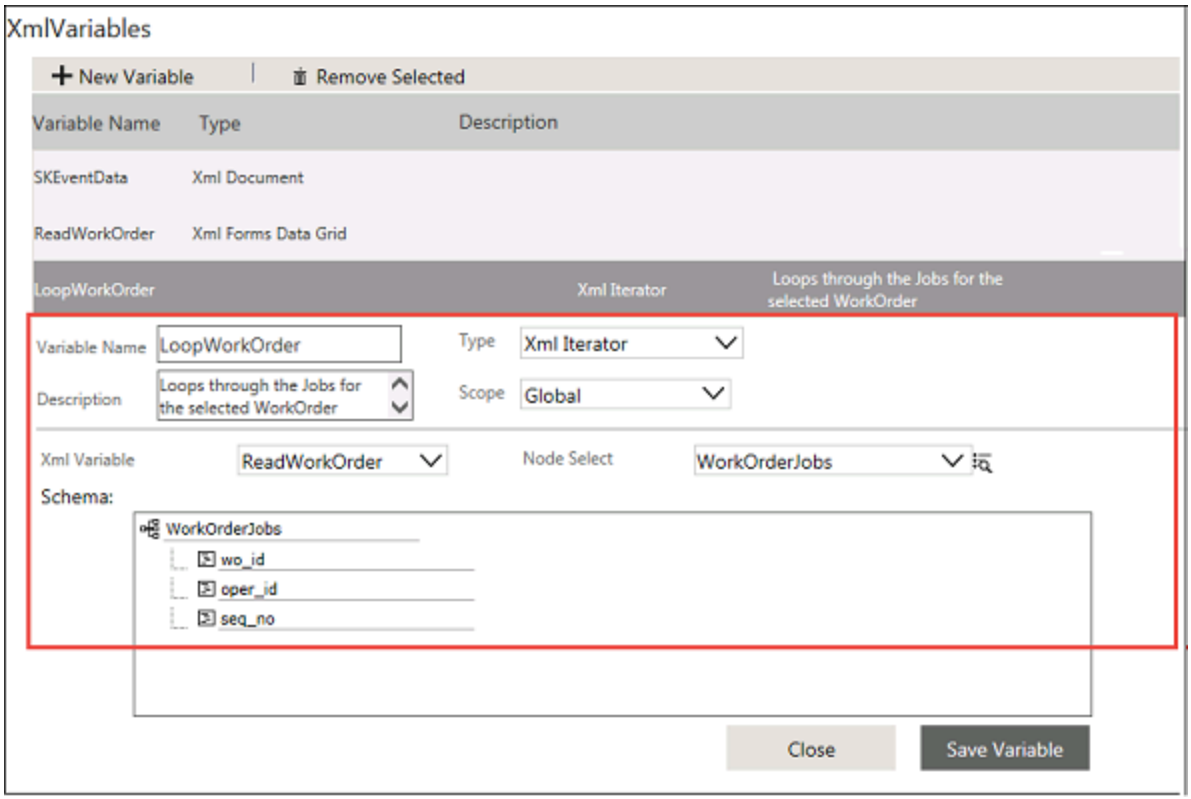
4. **WorkOrderJobsGrid** is the mapped XML node for the Work Order and Jobs grid.
5. The grid selection information is stored in these nodes. To extract the required values and use it within workflows, we need to create XML variables of type XML Forms Data Grid.
6. Create a new XML variable of type XML Forms Data Grid for **Work Order** and **Jobs** Grid.
7. Select **WorkOrderJobsGrid** XML Node which is mapped to the Work Order and Jobs grid. This contains the complete selection information of the Data Grid.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



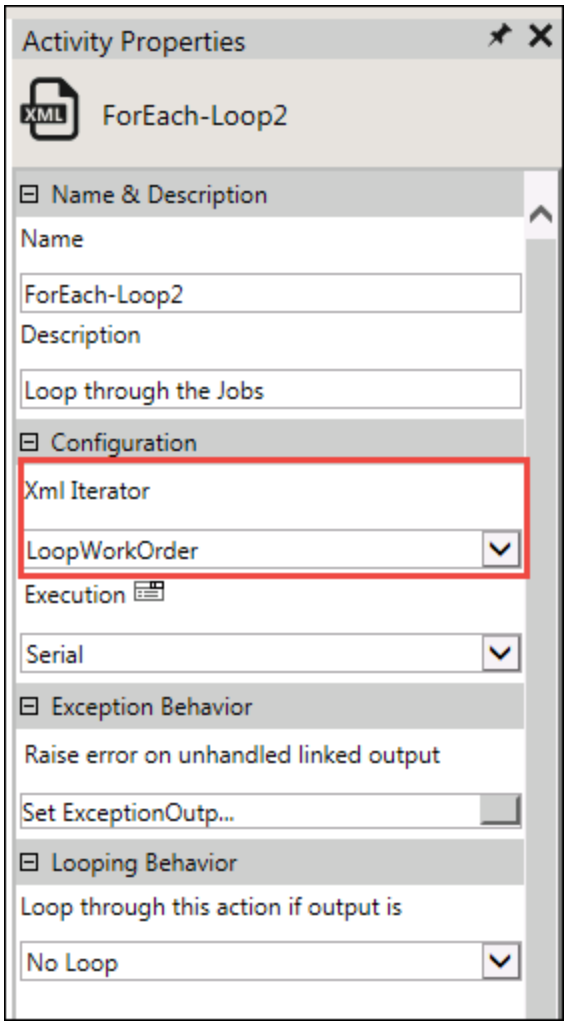
8. We need an XML Iterator variable to loop through the Jobs for the selected Work Order. Create an XML Iterator variable **LoopWorkOrder** as shown in the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

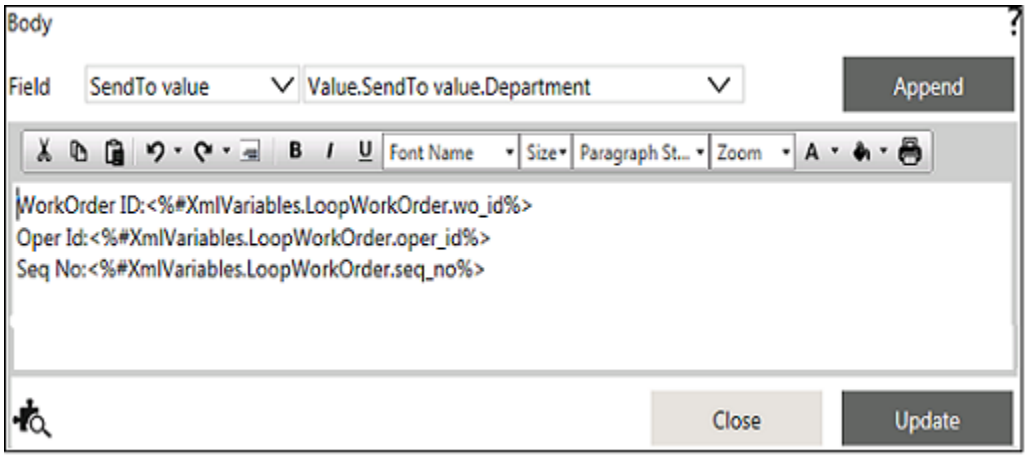


9. Add a For-Each Loop activity. Set its Xml Iterator property with Iterator variable created above.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

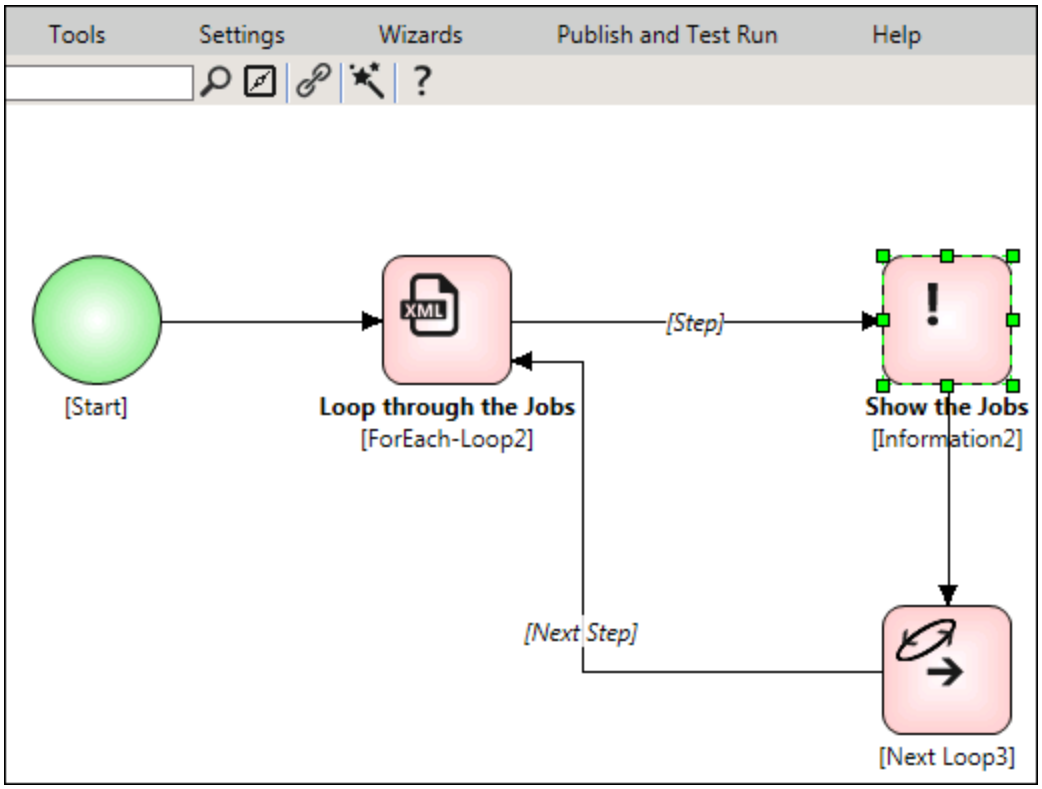


10. Add an Information activity. Set its **Body** property to show the Work Order ID, Oper ID, and the Seq No for the selected Job.

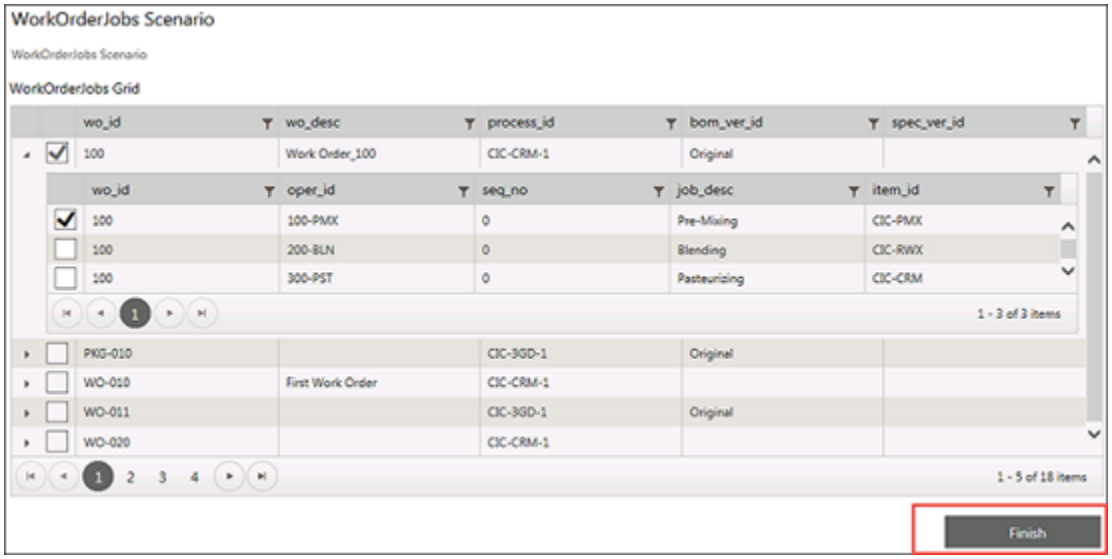


11. Publish and Test Run the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



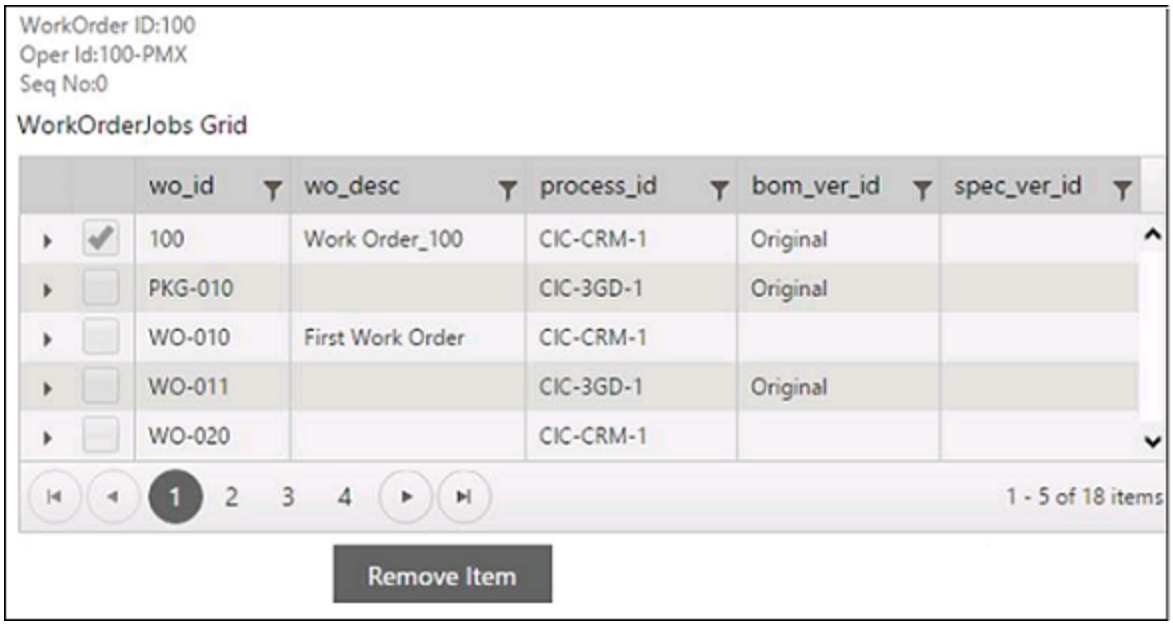
12. Select rows from **Work Order Jobs Grid** in the form and then click **Finish**.



13. Information is sent with the details of the selected job.

Workflow	Subject	Activity Display Name	Priority	Response By	Assigned On
WorkOrder Jobs WF	100 100-PMX 0	Information2	●		12/8/2015 5:00:02 PM
WorkOrder Jobs WF	100 0 100-PMX	Information2	●		12/8/2015 4:58:31 PM
WorkOrder Jobs WF	100-PMX 0	Information2	●		12/8/2015 3:58:19 PM

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Create a Hierarchical Data Grid

You can create a multilevel hierarchical data grid to map multiple data sources.

In the following example we create one non parameterized lookup (Work Order) and three parameterized lookups (**Jobs**, **Job Steps**, and **Job Steps Data**). Then, we create a four level hierarchical data grid using the lookups and map all the parameters.Then, design a form and use this data grid in the form using the **Data Grid** control.

1. Create the following four lookups:
   a. **Work Order** -Non-parameterized
   b. **Jobs** - Parameterized on the Work Order lookup(**wo_id**)
   c. **Job Steps** -Parameterized on the **Jobs** lookup (**wo_id**, **oper_id**, **seq_no**)
   d. **Job Steps Data** - Parameterized on the **Job Steps** lookup (**wo_id**, **oper_id**, **seq_no**, **step_no**)
2. Create a four level, hierarchical grid configuration and name it as **MES Configuration**.
3. Select the data sources in the same order as the lookups created in step 1 .
4. Click **Parameter Mapping**.
   The parameter mapping screen appears. Map the child parameters to the parent column.
5. Save the parameter mapping and the grid configuration.
6. Design a form and consume the grid configuration, **MES Configuration** in the **Data Grid** control.
7. Click **Preview**.

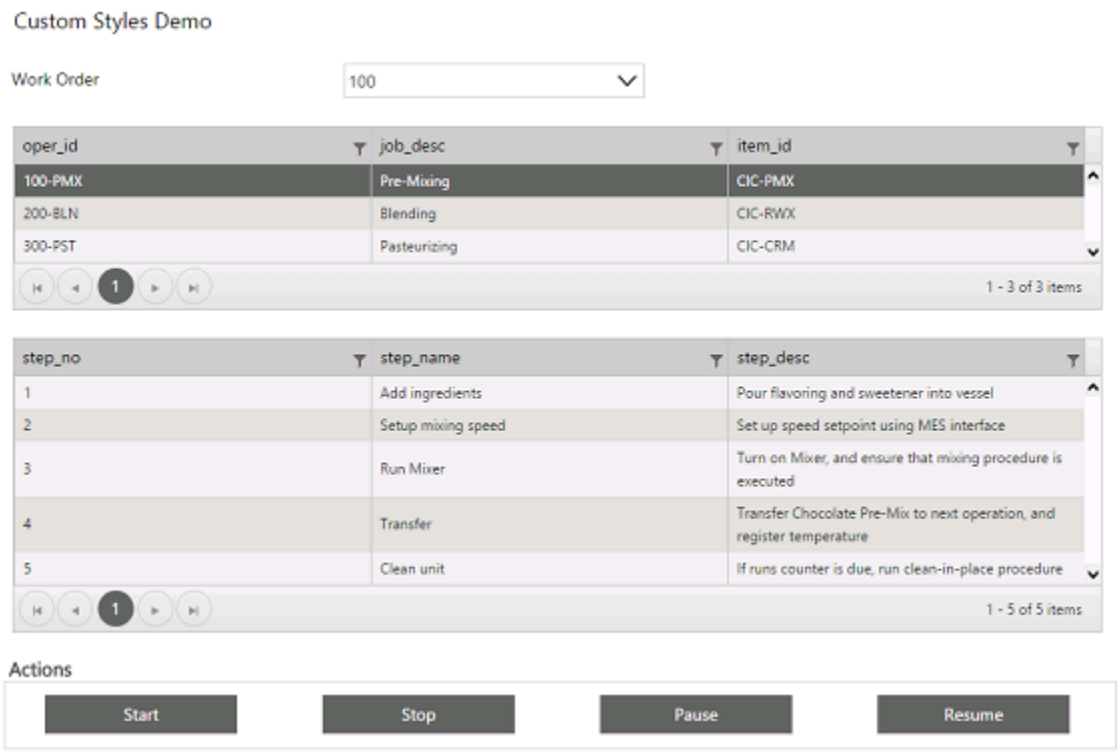
Customize Forms Using Custom Styles

You can customize forms using custom styles to change the style of the forms user interface.

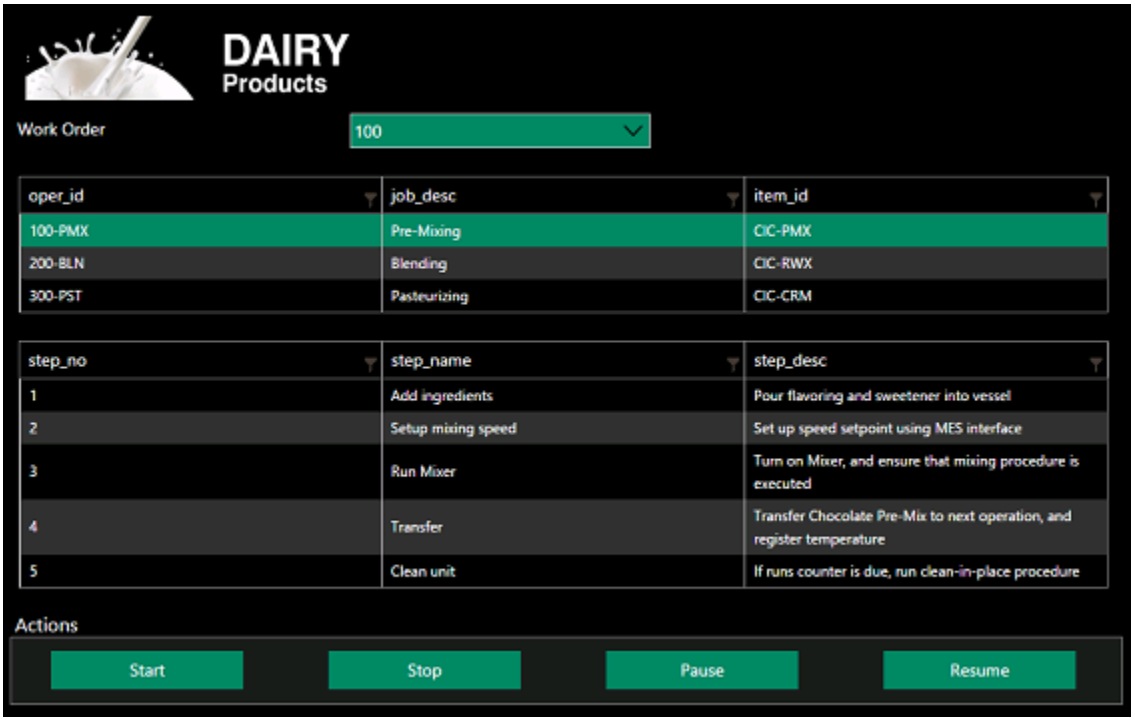
Here we will customize a form with **Data Lookup**, **Data Grid**, **Panel Form**, and **Button** controls using custom styles.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Default Form



Customized Form



1. Create a form, and add **Data Lookup**, **Data Grid**, **Panel Form**, and **Button** controls.
2. Modify properties of the form as follows:
   a. In the **Appearance** tab, set the **Custom Style Sheet** property to **WorkOrderStyles** (the name of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

custom style sheet).

For example, **WorkOrderStyles** for style sheet located at **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/css/WorkOrderStyles.css.**

Note:
- For earlier versions of the product, the location for style sheet is **BPMUITemplates/Default/Themes/<ThemeName>/NextGenForms/css/custom/WorkOrderStyles.css**
- If **Theme** is not applied and if the style sheet name is tagged with **FormCustomPath:** or **FormCustomThemePath:**, the location is **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/css/WorkOrderStyle.css**
- If **Theme** is applied and if the style sheet name is tagged with **FormCustomPath:** or **FormCustomThemePath:**, the location is **<Installpath>/Web/BPMUITemplates/Default/NextGenForms/custom/Themes/<ThemeName>/css/WorkOrderStyle.css**

3. In the **WorkOrderStyle.css**, code as follows:

```
/* Changing the Background Color */
html:not(.skd) .skfcn
{
background-color: black;
}
/* Form Header Styles */
html:not(.skd) .skh
{
background-image: url(../images/Dairy-Header2.png);
background-repeat: no-repeat;
background-position: 20px 10px;
height: 80px;
}
/* Hiding the Form Title Text*/
html:not(.skd) .skh .sktt
{
display: none;
}
/* Controls Label Styles - Using without Custom Style Identifier as it needs to be applied to all control labels */
html:not(.skd) .skctr .skfli
{
color: white;
}
/* Grid Styles - With Custom Style Identifier 'jobSteps' */
html:not(.skd) .jobSteps_skctr .k-grid
{
border-color: rgb(128, 128, 128);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
background-color: #000;
}
/* Grid Header Styles */
html:not(.skd) .jobSteps_skctr .k-grid .k-grid-header
{
border-color: rgb(128, 128, 128);
background-color: #000;
padding-right: 0 !important;
}
/* Grid Header Cell Styles */
html:not(.skd) .jobSteps_skctr .k-grid .k-grid-header table th[role="columnheader"]
{
background-color: #000;
}
/* Grid Header Cell Text Styles */
html:not(.skd) .jobSteps_skctr .k-grid .k-grid-header table th[role="columnheader"] .k-
link
{
color: #FFF !important;
}
/* Grid Content Styles */
html:not(.skd) .jobSteps_skctr .k-grid .k-grid-content
{
overflow: hidden !important;
}
/* Grid Alternate Row Styles- for even-indexed records */
html:not(.skd) .jobSteps_skctr .k-grid .k-grid-content table tr:not(.k-alt)
{
background-color: #000;
color: #FFF;
}
/* Grid Alternate Row Styles- for odd-indexed records */
html:not(.skd) .jobSteps_skctr .k-grid .k-grid-content table tr.k-alt
{
background-color: #333;
color: #FFF;
}
html:not(.skd) .jobSteps_skctr .k-grid .k-grid-content table tr.k-state-selected,
html:not(.skd) .skbtn
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
background-color: rgb(0, 136, 102) !important;
color: #FFF !important;
}
html:not(.skd) .skbtn:hover
{
background-color: #00aba9 !important;
color: #000 !important;
}
/* Grid Footer Styles */
html:not(.skd) .jobSteps_skctr [data-role="pager"]
{
display: none;
}
/* Data Lookup Styles - With Custom Style Identifier 'wo' */
html:not(.skd) .wo_skctr .ski,
html:not(.skd) .wo_skctr .skld
{
background-color: rgb(0, 136, 102) !important;
color: #FFF !important;
}
html:not(.skd) .wo_skctr .skllc
{
display: none !important;
visibility: hidden !important;
}
/* Panel Form Styles - With Custom Style Identifier 'actions' */
/* Changing Border Color and Background Color*/
html:not(.skd) .actions_skcw
{
border-color: rgb(128, 128, 128);
background-color: rgb(29, 29, 29);
}
/* Changing Title Color and Style */
html:not(.skd) .actions_skpt
{
color: #FFF;
font-weight: normal;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
```

Data Binding Drop-Down Controls Using Lookups

You can populate the options of a drop-down control, that is options binding, using a lookup through a data grid control set as data source on the same form.

You can use any type of lookup (**Database, Repository List**, or **Web API** for data binding options of **Check Box**, **List**, **Drop-Down**, or **Radio Button** control.

1. Create a lookup.
2. Create a **Grid Configuration** using the lookup.
3. Create a form, and add a **Data Grid** control.
4. Modify properties of the **Data Grid** control as follows:
   a. In the **Basic** tab, set the **Configuration Name** property to the grid configuration created in Step 2.
   b. In the **Advanced** tab, set the **Behavior** property to **Data Source**.
5. Add a **Drop-Down** control to the form.
6. Modify properties of the **Drop-Down** control as follows:
   a. In the **Advanced** tab, set the **Text Parameter** and **Value Parameter** properties to the column names of the grid as required.
   b. In the **Scripts** tab, set the script for the **Options** property as follows:

   ```
   return control.findById("G1").source;
   ```

Data Binding Drop-Down Controls Using Lookup Name (Non-parameterized)

You can populate the options of a drop-down control, that is options binding, using a lookup name.

You can use any type of lookup (**Database**, **Repository List**, or **Web API** for data binding options of **Check Box**, **List**, **Drop-Down**, or **Radio Button** control.

1. Create a lookup with the name **Products**.
   Let us consider that Products is a Database lookup with the following configuration:
   Query- select * from Products
   Value- Products.ProductKey
   Display Name- ProductName
2. Add a **Drop-Down** control to the form.
3. Modify properties of the Drop-Down control as follows:
   a. In the **Advanced** tab, set the **Text Parameter** to **ProductName** and **Value Parameter** to **ProductKey**.
   b. In the **Scripts** tab, set the script for the **Options** property as follows:
   `return SFU.getLookupSchemaAndData("Products").Data;`

Data Binding Drop-Down Controls Using a Lookup Name (Parameterized)

You can populate the options of a drop-down control that is options binding, using a parameterized lookup

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

name.

You can use any type of lookup (**Database**, **Repository List**, or **Web API** for data binding options of **Check Box**, **List**, **Drop-Down**, or **Radio Button** control.

1. Create a lookup with the name **Products** that is parameterized on **Productcategory**
   Let us consider that **Products** is a Database lookup with the following configuration:
   Query- select * from Products where ProductCategoryKey = @key
   Value- Products.ProductKey
   Display Name- ProductName
2. Add a **Drop-Down** control to the form.
3. Modify properties of the **Drop-Down** control as follows:
   a. In the **Advanced** tab, set the **Text Parameter** to **ProductName** and **Value Parameter** to **ProductKey**.
   b. In the **Scripts** tab, set the script for the **Options** property as follows:

```
var parameterCollection = [
{
Name: "@key",
Value: "Bikes"
}];
var lookupSchemaAndData = SFU.getLookupSchemaAndData("Products",
parameterCollection);
return lookupSchemaAndData.Data;
```

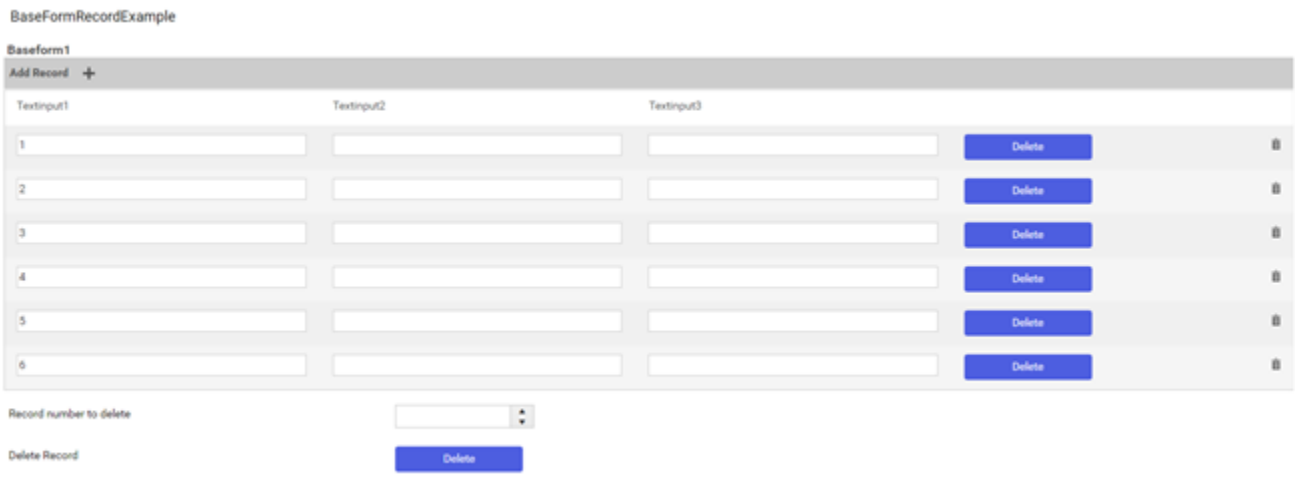
Delete Record in a Base Form Control

Consider a BaseForm control containing several **TextInput** controls and a **Delete** button with a custom OnClick script as follows:

```
control.dataContainerForm.removeRecord(control.associatedForm);
// you can also use below script to delete the record
// control.associatedForm.remove();
```

Here, the control referred to is the **Delete** button, as specified in the OnClick script.

`control.associatedForm` represents the record within the BaseForm control while `control.dataContainerForm` represents the BaseForm control itself.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For a BaseForm control, the `removeRecord` API, as demonstrated in the script, facilitates the removal of a record.

Another example is the presence of a **Number** control labeled **Record number to delete**, where the user can specify the record number to be deleted. Additionally, there is a delete record option with **Delete** button equipped with a custom OnClick script designed to delete the specified record as shown below:

```
numberRecordControlValue = control.findByXmlNode("Recordnumbertodelete").value;
var recordNumber = SFU.isNumber(numberRecordControlValue)?
parseInt(numberRecordControlValue) : 0;
if (recordNumber > 0 && recordNumber <=
control.findByXmlNode("Baseform1").records().length)
{
control.findByXmlNode("Baseform1").removeRecord(
control.findByXmlNode("Baseform1").records()[recordNumber - 1]
);
// you can also use below script by giving the record number to be deleted
// control.findByXmlNode("Baseform1").removeAt(recordNumber); // This method does not
require a zero based index value for the record number
}
else
{
SFU.showError(document.title, "Please specify a valid record number");
}
```

Related Links

- control.associatedForm
- control.dataContainerForm
- control.remove()
- control.removeAt()

Get User Information of Logged-on User

You can get user information of the logged-on user using SFU.getVirtualActorId() and SFU.getUserLookupExtendedInformation() APIs.

Here we will use these APIs to get user information of the logged-on **Active Directory** user.

1. Create a form with seven **Text** controls on the form.
2. Modify the properties of the **Form** as follows:
   a. In the **Scripts** tab, set the script for the **On Form Load** property as follows:

   // Get logged-on user information of the Active Directory User.

   var virtualActorId = SFU.getVirtualActorId();

   // Get the Extended Information for the user using the Virtual Actor ID.

   // Here loggedOnUserInformation holds the property names of the provider.

   var loggedOnUserInformation = SFU.getUserLookupExtendedInformation(virtualActorId);

   // Populate the Text controls with the user information.

   control.findById("T1").value = loggedOnUserInformation.UserDepartment;

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.findById("T2").value = loggedOnUserInformation.UserDesignation;
control.findById("T3").value = loggedOnUserInformation.UserDisplayName;
control.findById("T4").value = loggedOnUserInformation.UserDistinguishedName;
control.findById("T5").value = loggedOnUserInformation.UserEmail;
control.findById("T6").value = loggedOnUserInformation.UserId;
control.findById("T7").value = loggedOnUserInformation.UserManager;
control.findById("T8").value = loggedOnUserInformation.UserName;
control.findById("T9").value = loggedOnUserInformation.UserRole;
```

Hide the Column Header in a Data Grid

To hide the column header in a data grid, write the following script in ODB event of the grid:

```
var dataSource = skelta.grid.getDataSource(e.sender);
if(dataSource === "Job")
{
//this hides the columns header of Job grid
e.sender.thead.closest(".k-grid-header").hide();
}
```

Localize BlockingOutput Messages for Invoke Workflow Control

You can localize **BlockingOutput** messages for the **Invoke Workflow** control.

1. Create a workflow with a **BlockingOutput** activity and set the message as **@@MyBlockingOutput@@** (localization key).
2. Add a key-value pair in an existing or a new text file in the **<AVEVA Work Tasks Install Path>\Locales\<culture>\Forms2** folder in the following format:

   `@@<localization_key>@@:Localized Value`
3. Create a form, and add an **Invoke Workflow** control and a **Text** control.
4. Modify properties of the **Invoke Workflow** control as follows:
   a. In the **Basic** tab, set the **Workflow Details**, **XML Variable Name**, and **XML Schema** properties.
   b. In the **Advanced** tab, set the **Execution Mode** property to **Synchronous** to synchronously trigger the workflow created in step 1.
   c. In the **Scripts** tab, set the script for the **Post-Workflow Execution** property as follows:

      `control.findById("T1").value = skelta.localize.getString("@@MyBlockingOutput@@");`
5. Publish the workflow.
6. Preview the form, and click the **Invoke Workflow** control to get the localized value in the **Text** control.

Prevent Duplicate Data Entries in Controls

You can prevent duplicate data entries in controls for records.

In this example, we create a form with the fields, **Bill Ref No.** , **Bill Date**, and **Amount**. When you add multiple

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

records, and enter same values for the **BIll Ref No.** , a validation message appears.

1. Create a form. Here, we have named it **Duplicate Validation Sample**.
2. Add a **Base Form** control to the form and name it **Bill Details**.
3. Add a **Text** control to the **Base Form** control and modify the properties of the **Text** control as follows:
   a. In the **Basic** tab, set the **Name** property to **Bill Ref No.** , and **XML Node** to **BillRefNo**.
   b. In the **Scripts** tab, set the script for the **Validate** property as follows:

```
/*Check for Bill Reference Number duplication in the form.*/
var returnObj = new ValidationOptions();
returnObj.isValid = true;
returnObj.message = "";
var currentContainer = control.dataContainerForm;
var currentContainerRecords = currentContainer.records();
var currentRecordIndex  = currentContainerRecords.indexOf(control.associatedForm);
for (var i = 0; i < currentContainerRecords.length; i++)
{
// Need not check for duplicate values in the same record.
  if (currentRecordIndex !== i)
  {
    if
(currentContainerRecords[i].findChildRecordControlByXmlNode("BillRefNo").value.toLowerCase()
=== currentValue.toLowerCase())
    {
      returnObj.isValid = false;
      returnObj.message = "Duplicate values are not allowed";
      break;
    }
  }
}
return returnObj;
```

4. Add a **Date** control, and name it **Bill Date**.
5. Add a **Number** control, and name it **Amount**.
6. **Save** and **Publish** the form.
7. **Preview** the form.
8. Enter a value under the **Bill Ref No.** field.
9. Add another record, and enter the same value again under the **Bill Ref No.** field.
   The following validation message appears below the **Bill Ref No.** field of the second record:
   Duplicate values are not allowed
10. Click **Finish**. An error message appears. The validation message also appears below the **Bill Ref No.** field of

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the first record.

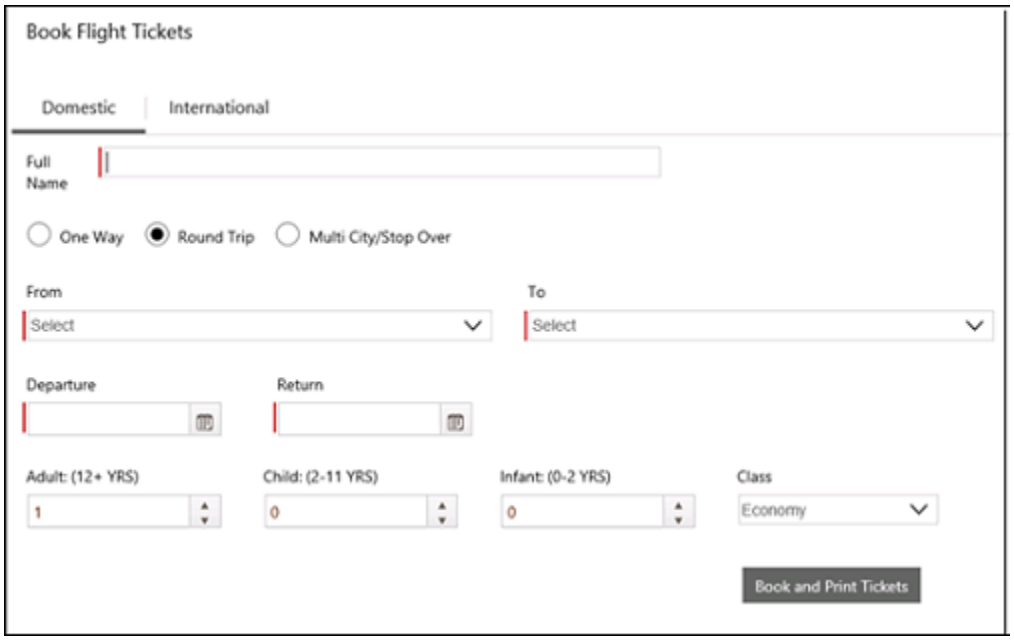
Print a Form

In the following example, we create two forms. The first form is the main form , called **Book Flight Tickets**. The user uses this form to book the flight tickets.

The second form is the **Booking Confirmation** form. This form serves as a confirmation for the tickets booked.

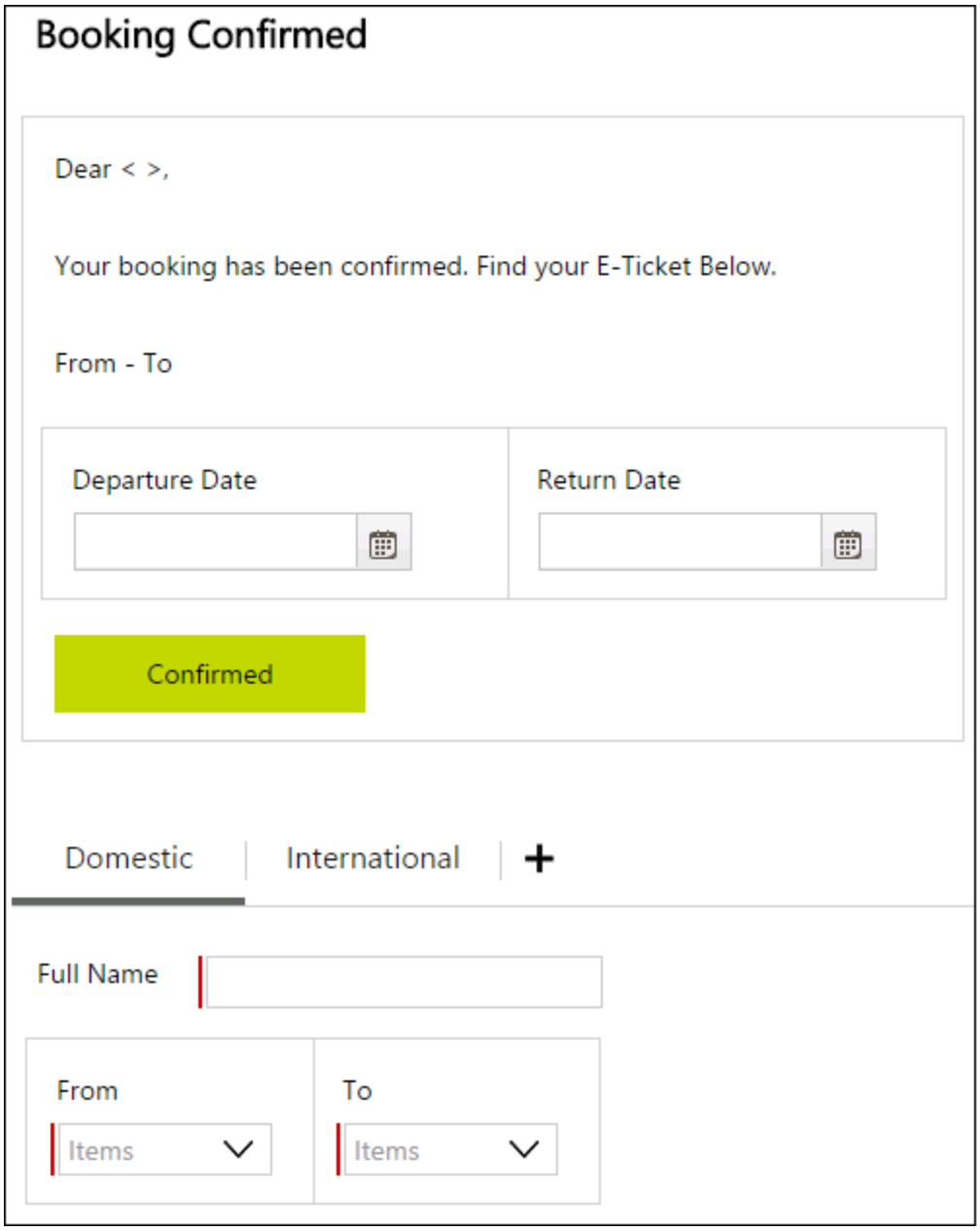
The controls in the **Booking Confirmation** form are mapped to the controls of the **Book Flight Tickets** form. Also, the **Print Form** property in the **Book Flight Tickets** form is set to the **Booking Confirmation** form. After the user books the flight tickets using the the **Book Flight Tickets** form, and clicks **Book and Print Tickets**, the user receives a confirmation.

1. Create the **Book Flight Tickets** form as shown in the image below.



2. Modify the properties of the **Book and Print Tickets** button control as follows:
   a. In the **Basic** Tab, set the **Button Type** as **Print**.
3. Create the **Booking Confirmation** form as shown in the following image:

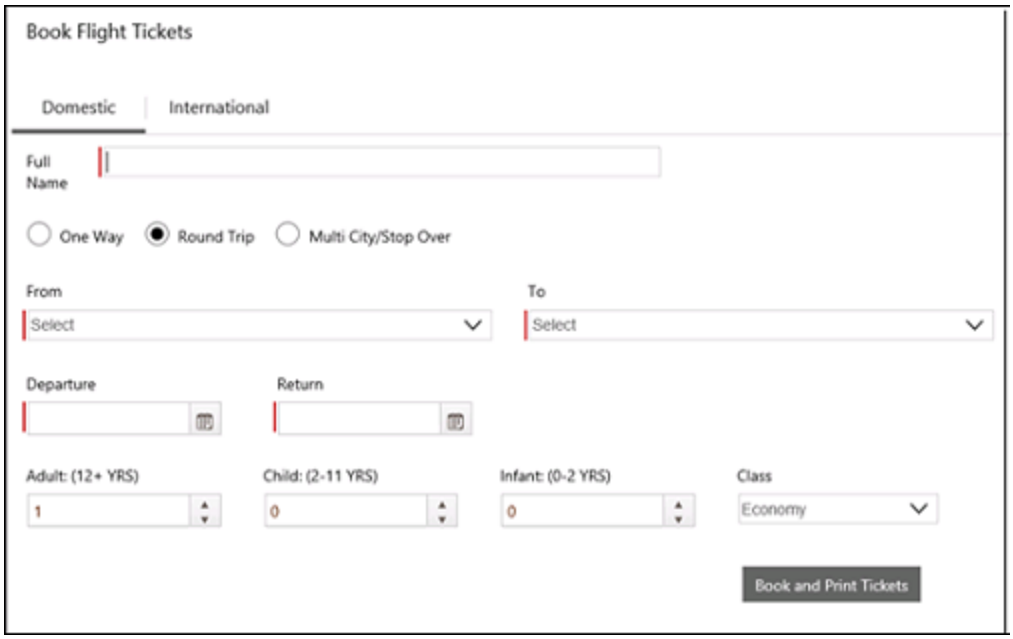
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



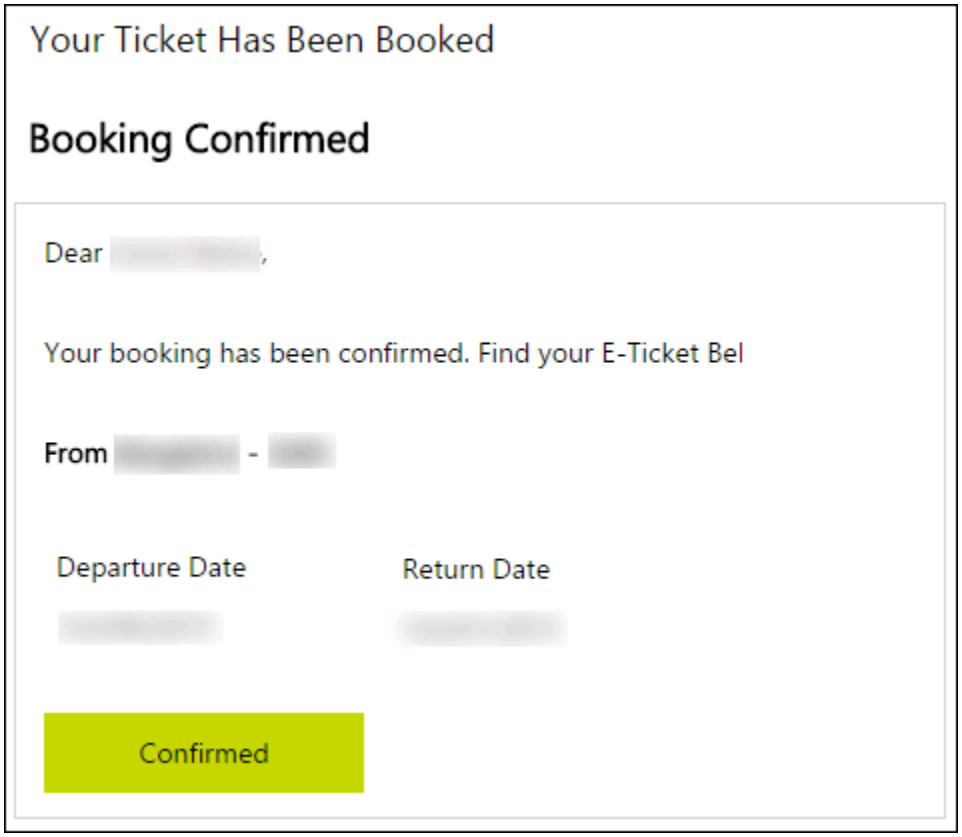
Note: Make sure the XML node value is same across the main form and the print form for the values to be mapped.

4. Modify the properties of the **Book and Print Tickets** button control as follows:
   a. In the **Basic** Tab, set the **Print Form** property to the **Booking Confirmation** form.
5. Publish both the forms.
6. Preview and fill the values in the **Book Flight Tickets** form. Refer to the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



7. Click **Book and Print** Tickets.The following screen appears.



You can now print this confirmation.

Select and Act on Multiple Work Items Using the Enable Multi Select option

In the following example, we demonstrate the use of the **Enable Multi Select** option in the **Inbox**, by selecting

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

multiple work items related to Invoke Form Activity and acting on them from the inbox.

1. Create a **Leave Request** form, with **From** and **To** fields using **Date** control and and a **Comments** field using the **Text** control.
2. Create a **Manager Approval** form with a **Comments** field using **Text** control.

   Note: Ensure that the XML Node property details of the **Text** control (Comments field) is the same in the **Leave Request** and **Manager Approval** forms.
3. Create a workflow, with an **Invoke Form Activity**. Name the workflow as **Leave Manager**.
4. Modify the activity properties of the **Invoke Form Activity** as follows:
   a. Set the **Form Name** property to the Leave Request form.
   b. Select **Input Instance XML** as **SFFormData**.
   c. Set the **Form to be opened on work item multi-select** to **Manager Approval** form.
   d. Select **Output Instance XML** as **SFFormData**.
5. In **Fill Form**, click **Leave Request**.The **Leave Request** form opens. Select the **From** and **To** date and click **Finish**.

   The first leave request gets submitted.
6. Repeat step 5 to create multiple leave requests.
7. In **Inbox**, click the **Enable Multi Select** option.

   Check boxes appear under **Workflows**, indicating that the **Enable Multi Select** option is enabled.
8. Right-click on a workflow and select **Select all similar items**.

   All the similar workflows get selected and the **Manager Approval** form opens.
9. Type your comments and click **Finish**.
10. All the selected leave requests are removed from **Fill Form** and their output instance XML gets updated with the common comments given in the **Manager Approval** form.

Set the Consider Available Height and Dynamic Page Size Properties

1. Create and design a form with the following controls:
   - Two Tabs
   - Panel Form with two columns
   - Data Grid
   - Embed Form
2. Configure the required properties in all the above controls and set the **Consider Available Height** property for all the controls to **Yes**.
3. In the **Data Grid** control, set the **Dynamic Page Size** property to **Yes**.
4. If the controls are within a nested container and if the value of the **Consider Available Height** property is set to **Yes** for the child control, then this property should be set to **Yes** in all parent containers.

Set Values for a Data Grid on Post Execution of a Workflow

You can set values for a Data Grid, after a workflow is executed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Create a form with a **Data Grid**, **Hidden Field**, and **Invoke Workflow** controls.
2. Create a **Work Order** lookup.
3. Modify properties of the Data Grid control as follows:
   a. In the **Basic** tab, set the **Configuration Name** property to **WorkOrder** lookup created in step 2.
   b. In the Advanced tab, set the script for the **On Data Bound** property as follows:

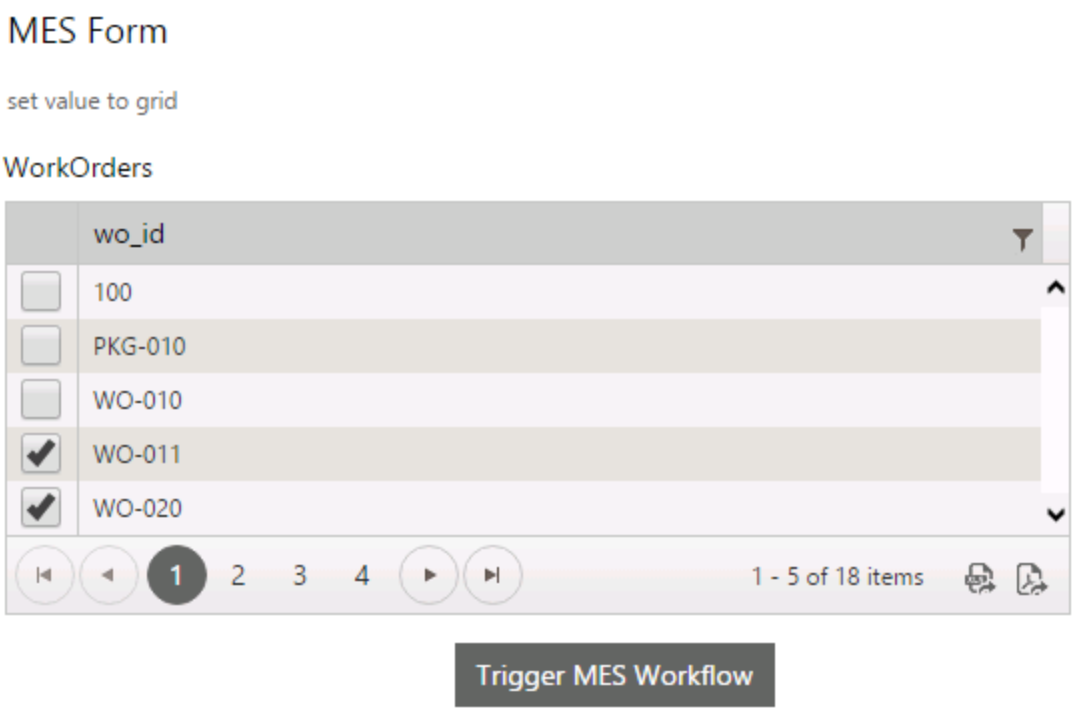
```
var gridValues = control.findById("H1").value;//"100,WO-010"
if (gridValues)
{
var workOrders = [];
var values = gridValues.split(",");
for (var i = 0; i < values.length; i++)
{
workOrders.push({ "wo_id": values[i] });
}
//grid object has to be in this json object format
//Grid object with property data which is again an object with property of workorder which is lookup name which is always an array of objects with property as perists column names
var gridValue = { Grid: { Data: { WorkOrder: workOrders } } };
//the below api is used to get xml from the object.
var gridXml = SFU.getXmlFromJSONObject(gridValue);
//grid value is encoded xml
control.value = encodeURIComponent(gridXml);
}
```

   The script gets a value from the **Hidden Field** control,and sets the value for the Data Grid. The Data Grid internally selects the appropriate checkboxes.
4. Save the Data Grid.
5. In the **Scripts** tab, set the script for the **Post-Workflow Execution** property as follows:

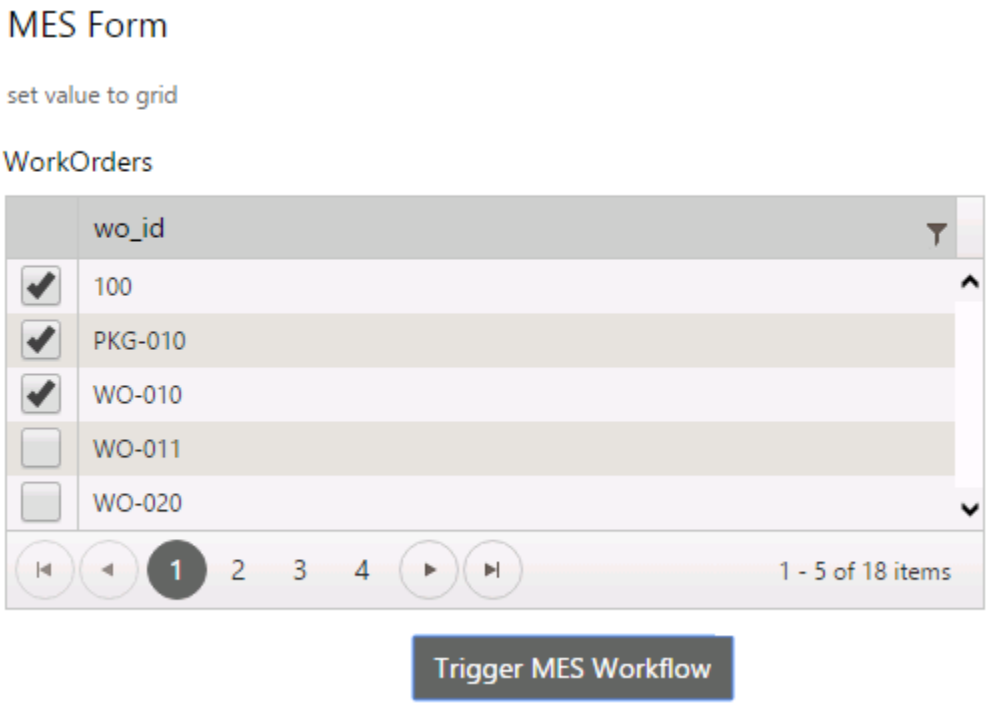
```
control.findById("H1").value= blockingOutput;
control.findById("G1").refreshGrid();
```

   The script sets the value for the **Hidden Field** control with the blocking output, and the **Data Grid** control is refreshed.
6. Save the Invoke Workflow.
7. During run time, select the appropriate checkboxes on the grid,and trigger the workflow using the Invoke Workflow .

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



8. During post execution of the workflow, the grid is refreshed and the new values are automatically selected.



Show Embedded Data from Web API Lookup in Data Grid

You can map **Web API Lookup** with columns containing JSON objects to **Data Grid** columns to show embedded data instead of JSON strings using scripts.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Here we will parse the JSON string in the 8th column of the **Data Grid** to show only the names.

1. Create a **Web API Lookup** with a column containing JSON object.
2. Create a **Grid Configuration** using the lookup.
3. Create a form, and add a **Data Grid** control.
4. Modify properties of the **Data Grid** control as follows:
   a. In the **Advanced** tab, set the script for the **On Data Bound** property as follows:

```
var kGrid = e.sender;
var allRows = $(kGrid)[0].tbody[0].children;
ko.utils.arrayForEach(allRows, function(eachRow)
{
// Here 8 is the column number containing JSON object.
var oldText = $(eachRow).find("td:eq(8)").text();
$(eachRow).find("td:eq(8)").text($.parseJSON(oldText).Name);
});
```

Note: The script only affects the displayed values in the **Data Grid**. Filtering and sorting is still based on the actual JSON string and not the displayed data.

Show Icons in Data Grid Columns

You can replace text with icons in a column of a **Data Grid** using CSS.

Here we will replace the text **Yes** and **No** with a green and a red colored icon respectively in the **Published** column of the **Data Grid**.

1. Create a lookup (**Database**, **Repository List**, or **Web API**).
2. Create a **Grid Configuration** using the lookup.
3. Create a form, and add a **Data Grid** control.
4. Modify properties of the **Data Grid** control as follows:
   a. In the **Basic** tab, set the **Configuration Name** property to the grid configuration created in Step 2.
5. Place images (icons) for replacing the text in the **<Installpath>/Web/<BPMUITemplates>/NextGenForms/custom/Themes/<ThemeName>/images** folder.

   Note:
   - If the image name is tagged with **FormCustomPath:**, the location of image is **<Installpath>/Web/<BPMUITemplates>/NextGenForms/custom/images**
   -If the image name is tagged with **FormCustomThemePath:**, the location of image is **<Installpath>/Web/<BPMUITemplates>/NextGenForms/custom/Themes/<ThemeName>/images**
   - For AVEVA Work Task Pro App, if the image name is tagged with **FormCustomPath:** or **FormCustomThemePath:**, the location is **<Installpath>/Web/<BPMUITemplates>/NextGenForms/custom/images**
   - For AVEVA Work Tasks Pro App, mapping of themes is not considered.
6. Create a CSS file, code it as follows, and place it in the **<Installpath>/Web/<BPMUITemplates>/NextGenForms/custom/Themes/<ThemeName>/NextGenForms/**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

css folder.

```
.Published
{
background-repeat: no-repeat;
background-image: url("../images/bullet_green.png");
background-position: center;
}
.Draft
{
background-repeat: no-repeat;
background-image: url("../images/bullet_red.png");
background-position: center;
}
```

7. Modify properties of the **Data Grid** control as follows:
   a. In the **Advanced** tab, set the script for the **On Data Bound** property as follows to replace text with icons in the **Published** column at run-time:

```
var kGrid = e.sender;
var allRows = $(kGrid)[0].tbody[0].children;
ko.utils.arrayForEach(allRows, function(eachRow)
{
// Here 5 is the Published column
// which is the fifth column in the Data Grid
// with values Yes or No.
var oldText = $(eachRow).find("td:eq(5)").text(), className = "";
if(oldText === "Yes")
{
className = "Published";
}
else
{
className = "Draft";
}
$(eachRow).find("td:eq(5)").text("");
$(eachRow).find("td:eq(5)").addClass(className);
});
```

Note: The records in a grid are not filtered when text is replaced with icons.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Update Column Names in a Base Form

You can update the column names in a **Base Form** with **Display Mode** set to **Grid** by updating the tag name of the control present in the column.

1. Create a form, and add a **Base Form** container control.
2. Add 2 controls to the record of the **Base Form** container control as follows:
   - **Text** control for the first column.
   - **Number** control for the second column.
3. Add a **Button** control and name it **Update Column Names**.
4. Modify properties of the **Button** control as follows:
   a. In the **Basic** tab, set the **Button Type** as **Custom**.
   b. In the **Advanced** tab, set the script for the **On Click** property as follows:

```
control.findByXmlNode("FirstColumn").tagName = "Updated Column 1 Name";
control.findByXmlNode("SecondColumn").tagName = "Updated Column 2 Name";
```

Use Data from Data Grid Set As Data Source

You can set the **Behavior** property of a **Data Grid** to **Data Source** to use the data without showing the **Data Grid** control on the form.

Here we will use data from a **Data Grid** set as **Data Source** to set the **Button Text** for several buttons.

1. Create a lookup (**Database**, **Repository List**, or **Web API**).
2. Create a **Grid Configuration** using the lookup.
3. Create a form, and add a **Data Grid** control.
4. Modify properties of the **Data Grid** control as follows:
   a. In the **Basic** tab, set the **Configuration Name** property to the grid configuration created in Step 2.
   b. In the **Advanced** tab, set the **Behavior** property to **Data Source**.
5. Add a **Text** control and several **Button** controls to the form.
6. Modify the properties of the **Text** control as follows:
   a. In the **Scripts** tab, set the script for the **Value** property as follows to set the **Button Text** with the **Job_wo_id**.

```
var numRec;
var ctrlId;
var dataSrc = control.findById("G1").source;
// Loop through the records returned.
if(dataSrc.length > 0)
{
for (i = 0; i < dataSrc.length; i++) {
numRec = i + 1;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
ctrlId = "B" + numRec;
control.findById(ctrlId).buttonText = dataSrc[i].Job_wo_id;
}
}
// Return number of records.
return dataSrc.length;
```

Use Data from Parameterized Data Grid in Base Form

You can populate controls on a **Base Form** on selection of a record in a **Data Grid**, using data from another **Data Grid** set as **Data Source**.

Here we will use the **Work Order** data lookup to populate the **Jobs** data grid on selection of a **Work Order ID** from the **Work Order** data lookup. Then we will populate the controls in the **Job Steps** base form from the selection of the record in the **Jobs** data grid, using data from **Job Steps** data grid that was set as data source.

1. Create 3 lookups (**Database**, **Repository List**, or **Web API**):
   - **Work Order**
   - **Jobs** (parameterized on **Order ID** from **Work Order** lookup)
   - **Job Steps** (parameterized on **Work Order ID** and **Jobs ID** from **Jobs** lookup)
2. Create a **Grid Configuration** using the lookups:
   - **Jobs**
   - **Job Steps**
3. Create a form, and add a **Data Lookup** control for **Work Order** lookup.
4. Modify properties of the **Data Lookup** control (**Work Order**) as follows:
   a. In the **Basic** tab, set the **Lookup Source** property to the **Work Order** lookup created in Step 1.
5. Add a **Data Grid** control for **Jobs** lookup.
6. Modify properties of the **Data Grid** control (**Jobs**) as follows:
   a. In the **Basic** tab, set the **Configuration Name** property to the grid configuration created in Step 2, and set the **Parameters** as **Work Order** (that is the **Work Order** data lookup control).
   b. In the **Advanced** tab, set the **Select First Record** property to **Yes** to automatically select the first record when the **Data Grid** control is loaded.
7. Add a **Data Grid** control for **Job Steps** lookup.
8. Modify properties of the **Data Grid** control (**Job Steps**) as follows:
   a. In the **Basic** tab, set the **Configuration Name** property to the grid configuration created in Step 2, and set the **Parameters** as **Work Order ID** and **Jobs ID** (from **Jobs** data grid control).
   b. In the **Advanced** tab, set the **Behavior** property to **Data Source**.
9. Add a **Base Form** container control for **Job Steps**.
10. Add 3 controls to the record of the **Base Form** container control (**Job Steps**) as follows:
    - **Text** control for **JobStepDescription**
    - **Number** control for **ProcessOrderID**
    - **Number** control for **BatchNumber**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

11. Modify properties of the **Data Grid** control (**Jobs Steps**) as follows:
    a. In the **Advanced** tab, set the script for the **On Data Bound** property as follows to populate the controls on the record of the **Base Form** container control (**Job Steps**):

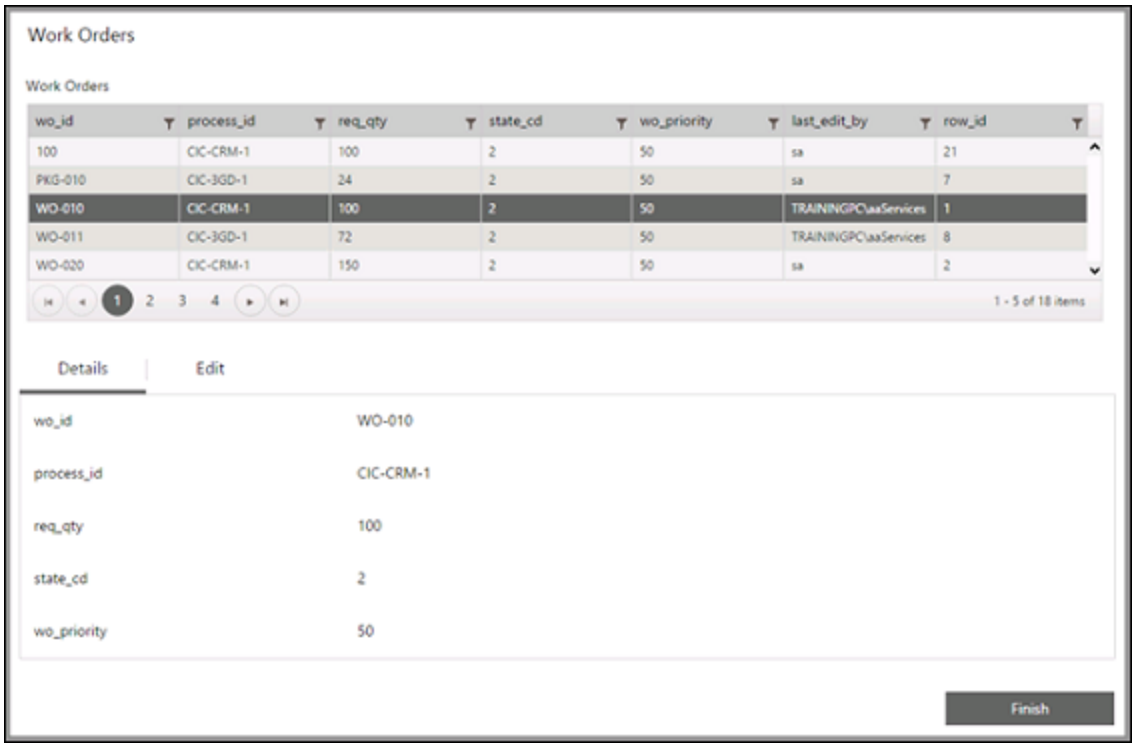
```
// Here B1 is the Job Steps Base Form container control.
var b1 = control.findById("B1");
b1.removeAll();
var stepsSource = control.source;
if (stepsSource && stepsSource.length > 0)
{
ko.utils.arrayForEach(stepsSource, function (parameter)
{
var newRecord = b1.addRecord();
newRecord.findById("T1").value = parameter.JobStepDescription;
newRecord.findById("N1").value = parameter.ProcessOrderID;
newRecord.findById("N2").value = parameter.BatchNumber;
});
}
```

Use the Security APIs

In the following example, we create a **Work Order** form and we hide controls based on the Security Group of the user viewing the form.

1. Create a **Work Order** form as shown in the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



2. Modify the properties of the **Edit** tab of the **Tabs** control as follows:
   a. In the **Scripts** tab, set the script for the **Visible** property as follows:

```
var securityGroup = SFU.getUserSecurityGroup();
if(securityGroup === "WorkOrderAdmin")
{
return true;
}
return false;
```

3. Publish the form.

The **Edit** tab is not visible to a user who does not belong to the **WorkOrderAdmin** security group.

Note: You can use `SFU.getUserRoles()` and `SFU.getValueForRightCode()` security API's in the same way as described in this example.

View Attachments uploaded from Forms

You can view attachments uploaded from forms in ASPX pages using scripts as follows.

Here we will view a photograph uploaded from a form in an ASPX page.

1. Create a list and the dynamic list form.
2. Populate the list using the dynamic list form, and also upload the photograph using the attachment control on the form.
3. Create an ASPX page with a button control.
4. Code the on click event for the button as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
protected void Button_Click(object sender, EventArgs e)
{
securedValue.HeaderPersisted = true;
this.LoadFileContent(true, "Emp001");
}
private void LoadFileContent(bool isHeaderPersisted, string title)
{
var dbConnection = new SqlConnection();
try
{
int startIndex = 0;
if (isHeaderPersisted)
{
AttachmentHeader header = new AttachmentHeader();
// Repository Name
header.ApplicationName = "Forms Repository";
// Attachment File Name
header.FileName = securedValue.FileName;
header.HeaderVersion = 1;
byte[] info = header.BuildHeader();
startIndex = info.Length;
}
// Connection string
dbConnection.ConnectionString =
ConfigurationManager.ConnectionStrings["myConnectionString"].ConnectionString;
// SQL command for querying the database table
var command = new SqlCommand("select Id, Photograph from Employee where Title=@title",
dbConnection);
// Parameter for the title column
command.Parameters.AddWithValue("@title", title);
// Open the database connection
dbConnection.Open();
// Execute reader
var reader = command.ExecuteReader();
// Read the data from reader and write the same into the Context.Response
if (reader.HasRows)
{
while (reader.Read())
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

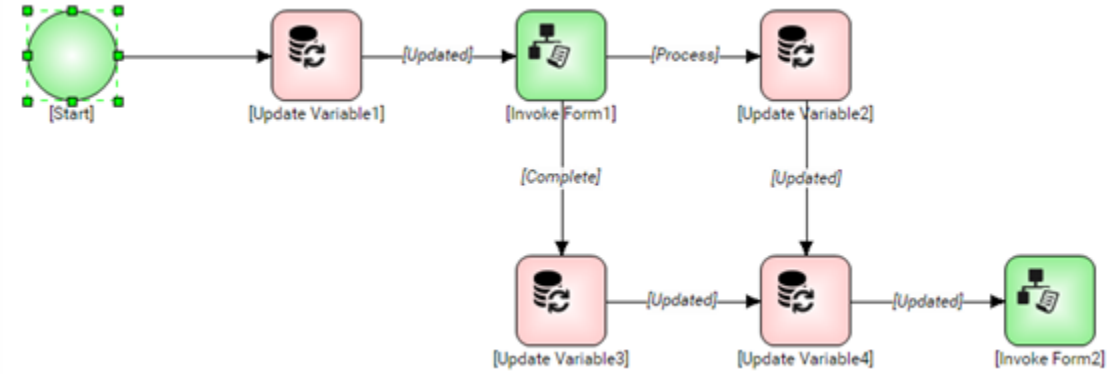
```
    Byte[] bytes = (Byte[])reader.GetValue(1);
    var ms = new MemoryStream(bytes);
    Byte[] thumbnailByteArray = new Byte[ms.Length];
    ms.Position = startIndex;
    ms.Read(thumbnailByteArray, 0, Convert.ToInt32(ms.Length));

// File type
Response.ContentType = "application/jpg";
Response.BinaryWrite(thumbnailByteArray);
Response.Flush();
}
}
}
catch (Exception ex)
{
throw ex;
}
finally
{
// Close the database connection
dbConnection.Close();
}
```

Access Attachments from Another Form

You can access or use the Attachments submitted in a different form by using Update Variable Activity.

Consider, we have a form that can submit the feedback from users with some attachment as a reference. After feedback has been submitted, it initiates a workflow, which has an Invoke Form activity pointing to another Form, wherein we need to access the attachment submitted by the user for further evaluation. To do this, we would need to use an Update Variable Activity.

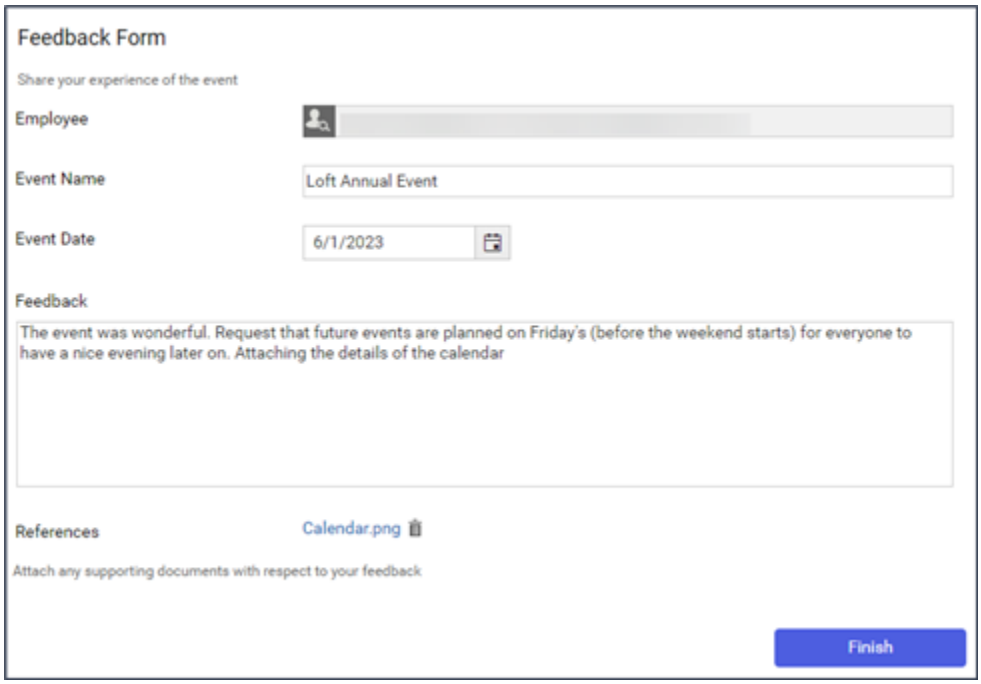
This is the sample workflow to understand the use case of accessing attachments from another Form.



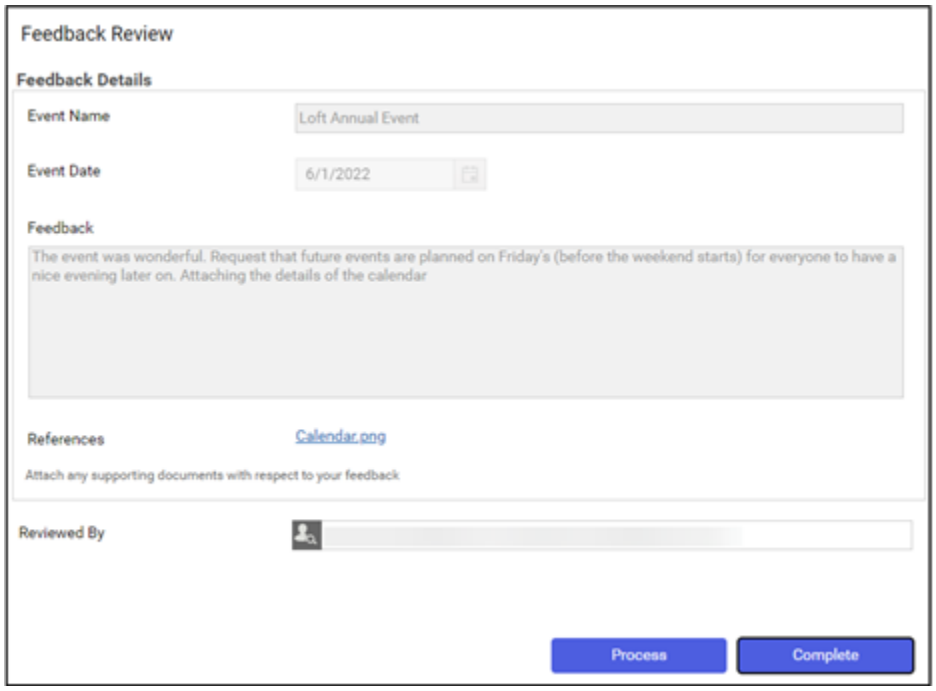
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To access attachments from another Form

1. Create a Feedback Form with fields and attachment control as shown in the image below.



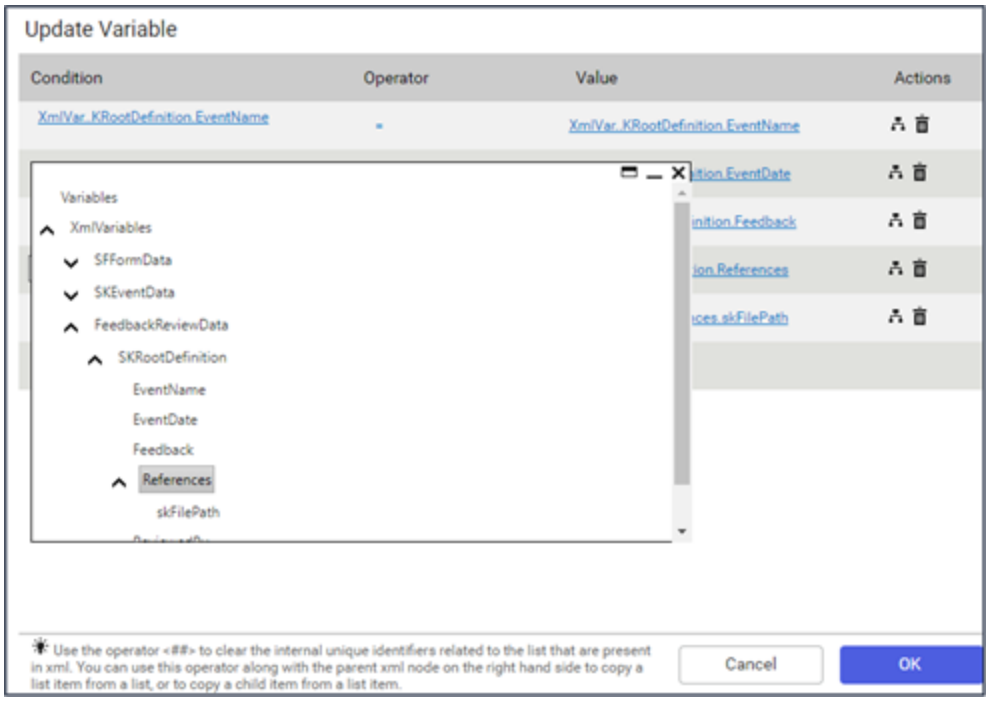
2. Create another Feedback Review Form.



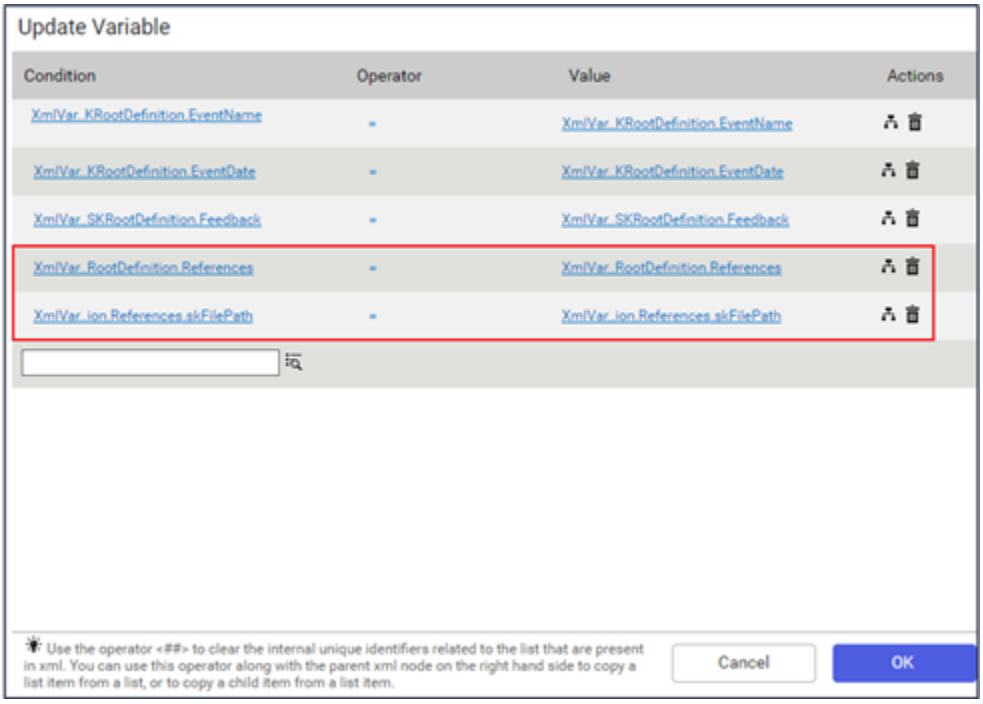
3. Create Update Variable activity with the XMLVariables shown in below image.

   In the Update Variable activity, update the LHS and RHS to transfer the Data between the XML Variables representing the **Feedback Form** and **Feedback Review Form**, wherein the **RHS** would be **Feedback Form Xml Variable** nodes and **LHS** would be the **Feedback Review Form Xml Variable** nodes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The submitted Form information gets stored in the **SFFormData** Xml Variable. The Invoke Form Activity pointing to another form used for Verification would be referring to another Form Xml Variable, say **FeedbackReviewData**. To have the Attachment details present in the **FeedbackReviewData** Xml Variable, we would need to assign the Xml Node **References** and its attribute node **skFilePath** from **SFFormData** to **FeedbackReviewData** as shown below (the Attachment node is named as **References** in the below image):



Now the Invoke Form Activity workitems, representing the **Feedback Review Form**, could refer to the **Attachment**, which was done in the **Feedback Form** submitted by the users.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Troubleshooting Forms

This section provides information on the common problems while using Forms and the resolutions to overcome those problems. It also lists the supported library versions and answers to some frequently asked questions.

Troubleshoot

1. Forms and other pages in the Enterprise Console does not respond when left idle for a long time.

Resolution:

Log on to the **Enterprise Console** again.

2. In the Safari browser:

1. Tabbing through the forms controls does not focus on the control and skips controls that are rendered as links.
2. Adding records in the **Grid** view of a **Base Form** at run-time by pressing the Tab key from the last focused control of the record does not work.

Resolution:

Edit the advanced settings of the Safari browser and select **Press Tab to highlight each item on a webpage** check box.

3. Unable to display image using the Image control.

Possible Causes:

In the HTML code, the image src attribute of the image tag takes the computer name instead of the IP address. The image is not displayed if the computer name of the machine is not resolved.

Resolution:

Analyze the HTML code using the developer tool of the browser. If the computer name is not being resolved in the **img src** attribute, contact your network administrator.

4. Adding values of number controls using the Value script appends the numbers instead of summing them up.

```
// Adding number values without typecasting the values will append the numbers.
instOne = control.findById("N1").value;
instTwo = control.findById("N2").value;
return instOne + instTwo;
```

Resolution:

Typecast the values of number controls to numeric values before summing them up.

```
// Typecast the values of number controls to numeric values to sum the numbers.
instOne = control.findById("N1").value;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
instTwo = control.findById("N2").value;
if(instOne != "")
 instOne = Number(instOne);
if(instTwo != "")
 instTwo = Number(instTwo);
return instOne + instTwo;
```

5. An alert message Unable to get property <column name> of undefined or null reference appears when an item is selected and cleared in the Data Grid control with the Extended Information property set to Yes, and the value in the bound Text control is not cleared.

Resolution:

Script as follows to resolve this issue.

```
var rowData = control.findById("G1").extendedInformation; // Array
if(rowData.length > 0)
 {
 control.findById("T1").value = rowData[0].ID;
 }
else
 {
 control.findById("T1").value = ""; // Undefined
 }
```

6. After upgrading from Skelta BPM.NET 2011 to Wonderware Skelta BPM 2014, the maximum transfer size of JSON strings from server to client is restricted to 102400 characters.

Possible Causes:

The default maximum transfer size of JSON strings from server to client is 102400 characters, which is equivalent to 100 KB of Unicode string data.

Resolution:

In the **web.config** file, edit the maximum length of JSON strings as follows:

```
<configuration>
 <system.web.extensions>
 <scripting>
 <webServices>
 <jsonSerialization maxJsonLength="4096000"/>
 </webServices>
 </scripting>
 </system.web.extensions>
</configuration>
```

7. In the On Form Load script, commented statements are executed.

Resolution:

In the JavaScript code, remove the commented statements.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

8. In the iOS WorkTask app:

1. Debugger does not work.
2. Form does not render.

Resolution:

In the JavaScript code, remove the debugger statements.

9. In Firefox browser, dialog boxes do not appear after selecting Prevent this page from creating additional dialogs.

Possible Causes:

The **Firefox** browser provides an option to the user to turn off subsequent dialog boxes if additional dialog boxes are shown within 3 seconds.

Resolution:

Reload the web page and do not select **Prevent this page from creating additional dialogs**, if it appears again.

10. Data Lookup control shows “T” separator between Date and Time for Date Time data.

Possible Causes:

JSON Date Time data has “T” separator between Date and Time.

Resolution:

Return database column containing Date Time data as string when retrieving date time from the database.

For example, convert LastUpdatedDateTime column from date time to string as follows:

```
SELECT Id,Name, LastUpdatedDateTime, CONVERT(nvarchar(150), LastUpdatedDateTime,121) AS
DisplayUpdatedDateTime FROM Employee
```

11. When API confirmation message, API Alert message, API Error message, and API warning message are present in one button control, only the first API message is displayed on a surface device.

Possible Causes:

Scripting guidelines to display messages are not followed.

Resolution:

Script as follows to resolve this issue.

```
SFU.showConfirmation("Test <b>Confirmation</b>", "Confirmation Details:Do you <b>wish</b>
to continue?",
function (val)
{
if (val)
{
SFU.showAlert("Continue", "Confirmation Process will be continued.<br/>Please contact admin
for more information.",
function ()
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
SFU.showWarning("Test ,<b>Warning</b>", "Warning Details:Do you <b>wish</b> to continue?",
function (val)
{
if (val)
{
SFU.showAlert("Continue", "Warning Process will be continued.<br/>Please contact admin for
more information.");
}
});
});
}
});
```

12. In a form with Data Grid control, at runtime if you select one record, multiple records are checked automatically. Also, the On Selection Change event gets triggered multiple times.

Possible Causes:

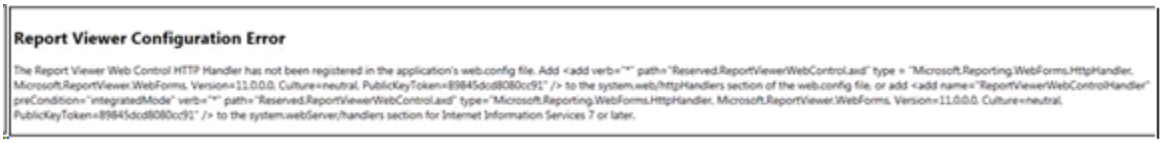
In the Data Grid configuration, the Persist Columns selected are not unique.

Resolution:

Ensure that the Persist Columns are unique.

13. Report Viewer configuration error.

The following error message appears when you try to view a Form that contains a Report Viewer control.



Resolution:

If the web site is created in integrated mode, enter the following entry in the webserver\handler section of the web.config file of the respective Microsoft SharePoint site:

```
<add verb="*" path="Reserved.ReportViewerWebControl.axd"
type="Microsoft.Reporting.WebForms.HttpHandler, Microsoft.ReportViewer.WebForms,
Version=11.0.0.0, Culture=neutral, PublicKeyToken=89845dcd8080cc91" validate="false" />
```

14. The error message, depicted in the image below, appears in the following scenarios:

- Design a form with a custom control, take a backup of the repository, and use that repository as an attached data source. Replace the control in a form when the custom control dll is not present in the specific environment.
- Import a form into an environment where the custom control is not present.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Resolution:

Perform the following to ensure that the custom control is present in the environment:

- Ensure that the dll is present in the SOA Folder or GAC.
- Ensure that all the required js and css files are present in the NextGenForms Folder.
- Create a SOA folder with the same name and add the same physical path where the custom control dll is present.
- Create Form Controls with the same name for the imported and exported Forms.

15. The error message, depicted in the image below, appears in the following scenarios:

- Design a form with a custom control, which has an exception.
- Replace a control from **Replacement Control Configuration** with a custom control, which has an exception.

Resolution:

Perform the following to ensure that there is no error while loading the control:

- Ensure that the dll is present in the SOA Folder or GAC.
- Ensure that custom control is configured in the Forms Control.
- Correct the exception and replace the dll.

16. The error messages, depicted in the below images, appears in the following scenarios:

- When a Form definition is imported, the below error appears from the custom control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Resolution:

Perform the following to ensure that there is no error while loading the control:

- Ensure that the dll is present in the SOA Folder or GAC.
- Ensure that custom control is configured in the Forms Control.
- Correct the exception and replace the dll.
- Open the XML file and delete the custom control definitions.

17. The JavaScript "debugger;" statement is not present in the Forms script during run time.

Resolution:

As part of performance tuning, we had introduced the minification of the View model when publishing the Form. The debugger works in the preview page, but in Fill Form, Form link, Work Item, or any places other than the preview page, the View model does not have the debugger statement due to minification.

In an environment, you can change this behavior. In the Farm DB, there is a SKRepository table. In the **Properties** column, there is a property **formEnableMinification** which has the value **True**. If changed to **False**, the minification does not happen for the published Forms and you can debug the Form. After changing the property value in the table for the respective repository, republish the Form for the changes to take effect.

Note: You should not change the property value in the production environment.

18. External URL for Web API, Embed Page, Images, and so on do not load in the Form, and an error gets logged in the Browser Console as "Refused to … because it violates the following Content Security Policy …"

Possible Cause:

Loading of the External URL for Web API, Embed Page, Images, and so on depends upon the Content Security Policy for the Enterprise Console Web Application.

Resolution:

This applies to images used in controls like Image, RichText, and so on in Forms and Embed Page control with the URL set to a different domain. The Content Security Policy can be modified as per user requirement and defined accordingly. For more details, refer to Content Security Policy documentation on the W3C website. You can specify the URLs based on the same domain in the Form controls. For images, you can use data-URLs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Supported Library Versions

The version of the library supported in the release is as follows:

AVEVA Work Tasks 2023

Library	Version
ajaxmin	5.14.5506
angular	12.2.x
codemirror	5.58.3
crypto-js	3.1.2
date	1.0 Alpha-1
jquery	3.6.0
jquery.dotdotdot	1.7.4
jshint	2.12.0
jsonpath	1.0.0
jszip	3.10.1
kendo UI	2021.3.1207
knockout	3.5.1
knockout-kendo	0.10.0
Microsoft.ReportingServices.ReportViewerControl.WebForms	15.0.1537.0
Newtonsoft JSON.NET	13.0.1
paho-mqtt	1.1.0
pako-deflate	2.0.4
spin	2.0.0
xml2json	1.1.5

AVEVA Work Tasks 2020 Update 2

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Library	Version
ajaxmin	5.14.5506
angular	12.2.x
codemirror	5.58.3
crypto-js	3.1.2
date	1.0 Alpha-1
jquery	3.5.1
jquery.dotdotdot	1.7.4
jshint	2.12.0
jsonpath	1.0.0
jszip	3.5.0
kendo UI	2021.3.1207
knockout	3.5.1
knockout-kendo	0.10.0
Newtonsoft JSON.NET	12.0.3
paho-mqtt	1.1.0
pako-deflate	2.0.2
spin	2.0.0
xml2json	1.1.5

AVEVA Work Tasks 2020 Update 1

Library	Version
ajaxmin	5.14.5506
codemirror	5.44.0
crypto-js	3.1.2
date	1.0 Alpha-1
jquery	3.5.1

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Library	Version
jquery.dotdotdot	1.7.4
jshint	2.10.2
jsonpath	1.0.0
jszip	3.1.5
kendo UI	2020.3.1118
knockout	3.5.1
knockout-kendo	0.10.0
paho-mqtt	1.1.0
pako-deflate	1.0.10
spin	2.0.0
xml2json	1.1.5

AVEVA Work Tasks 2020

Library	Version
ajaxmin	5.14.5506
codemirror	5.44.0
crypto-js	3.1.2
date	1.0 Alpha-1
jquery	3.4.1
jquery.dotdotdot	1.7.4
jshint	2.10.2
jsonpath	1.0.0
jszip	3.1.5
kendo UI	2019.1.220
knockout	3.4.2
knockout-kendo	0.9.8

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Library	Version
paho-mqtt	1.1.0
pako-deflate	1.0.10
spin	2.0.0
xml2json	1.1.5

AVEVA Workflow Management 2019 Update 1

Library	Version
ajaxmin	5.14.5506
codemirror	5.44.0
crypto-js	3.1.2
date	1.0 Alpha-1
jquery	3.3.1
jquery.dotdotdot	1.7.4
jshint	2.10.2
jsonpath	1.0.0
jszip	3.1.5
kendo UI	2019.1.220
knockout	3.4.2
knockout-kendo	0.9.8
paho-mqtt	1.1.0
pako-deflate	1.0.10
spin	2.0.0
xml2json	1.1.5

Workflow Management 2019

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Library	Version
ajaxmin	5.14.5506
codemirror	5.44.0
crypto-js	3.1.2
date	1.0 Alpha-1
jquery	3.3.1
jquery.dotdotdot	1.7.4
jshint	2.10.2
jsonpath	1.0.0
jszip	3.1.5
kendo UI	2019.1.220
knockout	3.4.2
knockout-kendo	0.9.8
paho-mqtt	1.1.0
pako-deflate	1.0.10
spin	2.0.0
xml2json	1.1.5

Skelta BPM 2017 R2 U2

Library	Version
ajaxmin	5.14.5506
codemirror	5.36.0
crypto-js	3.1.2
date	1.0 Alpha-1
jquery	3.3.1
jquery.dotdotdot	1.7.4
jshint	2.9.5

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Library	Version
jszip	3.1.5.0
kendo.all	2018.1.221
knockout	3.4.2
knockout-kendo	0.9.8
paho-mqtt	1.1.0
pako-deflate	1.0.6
spin	2.0.0
xml2json	1.1.5

Skelta BPM 2017 R2 U1

Library	Version
ajaxmin	5.14.5506
codemirror	5.36.0
crypto-js	3.1.2
date	1.0 Alpha-1
jquery	3.3.1
jquery.dotdotdot	1.7.4
jshint	2.9.5
jszip	3.1.5.0
kendo.all	2018.1.221
knockout	3.4.2
knockout-kendo	0.9.8
pako-deflate	1.0.6
spin	2.0.0
xml2json	1.1.5

Skelta BPM 2017 R2

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Library	Version
codemirror	5.33
crypto-js	3.1.2
date	1.0 Alpha-1
jquery	1.9.1
jquery.dotdotdot	1.7.4
jshint	2.9.1
jszip	2.5.0
kendo.all	2017.1.223
knockout	3.4.0
knockout-kendo	0.9.5
pako-deflate	1.0.0
spin	2.0.0
xml2json	1.1.5

Wonderware Skelta BPM 2017

Library	Version
codemirror	5.12
crypto-js	3.1.2
date	1.0 Alpha-1
jquery	1.9.1
jquery.dotdotdot	1.7.4
jshint	2.9.1
jszip	2.5.0
kendo.all	2017.1.223
knockout	3.4.0
knockout-kendo	0.9.5

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Library	Version
pako-deflate	1.0.0
spin	2.0.0
xml2json	1.1.5

Wonderware Skelta BPM 2014 R2 SP2 P01

Library	Version
codemirror	5.12
crypto-js	3.1.2
date	1.0 Alpha-1
jquery	1.9.1
jquery.dotdotdot	1.7.4
jshint	2.9.1
jszip	2.5.0
kendo.all	2016.1.226
knockout	3.4.0
knockout-kendo	0.9.5
pako-deflate	1.0.0
spin	2.0.0
xml2json	1.1.5

Wonderware Skelta BPM 2014 R2 SP2

Library	Version
date	1.0 Alpha-1
jquery	1.9.1
kendo.all	2015.2.624
knockout	3.3.0

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Library	Version
knockout-kendo	0.9.5
xml2json	1.1.5

Wonderware Skelta BPM 2014 R2 SP1

Library	Version
date	1.0 Alpha-1
jquery	1.9.1
kendo.all	2014.3.1316
knockout	3.1.0
knockout-kendo	0.8.1
xml2json	1.1.5

Wonderware Skelta BPM 2014 R2

Library	Version
date	1.0 Alpha 1
jquery	1.9.1
kendo.all	2014.3.1316
knockout	3.1.0
knockout-kendo	0.8.1
xml2json	1.1.5

Wonderware Skelta BPM 2014

Library	Version
jquery	1.9.1
kendo.all	2014.1.528
knockout	3.1.0

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Library	Version
knockout-kendo	0.8.1
xml2json	1.1.5

FAQs

Can I data bind the Grid Form using SOA?

No. However, using SOA, you can customize this functionality.

As soon as I submit a form, can I navigate to a different form to fill more details?

There is a feature called synchronization of forms which will redirect the user to a different form on submission of a form. The new form is a part of the workflow's Invoke Form Activity.

Lists

This section provides information on how to work with lists.

- Create and Organize
- Import and Export
- Design and Preview
- Manage and Secure

Introduction

Workflow Management Lists is a powerful platform to design and use functional lists in your enterprise. The lists platform uses the ASP.NET framework, and is developed for a rich user experience. You can connect to lists with various APIs from an external environment, and associate various events to workflow activities.

Key Capabilities	Workflow Enhancements
• Web based lists • Cross-browser support • Out of the box controls • Drag and drop support • Lookup infrastructure • Dynamic list forms • Reusable	• Paperless process management • Invoke workflow processes • Lists events association • Route lists between roles and between users • Mail and mobile notifications • Audit trails

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Access Lists

Use the **Manage List** page to create, organize, import, save, design, manage, and secure lists.

To access Manage List

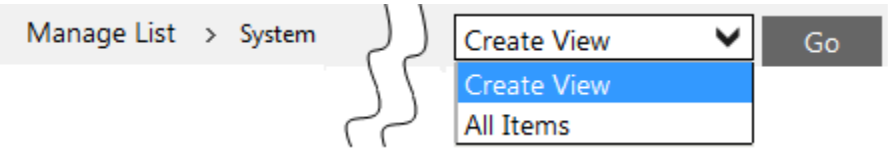
1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. The **Manage List** page shows the title, description, owner, and last modified date for the lists.

Manage List Interface

Manage List interface has a breadcrumbs, an action bar, and a list of lists.

Breadcrumbs

Breadcrumbs display the path to the child list or folder, and has commands to customize and view the list of lists.

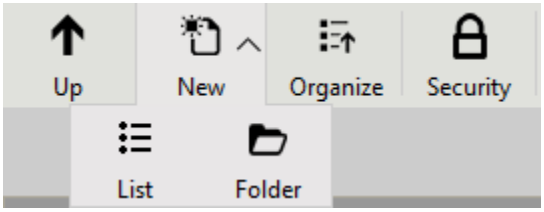


Command	Description
All Items	Default view to display the list of lists.
Create View	Create a new view to display the list of lists.

Action Bar

Action bar has commands to manage lists and folders. Standard commands are available all the time whereas contextual commands are available only when a list is selected.

Standard Commands



Menu/Command	Command	Description
Up		Go back to the parent list or folder.
New	List	Create a new list. For more information, see Create List.
	Folder	Create a new folder to group and organize lists.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

		For more information, see Creating Folders for Lists.
Organize		Organize the list by moving the list to a folder or by making the list a child item of another list. For more information, see Organizing Lists.
Security		View or customize the security level for the lists. For more information, see Set Security of Lists.

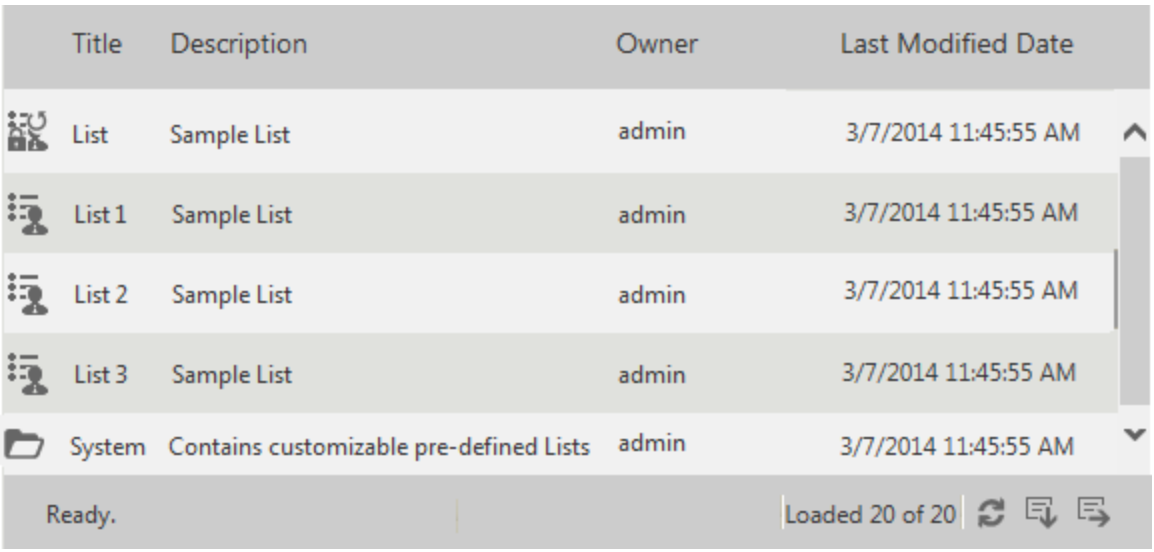
Contextual Commands

Command	Description
Design	Design the list with the list designer. For more information, see Design Lists.
Edit	Edit description of the list and then change design of the list with the list designer. For more information, see Editing Lists.
View	Preview the list. For more information, see Previewing Lists.
Delete	Delete the list. For more information, see Deleting Lists.
Change Owner	Change owner of the list. For more information, see Changing Ownership of Lists.

List

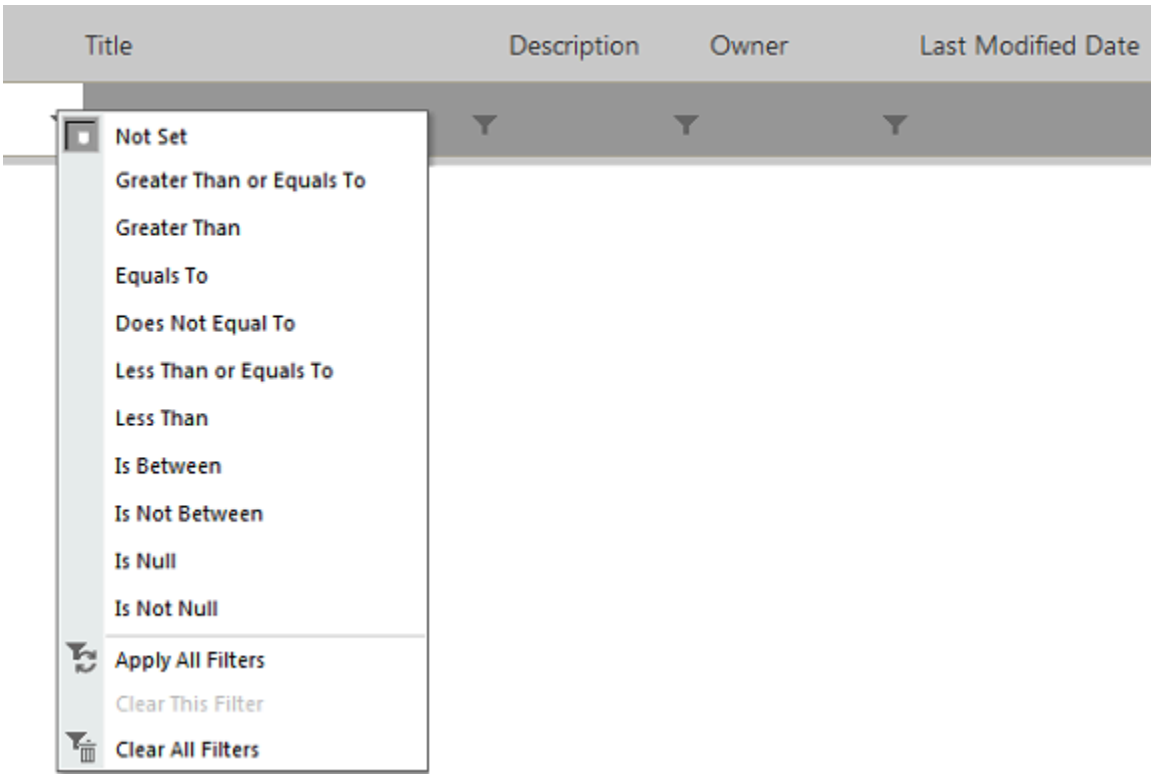
List of lists shows the title, description, owner, and last modified date for the lists. Click on a column to sort the lists.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Filter Bar

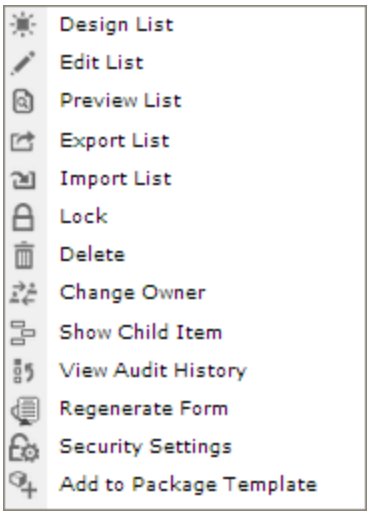
Filter bar can help to quickly find the desired lists from the list of lists. Filter bar has a funnel button and a parameter area for every column of the list of lists. The filter button shows the available filter options and the parameter area is for entering the filter criteria. The filter bar is hidden by default. Right-click on the column header, and then click **Filter Bar**, to access the filter bar. For more information about filter bar, see Filter Lists.



List Shortcut Menu

List shortcut menu has commands to manage the list. The list of commands available is dynamic and changes when the list is locked or disabled. Right-click on a list to access the shortcut menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Command	Description
Design List	Design the list with the list designer. For more information, see Design Lists.
Edit List	Edit description of the list and then change design of the list with the list designer. For more information, see Editing Lists.
Preview List	Preview the list. For more information, see Previewing Lists.
Export List	Export the list as an XML document. For more information, see Exporting Lists.
Import List	Import the list from an XML document. For more information, see Importing Lists. **Note:** Table or table schema associated with the list is not imported.
Lock	Lock the list. For more information, see Locking Lists.
Unlock	Unlock the list. For more information, see Unlocking Lists.
Delete	Delete the list. For more information, see Deleting Lists.
Change Owner	Change owner of the list. For more information, see Changing Ownership of Lists.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Show Child Item	Show child items of the list. For more information, see Viewing Child Items of Lists.
View Audit History	View audit history of the list. For more information, see Viewing Audit History of Lists.
Regenerate Form	Regenerate the form associated with the list. For more information, see Regenerating Forms. **Note:** When you regenerate a form, customizations previously made to the form is lost.
Security Settings	View or customize the security level for the lists. For more information, see Set Security of Lists.
Add to Package Template	Add the list to a package template.

Create and Organize

Map XML Nodes

XML node binds a database to a list, and maps a list to a form, and a form to a workflow. This ensures the data submitted from the form is available in the list and the database.

The default value for the **Mapped XML Node** property is the value for the **Tag name** property without the spaces. If you edit this value later, then it might affect the mapping between lists, forms, and workflows.

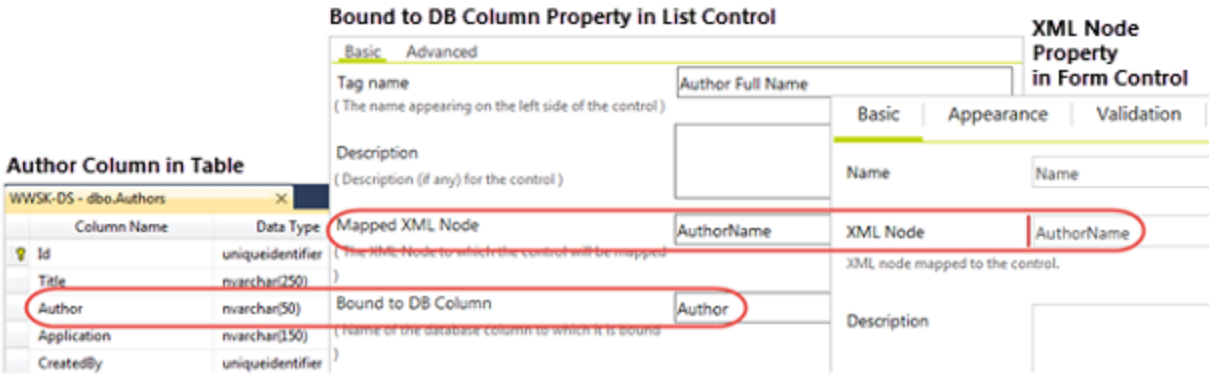
Mapping between a database, list, and form

When creating a list, you bind a control in the list to a field in the table of a database by setting the value of the field as the value of the **Bound to DB Column** property for the control in the list. Similarly, when creating a form, you should map a control in the form to a control in a list by setting the value of the **Mapped XML Node** property for the control in the list as the value of the **XML Node** property for the control in the form. When a dynamic list form is generated, the system takes care of this mapping.

Example

Here is an example showing the mapping between a database, list, and form using the database field and the XML Node.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Mapping	Description
Field of the table to control of the list	The **Author Full Name** control in the **Book Author** list is bound to the **Author** field in the **WWSK-DS-dbo.Authors** table of the repository database by setting **Author** (which is the value of the field) as the value of the **Bound to DB Column** property for the **Author Full Name** control in the **Book Author** list.
Control of the list to control of the form	The **Name** control in the **Writer** form is mapped to the **Author Full Name** control in the **Book Author** list by setting **AuthorName** (which is the value of the Mapped XML Node) as the value of the **XML Node** property for the **Name** control in the **Writer** form.

Create List

Create a list to use with forms and workflows.

To create a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. On the action bar, click **New**, and then click **List**. The **New List** wizard appears.
3. In the **Title** box, type a name for the list, and in the **Description** box, type a description for the list.
4. Do one of the following:

Task	Action
• Enable folder support for the list to group and organize the list.	• Under **Enable folder support?**, click **Yes**.
• Disable folder support for the list.	• Under **Enable folder support?**, click **No**.

5. Click **Save & Continue**. The list is created, and the **Select Table** wizard appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Do one of the following:

Task	Action
• Store the list data in generic table.	• Select **Generic Table**. **Note:** The list fields are automatically mapped to the fields in the generic table.
• Store the list data in an existing table.	1. Select **CustomList Table**. 2. Select a table from the list of existing tables. **Note:** - Table name and column name must not contain any spaces or special characters. - Click the **Refresh** icon to refresh the list of existing tables.
• Create a new table and add columns to it.	1. Select **CustomList Table**. 2. Click icon. The **Create Table** properties dialog box opens. 3. In the **Table Name** box, enter the name for the table. 4. To add column, in the **Column Name** box, enter the name for the column. The column name should not be the Reserved list column name. 5. Select the **Data Type** and the corresponding **Length** and **Scale** if applicable. 6. For **Allow Nulls** option, select **Yes** if you want to pass null values or else select **No**. 7. Click **Save**. The table gets saved to the database. **Note:** - Table name and column name must not contain any spaces or special characters. - Table should not have Reserved list column names. - Table can have maximum 64 columns apart from Reserved List columns. - You cannot create a new Folder in Custom List, which is mapped to a table with **Allow Nulls** option set to **No** for column(s).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
• Edit an existing table to add the columns.	1. Select **CustomList Table**. 2. Select a table from the list of existing tables. 3. Click icon to add the columns in the table. 4. Click **Save** after adding new column(s). **Note:** - Table name and column name must not contain any spaces or special characters. - Column name should not start with the number. - Table should not have Reserved list column names. - Table can have maximum 64 columns apart from Reserved List columns. - You cannot delete the existing column in a table. - You can edit the Existing table for a newly created list before clicking **Submit** button from the List designer. - Click **Refresh** icon to refresh the list of existing tables. - You cannot create a new Folder in Custom List, which is mapped to a table with **Allow Nulls** option set to **No** for column(s).

7. Click **Save & Continue**. The list is created, the **New List** wizard is closed, and the **Table** wizard appears.
8. Design the list, and then click **Submit**. The list is saved and the **Form Settings** dialog box appears.
9. Follow the steps for *To auto generate the form* in Auto Generating Forms.

Note:
- If an existing custom table from the repository is deleted from the database, the table name still appears in CustomList table List Box. Mapping table entry in SKListTables must be deleted to delist from CustomList table List box.

Create Child List Items

Every list item can have child items.

For example: For a workflow, the child items are either child workflows or folders.

To create child items for list items

1. Select a list item for which you want to create a child item.
2. Right-click and then select **Show Child Item**.
3. Click **New** and then create the child item.

   The child items are created successfully.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create Folders for Lists

Create a folder for a list to group and organize the list.

To create a folder for a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. On the action bar, click **New**, and then click **Folder**. The **New Folder** wizard appears.
3. In the **Title** box, type a name for the folder, and in the **Description** box, type a description for the folder.
4. Click **Save & Close**. The folder is created, and an information message appears.
5. Click **OK** (or click the **Close** button). The information message and the **New Folder** wizard are closed.

Organize Lists

Organize a list by moving the list to a folder or by making the list a child item of another list.

To organize a list by moving the list to a folder

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. On the action bar, click **Organize**. The **Organize Manage List** dialog box appears.
3. Drag a list to a folder. The list is moved to the folder.
4. Click **Close**. The **Organize Manage List** dialog box is closed.

To organize a list by making the list a child item of another list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. On the action bar, click **Organize**. The **Organize Manage List** dialog box appears.
3. Drag a list to another list. The list is now a child item of another list.
4. Click **Close**. The **Organize Manage List** dialog box is closed.

View Child Items of Lists

View child items of a list to see lists organized under the list.

To view child items of a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Show Child Item**. The child items page for the list appears.
3. The child items page for the list shows the available child lists.

Note: To go back to the **Manage List** page, on the action bar, click **Up**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Manage Data

Manage data by creating, exporting, and importing Excel Template with List Columns for both CustomList and Generic List table. The template has three rows - **Column Name**, **Data Type**, and **Size**.

To create and import List Template

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. On the action bar, click **Manage Data**.

   Note: The **Manage Data** option is available only to the users who are part of Standard Administrators Security Group.
3. Do one of the following:

Task	Action
• Create a schema of the table in the form of Excel Template.	• Click **Create List Template** from the Ribbon Bar. Excel template gets created with three rows (Column Name, Data Type, Size) where, **Column Name** is the **Mapped Xml node** in the List Designer if it is changed. Otherwise, it is the Column Name from the table. Excel Template gets downloaded in the **Downloads** folder with the filename **ListName_DateTime.xlsx**.
• Import list data in bulk through the Excel Template that was created. **Note**: - Before importing, fill data in the Excel Template created using **Create List Template** option. - Ensure that **Date** and **Time** value is in server or local culture and Boolean field has the value as **True** or **False**.	a. Click **Import List Data** from the Ribbon Bar. b. Select the file to be imported i.e. Excel Template file filled with list data. c. Click **Import** to insert the data into the list. The list data gets imported in bulk in the form of Excel Template with three rows (Column Name, Data Type, and Size).
• Export list items to the Excel sheet to export the list data in bulk.	• Click **Export List Data** from the Ribbon Bar. Excel template gets created with three rows (Column Name, Data Type, Size) with the list data. where, **Column Name** is the **Mapped Xml node** in the List Designer if it is changed. Otherwise, it is the Column Name from the table. Excel Template gets downloaded in the **Downloads** folder with the filename **ListName_DateTime.xlsx**. Exported List Items to the Excel will be sorted

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
	based on the LastUpdatedDateTime Column.

Note:
- The **Create List Template** and **Import List Data** options support the Lists with Parent table only. You can upload up to 5000 records in the List Template excel file using **Import List Data** option.
- You need to provide data in all the **Columns** which are marked as mandatory in the excel template.
- Ensure that you do not have the duplicate values for the primary key columns while filling the data in Excel Template.
- Modifying **Column Name** or adding new columns in the Excel Template is not supported.
- There is no validation of the data entered in Excel. Hence, make sure the value is entered as per the datatype mentioned.
- Attachment Control in the list is not supported while exporting or importing the list data.

Import and Export

Import Lists

Import a list to create the list from an XML document.

The XML schema defines the controls and properties of the controls for the list.

To import a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Import List**. The **Import** dialog box appears.
3. Click **Browse**. The **Choose File to Upload** dialog box appears.
4. Select an XML document, and then click **Open**. The XML document is selected and the **Choose File to Upload** dialog box is closed.
5. Click **Import**. The list is imported and an information message appears.
6. Click **Close**. The **Import** dialog box is closed.

Note:
When a list is imported:
- The table or table schema associated with the list is not imported.
- The existing controls and properties of the controls for the list are overwritten with the controls and properties of the controls defined in the XML schema.
- Ensure the repository has the same data provider as the repository from which the list was exported.

Export Lists

Export a list to save the list as an XML document.

The XML schema defines the controls and properties of the controls for the list. After successfully exporting a list, you can import the XML document to create similar lists.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To export a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Export List**. The **Export** dialog box appears.
3. Click **Export**. The **Save As** dialog box appears.
4. Click **Save**. The list is exported as an XML document, and the **Save As** dialog box is closed.

Design and Preview

This section provides information on designing, editing, deleting, and previewing list.

- Design Lists
- Edit Lists
- Delete Lists
- Preview Lists

Design Lists

Design a list to add controls to the list.

To design a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on list, and then click **Design**). The **List Designer** page appears.
3. Design the list, and then click **Submit**. The list is saved and the **Form Settings** dialog box appears.
4. Follow the steps for *To auto generate the form* in Auto Generating Forms..

Edit Lists

Edit a list to change description and design of the list.

To edit a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Edit List** (or click on a list, and then click **Edit**). The **Edit List** dialog box appears.
3. In the **Description** box, edit description for the list.
4. Click **Save & Continue**. The description for the list is saved, the **Edit List** dialog box is closed, and the **List Designer** page appears.
5. Design the list, and then click **Submit**. The list is saved and the **Form Settings** dialog box appears.
6. Follow the steps for *To auto generate the form* in Auto Generating Forms.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: You cannot edit the name of the list. To edit properties of a list, see Editing Properties of Lists.

Delete Lists

Delete a list when the list is no longer required.

To delete a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Delete** (or click on a list, and then click **Delete**). A confirmation message appears.
3. Click **OK**. The confirmation message is closed, the list is deleted, and an information message appears.
4. Click **Close** (or click the **Close** button). The information message is closed.

Note: If you delete a list, then the list cannot be restored.

Preview Lists

Preview a list to check look and feel of the list at run-time and the XML generated for the list.

To preview only the list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Preview List** (or click on a list, and then on the action bar, click **View**). The **Preview** dialog box appears.
3. Check look and feel of the list.
4. Click the **Close** button. The **Preview** dialog box is closed.

To preview the list and the XML generated for the list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Preview List** (or click on a list, and then on the action bar, click **View**). The **Preview** dialog box appears.
3. Check look and feel of the list, and then click **Submit**. The **Instance XML** dialog box appears.
4. Check the generated XML, and then click the **Close** button. The **Instance XML** dialog box is closed.
5. Click the **Close** button. The **Preview** dialog box is closed.

Manage and Secure

This section explains you on how to manage and secure the lists.

- Lock Lists
- Unlock Lists

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Filter Lists
- Change Ownership of Lists
- Set Security of Lists
- View Audit History of Lists
- Regenerate Forms

Delete List Items

You can delete multiple list items.

.

To delete list item(s)

1. In the **Enterprise Console**, click the **Menu** button, click **List**, and then select the **Custom List** from which you want to delete the list items or folders.

   Note: **List** option is available under **Menu** if you have created custom lists.
2. To delete List Items or Folders, you can either,
   - Select ALL check box to delete all the list items.
   - Select a list item check box to delete that particular list item.
3. Click **Delete** icon on the ribbon bar. All the selected list items will get deleted.

   Note:
   - If the List Item or Folder has Child Items, that List Item or Folder will not get deleted. You need to first delete the child List Item.

Lock Lists

Lock a list to prevent other users from editing the list.

To lock a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The Manage List page appears.
2. Right-click on a list, and then click **Lock**. The list is locked and an information message appears.
3. Click **OK** (or click the Close button). The information message is closed.

   Note: When a list is locked, a lock overlay icon appears on the icon of the list.

Unlock Lists

Unlock a list to allow other users to edit the list.

To unlock a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a locked list, and then click **Unlock**. The list is unlocked and an information message appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Click **OK** (or click the Close button). The information message is closed.

Note: When a list is unlocked, the lock overlay icon disappears from the icon of the list.

Filter Lists

Filter lists to quickly find the lists you are looking for from the list of lists.

Filtering lists is useful especially when there are a large number of lists.

To apply a filter for the lists

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click the column header, and then click **Filter Bar**. The filter bar appears.
3. Enter a value in the filter bar for a column, and then press Enter.
4. Click the **Funnel** button on the column, and then click a filter option. The filter option is selected.
5. Click **Apply All Filters**. The filtered result appears.

To clear a filter for the lists

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click the filter bar (or click the **Funnel** button on the column), and then click **Clear All Filters**. The filtered result is cleared.

Filter Options

Filter lists using one of the filter option for each column:

Option	Description
Not Set	Show lists without applying any filters. This is the default filter option.
Greater Than or Equal To	Show lists where the value of the column is greater than or equal to the specified value.
Greater Than	Show lists where the value of the column is greater than the specified value.
Equal To	Show lists where the value of the column is equal to the specified value.
Does Not Equal To	Show lists where the value of the column is not equal to the specified value.
Less Than or Equal To	Show lists where the value of the column is less than or equal to the specified value.
Less Than	Show lists where the value of the column is less than

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Option	Description
	the specified value.
Is Between	Show lists where the value of the column is between the specified inclusive range of values.
Is Not Between	Show lists where the value of the column is not between the specified inclusive range of values.
Like	Show lists where the value of the column is as per the specified value.
Does Not Like	Show lists where the value of the column is not as per the specified value.
Contains	Show lists where the value of the column contains the specified value.
Does Not Contain	Show lists where the value of the column does not contain the specified value.
Is Null	Show lists where the value of the column is null.
Is Not Null	Show lists where the value of the column is not null.
Is Empty	Show lists where the value of the column is an empty string.
Is Not Empty	Show lists where the value of the column is not an empty string.
Apply All Filters	Apply all filters set for the columns.
Clear This Filter	Clear all filters set for a column.
Clear All Filters	Clear all set filters.

Change Ownership of Lists

Change owner of a list to allow other user to own and control the list.

To change owner of a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Change Owner** (or click on a list, and then on the action bar, click **Change Owner**). The **Change Owner** dialog box appears.
3. Click the **User Lookup** button. The **User Lookup** dialog box appears.
4. Select a user, and then click **Update**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Click **Save**. The owner of the list is changed and an information message appears.
6. Click **OK** (or click the **Close** button). The information message and the **Change Owner** dialog boxes are closed.

Set Security of Lists

Secure lists by defining rights assigned to a list or a set lists grouped under a folder.

A lists user or role inherits the rights assigned at a repository level. However, the administrator can customize the rights at a list level and also assign rights to groups and map users to groups.

Standard Settings

Use standard settings to view security set at the list level, if available, or at the repository level for:

- a role or user.
- the security group customization and resource mapping.

To view security set for a role or user

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. On the action bar, click **Security**. The **Security Settings** page appears.
3. In the **Standard settings** tab, click **View Effective Security For User/Role**.
4. Do one of the following:

Task	Action
• View security set at the list level, if available, or at the repository level for a role.	1. In the **Role Based** tab, click the **Role Lookup** button. The **Role Lookup** dialog box appears. 2. Select a role, and then click **Update**. 3. Click **View Rights**. The effective rights for the role appears.
• View security set at the list level, if available, or at the repository level for a user.	1. Click the **User Based** tab. 2. Click the **User Lookup** button. The **User Lookup** dialog box appears. 3. Select a user, and then click **Update**. 4. Click **View Rights**. The effective rights for the user appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To view security group customization and resource mapping

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. On the action bar, click **Security**. The **Security Settings** page appears.
3. In the **Standard settings** tab, click **View Security Mapping at List Level**.
4. Click on a **Security Group**. The **Viewing Security Group Rights** dialog box appears.
5. Check the security group customization and resource mapping.
6. Click **Close** (or click the **Close** button). The **Viewing Security Group Rights** dialog box is closed.

Advanced Settings

Use advanced settings to edit the existing mapping, add a new resource mapping or delete the existing mapping by enabling mapping at List level.

To set or edit the security group customization and resource mapping

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. On the action bar, click **Security**. The **Security Settings** page appears.
3. Click the **Advanced settings** tab, and then do one or more of the following:

Task	Action
• Set the security group customization and resource mapping.	• Click **Enable List Specific Security**. **Note:** This option is available only when the security group customization and resource mapping has not been set or removed.
• Remove the security group customization and resource mapping.	• Click **Remove Security from this List**.
• Edit the security group customization.	• Click **Customize Security Group for this List**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
• Edit the resource mapping.	• Click **Security Group - Resource Mapping for this List**.

View Audit History of Lists

View audit history of a list to see date, time and user information when the list was locked and unlocked, and when the list owner was changed.

To view audit history of a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **View Audit History**. The **Audit History** dialog box appears.
3. The **Audit History** dialog box shows the action, user, version, message, and date for the list.
4. Click the **Close** button. The **Audit History** dialog box is closed.

Regenerate Forms

Regenerate a form to automatically generate the form again to create a dynamic list form.

To regenerate a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Regenerate Form**. The **Form Settings** dialog box appears.
3. Do one of the following:

Task	Action
• Deny the system to regenerate the form.	a. Click the **No** option. b. Click **OK**. The **Form Settings** dialog box is closed.
• Allow the system to regenerate the form.	a. Click the **Yes** option. b. Click **OK**. The **Form Settings** dialog box is closed and the **Forms Designer** page appears. c. If you want to validate the auto generated form, then follow the steps for Validate Dynamic List Forms. d. Click the **Close** button. The form is saved and the **Forms Designer** page is closed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: When you regenerate a form, customizations previously made to the form is lost.

Lists Designer

List Designer is a web-based design tool, with a drag-and-drop capable interface, to create functional lists. Drag a control from the toolbox to the design area to add the control to the list. Dragging is also supported for repositioning and copying controls. Use **List Designer** to create functional lists with various controls.

To access List Designer

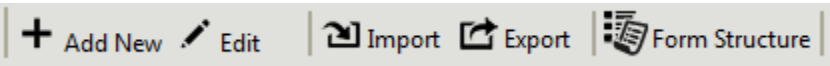
1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on a list, and then on the action bar, click **Design**). The **List Designer** page appears.

List Designer Interface

List Designer interface has an action bar, toolbox, and canvas.

Action Bar

Action bar has commands to manage the list and controls on the list.

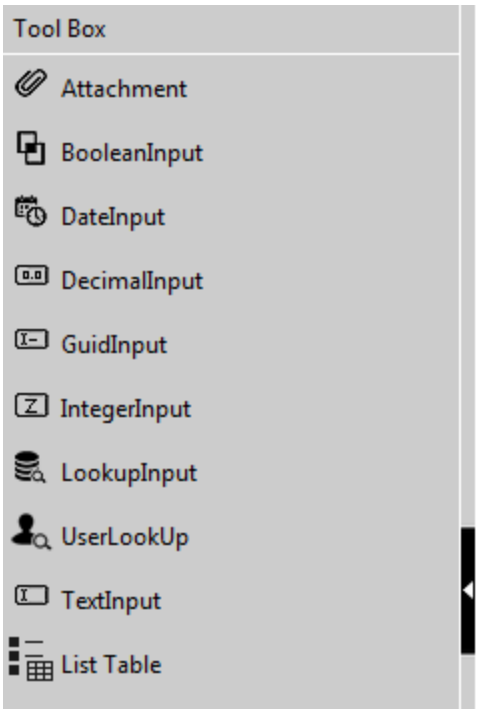


Command	Description
Add New	Add new control to the list. For more information, see Adding Controls.
Edit	Edit properties of the list. For more information, see Editing Properties of Lists.
Import	Import the list from an XML document. For more information, see Importing Lists.
Export	Export the list as an XML document. For more information, see Exporting Lists.
Form Structure	View the structure of the form associated with the list. For more information, see Viewing Form Structure.

Toolbox

Toolbox has controls for designing the list.

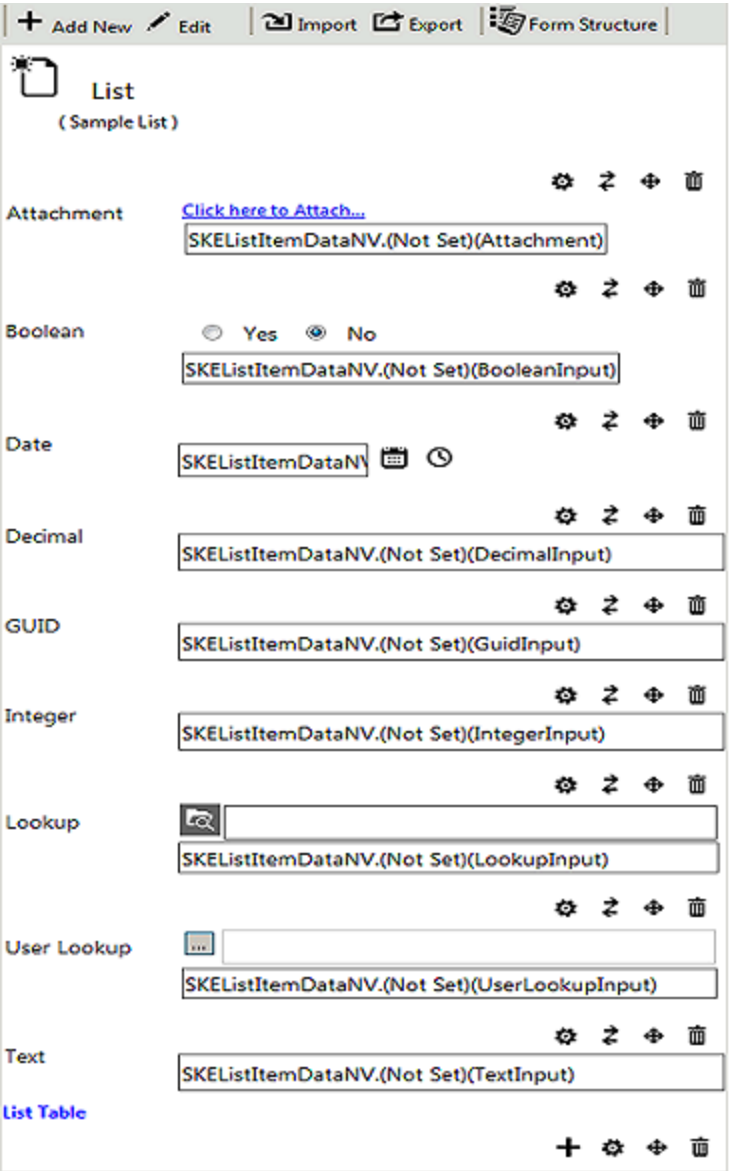
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Canvas

Canvas is the area in the **List Designer** where the list is designed by dragging controls from the toolbox. Dragging is also supported for repositioning controls.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Control Menu

Control menu has commands to manage the control. The menu is placed on the right side above the control.

Command	Description
Properties	Edit properties of the control. For more information, see Edit Properties of Controls.
Replace	Replace the control. For more information, see Replace Controls.
Move	Move the control to the desired location on the canvas.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Command	Description
	For more information, see Move Controls.
Delete	Delete the control. For more information, see Deleting Controls.

Note: To design a form or edit an existing form in **Lists Designer**, the Workflow Management Forms Server license has to be acquired.

Manage Lists

This section explains how you can import, export, edit properties, view form structure, and auto generate forms lists using List Designer.

- Import Lists
- Export Lists
- Edit Properties of Lists
- View Form Structure
- Auto Generating Forms

Import Lists

Import a list to create the list from an XML document.

The XML schema defines the controls and properties of the controls for the list.

To import a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on a list, and then on the action bar, click **Design**). The **List Designer** page appears.
3. On the action bar, click **Import**. The **Import** dialog box appears.
4. Click **Browse**. The **Choose File to Upload** dialog box appears.
5. Select an XML document, and then click **Open**. The XML document is selected and the **Choose File to Upload** dialog box is closed.
6. Click **Import**. The list is imported and an information message appears.
7. Click **Close**. The **Import** dialog box is closed.

Note:
When a list is imported:
- The table or table schema associated with the list is not imported.
- The existing controls and properties of the controls for the list are overwritten with the controls and properties of the controls defined in the XML schema.
- Ensure the repository has the same data provider as the repository from which the list was exported.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Export Lists

Export a list to save the list as an XML document.

The XML schema defines the controls and properties of the controls for the list. After successfully exporting a list, you can import the XML document to create similar lists.

To export a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on a list, and then on the action bar, click **Design**). The **List Designer** page appears.
3. On the action bar, click **Export**. The **Save As** dialog box appears.
4. Click **Save**. The list is exported as an XML document, and the **Save As** dialog box is closed.

Edit Properties of Lists

Edit properties of a list to change appearance and functionality of the list.

To edit properties of a list

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on a list, and then on the action bar, click **Design**). The **List Designer** page appears.
3. On the action bar, click **Edit**. The **Properties** dialog box appears.
4. Edit properties of the list, and then click **Save**. The properties are saved and the **Properties** dialog box is closed.

View Form Structure

View structure of the form associated with a list to check the structure of the form and to save the form as an XML document.

To view structure of the form

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on a list, and then on the action bar, click **Design**). The **List Designer** page appears.
3. On the action bar, click **Form Structure**. The **Form Structure** dialog box appears.
4. Do one of the following:

Task	Action
• Include controls with blank XML node.	• Under **Ignore blank XmlNodeBoundTo ?**,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
	click **Yes**.
• Exclude controls with blank XML node.	• Under Ignore blank **XmlNodeBoundTo ?**, click **No**.

5. Do one of the following:

Task	Action
• Export the form associated with the list.	a. Click **Export**. The **Save As** dialog box appears. b. Click **Save**. The list is exported as an XML document, and the **Save As** dialog box is closed. c. Click **Close**. The **Form Structure** dialog box is closed.
• Close the Form Structure dialog box.	• Click **Close**. The **Form Structure** dialog box is closed.

Auto Generating Forms

Automatically generate the form when you create a list to create a dynamic list form.

Follow the steps for To access List Designer, and then click **Submit.** The confirmation message appears.

To auto generate the form

1. In the confirmation message, click **CANCEL**. The form is generated and an information message appears.
2. Click **OK**. The **List Designer** page is closed.

To edit the auto generated form

1. In the confirmation message, click **OK** to edit the auto generated form. List is successfully saved, the form associated with this list is generated, and an information message appears..
2. Click **OK**. The **List Designer** page is closed and the **Forms Designer** page appears.
3. Design the form and Publish it, so that the new list can be used.

Note:
- The auto generated form is saved, with the same name as that of the list, in the **Dynamic List Forms** folder in the **Forms** page.
- The form associated with the list must be published before using the list.

Manage Controls

This section explains how you can add, replace, move, delete and edit properties of controls.

- Add Controls

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Replace Controls
- Move Controls
- Delete Controls
- Edit Properties of Controls

Add Controls

Add a control to the canvas to build the list.

To add a control from the toolbox

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on a list, and then on the action bar, click **Design**). The **List Designer** page appears.
3. Do one of the following:

Task	Action
• Place a control at the end of other controls on the canvas.	• Drag a control from the toolbox to the orange bar on the canvas.
• Place a control above a control on the canvas.	• Drag a control from the toolbox to the green bar of a control on the canvas.

4. The selected control appears on the canvas and the **Properties** dialog box appears
5. Edit properties of the control, and then click **Apply**. The properties are saved and the **Properties** dialog box is closed.

To add a control from the action bar

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on a list, and then on the action bar, click **Design**). The **List Designer** page appears.
3. On the action bar, click **Add New**. The **Add a new control** dialog box appears.
4. In the **Title** box, type a name for the control, and in the **Description** box, type a description for the control.
5. Click a control, and then click **Next**. The control appears at the end of other controls on the canvas, the **Add a new control** dialog box is closed, and the **Properties** dialog box appears.
6. Edit properties of the control, and then click **Apply**. The properties are saved and the **Properties** dialog box is closed.

Note: The name of the control should not be the same as the column names that are defined in the SKEListItemDataNV table in the Repository Database.

The below are the column names defined in the SKEListItemDataNV table in the Repository Database:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Id, Application, CreatedBy, CreatedDateTime, LastUpdatedDateTime, Title, Description, WIP, WIPItemId, MainItemId, ParentItemId, RecordParentItemId, TableId, ItemRowIndex, IsLatest, Version, VersionHistory, Status, LockedBy, LockedOn, ItemId, ItemType, SecurityItemId, Owner, ModifiedBy, IsDisabled, WorkflowStatus, SecurityCustomizationId, ListID, SecuritySettings, AuditTrail, nvarchar1, nvarchar2, nvarchar3, nvarchar4, nvarchar5, nvarchar6, nvarchar7, nvarchar8, ntext1, ntext2, ntext3, ntext4, sql_variant1, nvarchar9, nvarchar10, nvarchar11, nvarchar12, nvarchar13, nvarchar14, nvarchar15, nvarchar16, ntext5, ntext6, ntext7, ntext8, sql_variant2, nvarchar17, nvarchar18, nvarchar19, nvarchar20, nvarchar21, nvarchar22, nvarchar23, nvarchar24, ntext9, ntext10, ntext11, ntext12, sql_variant3, nvarchar25, nvarchar26, nvarchar27, nvarchar28, nvarchar29, nvarchar30, nvarchar31, nvarchar32, ntext13, ntext14, ntext15, ntext16, sql_variant4, nvarchar33, nvarchar34, nvarchar35, nvarchar36, nvarchar37, nvarchar38, nvarchar39, nvarchar40, ntext17, ntext18, ntext19, ntext20, sql_variant5, nvarchar41, nvarchar42, nvarchar43, nvarchar44, nvarchar45, nvarchar46, nvarchar47, nvarchar48, ntext21, ntext22, ntext23, ntext24, sql_variant6, nvarchar49, nvarchar50, nvarchar51, nvarchar52, nvarchar53, nvarchar54, nvarchar55, nvarchar56, ntext25, ntext26, ntext27, ntext28, sql_variant7, nvarchar57, nvarchar58, nvarchar59, nvarchar60, nvarchar61, nvarchar62, nvarchar63, nvarchar64, ntext29, ntext30, ntext31, ntext32, sql_variant8, int1, int2, int3, int4, int5, int6, int7, int8, int9, int10, int11, int12, int13, int14, int15, int16, decimal1, decimal2, decimal3, decimal4, decimal5, decimal6, decimal7, decimal8, decimal9, decimal10, decimal11, decimal12, datetime1, datetime2, datetime3, datetime4, datetime5, datetime6, datetime7, datetime8, bit1, bit2, bit3, bit4, bit5, bit6, bit7, bit8, bit9, bit10, bit11, bit12, bit13, bit14, bit15, bit16, uniqueidentifier1, uniqueidentifier2, uniqueidentifier3, uniqueidentifier4, uniqueidentifier5, uniqueidentifier6, uniqueidentifier7, uniqueidentifier8, uniqueidentifier9, uniqueidentifier10, uniqueidentifier11, uniqueidentifier12, uniqueidentifier13, uniqueidentifier14, uniqueidentifier15, uniqueidentifier16, uniqueidentifier17, uniqueidentifier18, uniqueidentifier19, uniqueidentifier20, uniqueidentifier21, uniqueidentifier22, uniqueidentifier23, uniqueidentifier24, uniqueidentifier25, image1, image2, image3, image4, image5, image6, image7, image8, image9, image10, image11, image12, image13, image14, image15

Replace Controls

Replace a control to change the control on the canvas.

To replace a control

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on a list, and then on the action bar, click **Design**). The **List Designer** page appears.
3. On the control, click the **Replace** icon. The **Replace Control** dialog box appears.
4. Click a control, and then do one of the following:

Task	Action
• Allow the system to copy the values of similar properties from the existing control to the new control.	• Under **Copy similar properties ?**, click **Yes**.
• Deny the system to copy the values of similar properties from the existing control to the new control.	• Under **Copy similar properties ?**, click **No**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Click **Replace**. The control is replaced on the canvas and the **Replace Control** dialog box is closed.

Move Controls

Move a control to reposition the control to the desired location on the canvas.

To move a control

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on a list, and then on the action bar, click **Design**). The **List Designer** page appears.
3. On the control, click the **Move** button, and then do one of the following:

Task	Action
• Place a control at the end of other controls on the canvas.	• Drag a control from the toolbox to the orange bar on the canvas.
• Place a control above a control on the canvas.	• Drag a control from the toolbox to the green bar of a control on the canvas.

4. The selected control appears at the desired location on the canvas.

Delete Controls

Delete a control when the control is no longer required.

To edit properties of a control

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on a list, and then on the action bar, click **Design**). The **List Designer** page appears.
3. On the control, click the **Delete** button. A confirmation message appears.
4. Click **Yes**. The control is deleted from the canvas and the confirmation message is closed.

Note: If you delete a control, then the control cannot be restored.

Edit Properties of Controls

Edit properties of a control to change appearance and functionality of the control.

To edit properties of a control

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Right-click on a list, and then click **Design List** (or click on a list, and then on the action bar, click **Design**). The **List Designer** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. On the control, click the **Properties** button. The **Properties** dialog box appears.
4. Edit properties of the control, and then click **Save**. The properties are saved and the **Properties** dialog box is closed.

Control Properties

The properties for a control are grouped under the following tabs depending on the control:

- Basic
- Appearance

For information about the properties of a control, see Controls.

Lists API

Application programming interface (API) is a set of functions to accomplish a specific task. APIs provide speed, reliability, and extensibility.

Use lists APIs to customize your business applications.

Create List using ListDefinition

Create a list using the list API.

The list API makes use of the following DLL references, namespaces, classes, and methods.

DLL Reference	Namespace
• Workflow.NET.NET2 • Skelta.Forms.Core • Skelta.Forms.Web	• System • Skelta.Repository.List • Skelta.Forms • Skelta.Forms.Core.Controls • Skelta.Core • Skelta.Repository.List.Action • System.Data

Class
• Skelta.Repository.List.ListDefinition • Skelta.Repository.List.ListItem • Skelta.Forms.Web.Common.ListFormDesignerFunctions • Skelta.Forms.Web.Common.DynamicListCommonFunctions

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Class
• Skelta.Forms.Core.Controls.ListMainForm • Skelta.Repository.List.Action.ListASPXUIAction • Skelta.Repository.List.Action.ActionDelete • Skelta.Repository.List.Action.ActionNewCategory • Skelta.Repository.List.Action.ListEventHandlerProvider • Skelta.Repository.List.ListTableCollection • Skelta.Core.DS.SkeltaParameter

Methods
• `ListItem.Save(ListItemVersionStatus)` This method saves a list item to a specific version. • `ListDefinition.Save()` This method saves a list.

Scenario

Create an Employee Details list mapped to an Employee table with the fields such as Title, Name, Age, Salary, Department, and Manager by running the following code.

```
CREATE TABLE [dbo].[Employee](
 [Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF__Employee__Id__1E6F845E] DEFAULT
(newid()),
 [Title] [nvarchar](250) NULL,
 [Name] [varchar](50) NULL,
 [Age] [int] NULL,
 [Salary] [money] NULL,
 [Department] [varchar](50) NULL,
 [Manager] [varchar](50) NULL,
 [Application] [nvarchar](150) NULL,
 [CreatedBy] [uniqueidentifier] NULL,
 [CreatedDateTime] [datetime] NULL,
 [LastUpdatedDateTime] [datetime] NOT NULL,
 [WIP] [bit] NULL,
 [WIPItemId] [uniqueidentifier] NULL,
 [MainItemId] [uniqueidentifier] NULL,
 [ParentItemId] [uniqueidentifier] NULL,
 [RecordParentItemId] [uniqueidentifier] NULL,
 [TableId] [uniqueidentifier] NULL,
 [ItemRowIndex] [int] NULL,
 [IsLatest] [bit] NULL,
 [Version] [nvarchar](25) NULL,
 [VersionHistory] [nvarchar](max) NULL,
 [Status] [int] NULL,
 [LockedBy] [uniqueidentifier] NULL,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 [LockedOn] [datetime] NULL,
 [ListID] [uniqueidentifier] NULL,
 [SecuritySettings] [nvarchar](max) NULL,
 [ItemId] [uniqueidentifier] NULL,
 [ItemType] [int] NULL,
 [SecurityItemId] [uniqueidentifier] NULL,
 [Owner] [uniqueidentifier] NULL,
 [ModifiedBy] [uniqueidentifier] NULL,
 [AuditTrail] [nvarchar](max) NULL CONSTRAINT [DF_Employee_AuditTrail] DEFAULT (''),
 [IsDisabled] [bit] NULL CONSTRAINT [DF_Employee_IsDisabled] DEFAULT ((0)),
 [WorkflowStatus] [varchar](50) NULL,
 [SecurityCustomizationId] [uniqueidentifier] NULL,
CONSTRAINT [PK_Employee] PRIMARY KEY NONCLUSTERED
(
[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
= ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [IX_AMI_Employee] ON [dbo].[Employee]
(
 [Application] ASC,
 [MainItemId] ASC,
 [Id] ASC
)WITH (PAD_INDEX  = OFF, STATISTICS_NORECOMPUTE  = OFF, SORT_IN_TEMPDB = OFF,
IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS  = ON,
ALLOW_PAGE_LOCKS  = ON) ON [PRIMARY]
Go
if Not exists (select * from SKListTables where Application='<RepositoryName>' And
TableName='Employee')
 Insert into SKListTables (Application,TableName) Values ('<RepositoryName>','Employee')
Go
using System;
using Skelta.Repository.List;
using Skelta.Forms;
using Skelta.Forms.Core.Controls;
using Skelta.Core;
using Skelta.Repository.List.Action;
using System.Data;
namespace ListObject
{
    public class DynamicList
    {
        static void Main(string[] args)
        {
            DynamicList DL = new DynamicList();
            DL.CreateDynamicList("SkeltaRepo", "admin", "Employee", "Employee",
"Employee");
        }
        public Guid CreateDynamicList(string applicationName, string LoggedInUser, string
listName, string listDescription, string tableName)
        {
 // Deleting the list if it already exists.
            ListDefinition _List = ListDefinition.GetList(new
Skelta.Core.ApplicationObject(applicationName), "Manage List");
            ListItem item = new Skelta.Repository.List.ListItem(_List,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
"<SkeltaDLList><EnableFolderSupport>True</EnableFolderSupport><Description></Description><T
itle>" + listName + "</Title></SkeltaDLList>");
            item.LoggedInUserId = LoggedInUser;
            item.Save(ListItemVersionStatus.Published);
            ListDefinition newList;
            newList = new ListDefinition(new
Skelta.Core.ApplicationObject(applicationName));
            newList.Definition = new BaseForm();
            newList.Name = listName;
            newList.ImageFileName = "skelta-DLItem-Item.png";
            newList.Definition = GetListDefinition(listName, listDescription, tableName);
            // Adding actions like new, edit, delete and so on.
            AddActions(newList, listName);
            newList.Save();
            // Adding list to the navigation menu.
            InsertListNavigation(applicationName, LoggedInUser, listName);
            // Generating form.
            // If you want to associate the list to an existing form, then the next two
lines are not required. Also update formItemId with the existing form ID.
            Skelta.Forms.Web.Common.ListFormDesignerFunctions lstFrmFns = new
Skelta.Forms.Web.Common.ListFormDesignerFunctions();
            Guid formItemId = lstFrmFns.SaveFormDefinition(applicationName, LoggedInUser,
listName);
            // Associating the list with the form.
            Skelta.Forms.Web.Common.DynamicListCommonFunctions dlCmnFns = new
Skelta.Forms.Web.Common.DynamicListCommonFunctions();
            dlCmnFns.AddListFormAssociation(new
Skelta.Core.ApplicationObject(applicationName), listName, newList.Id, formItemId);
            return newList.Id;
        }
 // Creating ListMainForm for adding new Employees.
        private ListMainForm GetListDefinition(string listName, string listDescription,
string tableName)
        {
            ListMainForm mainForm = new ListMainForm();
            mainForm.IsTableVersioned = false;
            mainForm.Id = "_sys_" + listName.ToLower() + "_list" + "_skeltadynamiclist";
            mainForm.UniqueTitle = false;
            mainForm.XmlNodeBoundTo = listName;
            mainForm.Name = listName;
            mainForm.IsFolderSupported = true;
            mainForm.TableName = tableName;
            mainForm.Description = listDescription;
            ListTextDataItem txtInput = new ListTextDataItem();
            txtInput.Name = "Title";
            txtInput.Id = "_sys_ListForms_" + txtInput.Name + Guid.NewGuid().ToString("N");
            txtInput.GridCaption = "Title";
            txtInput.BoundToTableField = "Title";
            txtInput.BoundToTableFieldType = typeof(string);
            txtInput.XmlNodeBoundTo = "Title";
            txtInput.DisplayInLookup = true;
            txtInput.IsLookupTitle = true;
            txtInput.GridDisplayWidth = 150;
            txtInput.GridDisplayIndex = 100;
            txtInput.AllowSorting = true;
            mainForm.Controls.Add(txtInput);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
            txtInput = new ListTextDataItem();
            txtInput.Name = "Name";
            txtInput.Id = "_sys_ListForms_" + txtInput.Name + Guid.NewGuid().ToString("N");
            txtInput.GridCaption = "Name";
            txtInput.BoundToTableField = "Name";
            txtInput.BoundToTableFieldType = typeof(string);
            txtInput.XmlNodeBoundTo = "Name";
            txtInput.DisplayInLookup = true;
            txtInput.IsLookupTitle = true;
            txtInput.GridDisplayWidth = 150;
            txtInput.GridDisplayIndex = 100;
            txtInput.AllowSorting = true;
            mainForm.Controls.Add(txtInput);
            ListIntDataItem intInput = new ListIntDataItem();
            intInput.Name = "Age";
            intInput.Id = "_sys_ListForms_" + intInput.Name + Guid.NewGuid().ToString("N");
            intInput.GridCaption = "Age";
            intInput.BoundToTableField = "Age";
            intInput.BoundToTableFieldType = typeof(int);
            intInput.XmlNodeBoundTo = "Age";
            intInput.IsLookupTitle = true;
            intInput.GridDisplayWidth = 150;
            intInput.GridDisplayIndex = 100;
            intInput.AllowSorting = true;
            mainForm.Controls.Add(intInput);
            intInput = new ListIntDataItem();
            intInput.Name = "Salary";
            intInput.Id = "_sys_ListForms_" + intInput.Name + Guid.NewGuid().ToString("N");
            intInput.GridCaption = "Salary";
            intInput.BoundToTableField = "Salary";
            intInput.BoundToTableFieldType = typeof(int);
            intInput.XmlNodeBoundTo = "Salary";
            intInput.IsLookupTitle = true;
            intInput.GridDisplayWidth = 150;
            intInput.GridDisplayIndex = 100;
            intInput.AllowSorting = true;
            mainForm.Controls.Add(intInput);
            txtInput = new ListTextDataItem();
            txtInput.Name = "Department";
            txtInput.Id = "_sys_ListForms_" + txtInput.Name + Guid.NewGuid().ToString("N");
            txtInput.GridCaption = "Department";
            txtInput.BoundToTableField = "Department";
            txtInput.BoundToTableFieldType = typeof(string);
            txtInput.XmlNodeBoundTo = "Department";
            txtInput.DisplayInLookup = true;
            txtInput.IsLookupTitle = true;
            txtInput.GridDisplayWidth = 150;
            txtInput.GridDisplayIndex = 100;
            txtInput.AllowSorting = true;
            mainForm.Controls.Add(txtInput);
            txtInput = new ListTextDataItem();
            txtInput.Name = "Manager";
            txtInput.Id = "_sys_ListForms_" + txtInput.Name + Guid.NewGuid().ToString("N");
            txtInput.GridCaption = "Manager";
            txtInput.BoundToTableField = "Manager";
            txtInput.BoundToTableFieldType = typeof(string);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
            txtInput.XmlNodeBoundTo = "Manager";
            txtInput.DisplayInLookup = true;
            txtInput.IsLookupTitle = true;
            txtInput.GridDisplayWidth = 150;
            txtInput.GridDisplayIndex = 100;
            txtInput.AllowSorting = true;
            mainForm.Controls.Add(txtInput);
            return mainForm;
        }
        private void AddActions(ListDefinition newList, string listName)
        {
            ListASPXUIAction newListItem = new ListASPXUIAction();
            newListItem.Id = "_sys_new";
            newListItem.Name = "New " + listName;
            newListItem.ShowWhenSecurityAllowsRights = "";
            newListItem.ValidOnlyForItemStatus = "";
            newListItem.ValidOnlyForItemTypes = "1";
            newListItem.PagePath = "DynamicListNewForm.aspx";
            newListItem.Type = ListActionType.New;
            newListItem.IconPath = "new";
            newListItem.Top = 100;
            newListItem.Left = 100;
            newListItem.ValidForDisplayInGrid = ValidForDisplayGrid.MainGrid;
            newListItem.WindowType = ListUIActionWindowTypes.Popup;
            newListItem.WindowPosition = ListUIActionWindowPosition.Center;
            newListItem.WindowSize = ListUIActionWindowSize.ScreenPropotional;
            newListItem.DisplayValidRightValues = "ActionCreate<True>";
            newListItem.ActionAllowValidRightValues = "ActionCreate<True>";
            newListItem.ApprovalRequiredValidRightValues = "";
            newListItem.ApprovalRequiredWithPromptValidRightValues = "";
            newList.Operations.Actions.Add(newListItem);
            ListASPXUIAction editForm = new ListASPXUIAction();
            editForm.Id = "_sys_edit";
            editForm.Name = "Edit";
            editForm.WindowType = ListUIActionWindowTypes.Popup;
            editForm.ShowWhenSecurityAllowsRights = "";
            editForm.Type = ListActionType.MenuItem;
            editForm.ValidOnlyForItemTypes = "";
            editForm.ValidOnlyForItemStatus = "";
            editForm.IconPath = "designerView";
            editForm.PagePath = "DynamicListNewForm.aspx?mode=Edit";
            editForm.DisplayValidRightValues = "ActionEdit<true:,owneditemsonly:>";
            editForm.ActionAllowValidRightValues = "ActionEdit<true:,owneditemsonly:>";
            editForm.ApprovalRequiredValidRightValues = "";
            editForm.ApprovalRequiredWithPromptValidRightValues = "";
            newList.Operations.Actions.Add(editForm);
            ListASPXUIAction viewForm = new ListASPXUIAction();
            viewForm.Id = "_sys_view_item_details";
            viewForm.Name = "View Item Details";
            viewForm.WindowType = ListUIActionWindowTypes.Popup;
            viewForm.ShowWhenSecurityAllowsRights = "";
            viewForm.Type = ListActionType.MenuItem;
            viewForm.ValidOnlyForItemTypes = "";
            viewForm.ValidOnlyForItemStatus = "";
            viewForm.IconPath = "viewProperties";
            viewForm.PagePath = "DynamicListNewForm.aspx?mode=View";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
            viewForm.DisplayValidRightValues =
"ActionViewItemDetails<true:,owneditemsonly:>";
            viewForm.ActionAllowValidRightValues =
"ActionViewItemDetails<true:,owneditemsonly:>";
            viewForm.ApprovalRequiredValidRightValues = "";
            viewForm.ApprovalRequiredWithPromptValidRightValues = "";
            newList.Operations.Actions.Add(viewForm);
            ActionDelete deleteItem = new ActionDelete();
            deleteItem.Id = "_sys_delete";
            deleteItem.Name = "Delete";
            deleteItem.ActionCommand = "_sys_delete";
            deleteItem.ShowWhenSecurityAllowsRights = "";
            deleteItem.Type = ListActionType.MenuItem;
            deleteItem.ExecutionType = ActionExecutionType.SystemAction;
            deleteItem.ValidOnlyForItemTypes = "";
            deleteItem.ValidOnlyForItemStatus = "";
            deleteItem.IconPath = "Delete";
            deleteItem.DisplayValidRightValues =
"ActionDelete<allow::,approval::,approvalprompt::>";
            deleteItem.ActionAllowValidRightValues = "ActionDelete<allow::>";
            deleteItem.ApprovalRequiredValidRightValues = "ActionDelete<approval::>";
            deleteItem.ApprovalRequiredWithPromptValidRightValues =
"ActionDelete<approvalprompt::>";
            newList.Operations.Actions.Add(deleteItem);
            ActionNewCategory newCategory = new ActionNewCategory();
            newCategory.Id = "_sys_newfolder";
            newCategory.Name = "ecm_list_cm_newcategory";
            newCategory.ShowWhenSecurityAllowsRights = "";
            newCategory.ValidOnlyForItemStatus = "";
            newCategory.PagePath = "NewFormCategory.aspx";
            newCategory.Left = 100;
            newCategory.Top = 100;
            newCategory.Width = 1000;
            newCategory.ItemType = 0;
            newCategory.Type = ListActionType.Task;
            newCategory.Title = "New Category";
            newCategory.WindowType = ListUIActionWindowTypes.Popup;
            newCategory.WindowPosition = ListUIActionWindowPosition.Center;
            newCategory.WindowSize = ListUIActionWindowSize.ScreenPropotional;
            newCategory.DisplayValidRightValues = "ActionCreateFolder<true>";
            newCategory.ActionAllowValidRightValues = "ActionCreateFolder<true>";
            newCategory.ApprovalRequiredValidRightValues = "";
            newCategory.ApprovalRequiredWithPromptValidRightValues = "";
            newList.Operations.Actions.Add(newCategory);
            ListASPXUIAction manageitemsecurity = new ListASPXUIAction();
            manageitemsecurity.Id = "_sys_security_settings";
            manageitemsecurity.Name = "Security Settings";
            manageitemsecurity.ShowWhenSecurityAllowsRights = "";
            manageitemsecurity.ValidOnlyForItemStatus = "";
            manageitemsecurity.Type = ListActionType.MenuItem;
            manageitemsecurity.IconPath = "settingsMSG";
            manageitemsecurity.PagePath = "ListItemSettings.aspx";
            manageitemsecurity.IsModal = false;
            manageitemsecurity.ItemType = 1;
            manageitemsecurity.WindowPosition = ListUIActionWindowPosition.Center;
            manageitemsecurity.DisplayValidRightValues = "ManageSecurity<True>";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
            manageitemsecurity.ActionAllowValidRightValues = "ManageSecurity<True>";
            manageitemsecurity.ApprovalRequiredValidRightValues = "";
            manageitemsecurity.ApprovalRequiredWithPromptValidRightValues = "";
            newList.Operations.Actions.Add(manageitemsecurity);
            ListEventHandlerProvider eventProvider = new ListEventHandlerProvider();
            eventProvider.ClassName = " Skelta.Repository.List.ListEventHandler";
            eventProvider.AssemblyName = "Workflow.NET.NET2.dll";
            eventProvider.Name = "_sys_" + listName.ToLowerInvariant() + "_listhandler";
            newList.Operations.EventHandlers.Add(eventProvider);
        }
        private void InsertListNavigation(string applicationName, string loggedInUser,
string listName)
        {
            Guid DLNavgnDtlsId = Guid.Empty;
            ListTableCollection listCollection = new ListTableCollection(new
Skelta.Core.ApplicationObject(applicationName), "SKENavigationDetails", "Id");
            Skelta.Core.DS.SkeltaParameter paramTitle = new
Skelta.Core.DS.SkeltaParameter("@Title", "ec_menu_Lists");
            DataSet ds = listCollection.GetMatchingRecords(" And Title=@Title",
paramTitle);
            if (ds.Tables[0].Rows.Count > 0)
                DLNavgnDtlsId = new Guid(ds.Tables[0].Rows[0][0].ToString());
            else
                DLNavgnDtlsId =
Skelta.Forms.Web.Common.DynamicListCommonFunctions.InsertIntoListNavigation(applicationName
, loggedInUser, "ec_menu_Lists", "ec_menu_Lists", null, "icon-lists.gif");
            ds.Dispose();
            ListDefinition listdef = ListDefinition.GetList(new
Skelta.Core.ApplicationObject(applicationName), "NavigationDetails");
            Skelta.Repository.List.ListItem item = new
Skelta.Repository.List.ListItem(listdef);
            ((ListTextDataItem)item.ListForm.Records[0].FindControlByID("_sys_menu_disptext
")).Value = listName;//"ec_menu_" + _listName;
            ((ListTextDataItem)item.ListForm.Records[0].FindControlByID("_sys_menu_desc")).
Value = listName;//"ec_menu_" + _listName;
            ((ListTextDataItem)item.ListForm.Records[0].FindControlByID("_sys_menu_navto"))
.Value = "DynamicList.aspx?ListName=" + listName;
            item.LoggedInUserId = loggedInUser;
            item.ParentItemId = DLNavgnDtlsId;
            item.Save(ListItemVersionStatus.Published);
        }
    }
}
```

Note: Ensure to replace all <RepositoryName> instances with your repository name.

Update List Item by Validating Security

Check and update a list item using the list API.

The list API makes use of the following DLL references, namespaces, classes, and methods.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

DLL Reference	Namespace
Workflow.NET.NET2	• System • Skelta.Repository.List

Class
• Skelta.Repository.List.ListDefinition • Skelta.Repository.List.ListItem

Methods
• `ListItem.CanChangeStatus (ListItemVersionStatus)` This method checks whether you can change the status of a list item, and returns a Boolean value. • `ListItem.ChangeStatusAndUpdate(ListItemVersionStatus)` This method changes the status of a list item and updates the list item.

Scenario

For the **Employee Details** list mapped to an **Employee** table with the fields such as **Title**, **Name**, **Age**, **Salary**, **Department**, and **Manager**, update list item by running the following code.

```
using System;
using Skelta.Repository.List;
namespace ListObject
{
public class DynamicList
{
static void Main(string[] args)
{
DynamicList DL = new DynamicList();
DL.UpdateListItem("SkeltaRepo", "Employee", "admin");  }
private void UpdateListItem(string applicationname, string listname, string LoggedInUser)
{
// Creating ListDefinition Object.
ListDefinition userlist = new ListDefinition(new
Skelta.Core.ApplicationObject(applicationname), listname);
Guid listitemguid = new Guid("cead17b3-d91d-42d2-ac00-da0ccd1e3611");
// Creating ListItem Object.
ListItem listitem = new ListItem(userlist, listitemguid);
// Checking whether list item can be updated.
if (listitem.CanChangeStatus(ListItemVersionStatus.Published))
{
// Updating list item.
((ListTextDataItem)(listitem.ListForm.Records[0].FindControlByName("Name"))).Value = "ken
smith";
listitem.ChangeStatusAndUpdate(ListItemVersionStatus.Completed);
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Console.WriteLine("ListItem has been successfully Updated");
}
else
{
Console.WriteLine("Can not update ListItem");
}
}
}
}
}
```

Delete List Item by Validating Security

Check and delete a list item using the list API.

The list API makes use of the following DLL references, namespaces, classes, and methods.

DLL Reference	Namespace
Workflow.NET.NET2	• System • Skelta.Repository.List

Class
• Skelta.Repository.List.ListDefinition • Skelta.Repository.List.ListItem

Methods
• `ListItem.CanDelete()` This method checks whether you can delete an item, and returns a Boolean value. • `ListItem.Delete()` This method deletes a list item.

Scenario

For the **Employee Details** list mapped to an **Employee** table with the fields such as **Title**, **Name**, **Age**, **Salary**, **Department**, and **Manager**, delete list item by running the following code.

```
using System;
using Skelta.Repository.List;
namespace ListObject
{
public class DynamicList
{
static void Main(string[] args)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
DynamicList DL = new DynamicList();
DL.DeleteListItem("SkeltaRepo", "Employee", "admin");
}
 private void DeleteListItem(string applicationname, string listname, string LoggedInUser)
{
// Creating ListDefinition Object.
ListDefinition userlist = new ListDefinition(new
Skelta.Core.ApplicationObject(applicationname),
listname);
// Creating ListItem object.
ListItem listitem = new ListItem(userlist, new
Guid("cead17b3-d91d-42d2-ac00-da0ccd1e3611"));
// Checking whether list item can be deleted.
if (listitem.CanDelete())
{
// Deleting list item.
listitem.Delete();
Console.WriteLine("ListItem has been successfully deleted");
}
else
{
Console.WriteLine("Can not delete ListItem");
}
}
}
}
}
```

Get Values of Default Field of List

Retrieve values of the list items for the default field using the list API.

The list API makes use of the following DLL references, namespaces, classes, and methods.

DLL Reference	Namespace
Workflow.NET.NET2	• System • Skelta.Repository.List

Class
ListDataHandler

Methods
• `GetAllListItemsWithDefaultField()` This method retrieves values of all list items for the default field. • `GetTopLevelListItemsWithDefaultField()`

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Methods
This method retrieves values of the top level list items for the default field. • `GetChildListitemsWithDefaultFieldForValue()` This method retrieves values of the child level list items for the default field. • `GetChildListItemsWithDefaultFieldForId(id)` This method retrieves values of the child level list items for the default field for a specific ID. • `GetChildListitemsWithDefaultFieldForValue(name)` This method retrieves values of the child level list items for the default field for a specific name.

Scenario

Create a list called **Customers** with **CustomerId (Int Datatype)** configured as the **UniqueField** and **CustomerName** configured as the **DefaultField**. Retrieve the values of the new Properties **DefaultField** and **Unique Field** added to the ListDefintion by running the following code.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Skelta.Repository.List;
namespace ConsoleApplication1
{
class Program
{
static void Main(string[] args)
{
string repository = "March 2011";
string list = "Customers";
// Creating ListDataHandler object.
ListDataHandler ldh = new ListDataHandler(repository, list, false);
// Retrieving DefaultField.
string defaultfield = ldh.DefaultField;
// Retrieving UniqueField.
string uniquefield = ldh.UniqueField;
// Retrieving top level list items with DefaultField.
Dictionary<Guid, object> customeritems = new Dictionary<Guid, object>();
customeritems = ldh.GetTopLevelListItemsWithDefaultField();
// Looping through individual items to retrieve them.
foreach (KeyValuePair<Guid, object> customer in customeritems)
{
Guid Id = customer.Key;
string customername = customer.Value.ToString();
}
// Retrieving all list items with DefaultField.
Dictionary<Guid, object> customeritems1 = new Dictionary<Guid, object>();
customeritems1 = ldh.GetAllListItemsWithDefaultField();
// Looping through individual items to retrieve them.
foreach (KeyValuePair<Guid, object> customer in customeritems1)
{
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Guid Id = customer.Key;
string customername = customer.Value.ToString();
}
// Retrieving all child items (sub customers) of a specific item (customer) with name John.
object cusotmername = "John";
Dictionary<Guid, object> subcustomeritems = new Dictionary<Guid, object>();
subcustomeritems = ldh.GetChildListitemsWithDefaultFieldForValue(customername);
// Looping through individual items to retrieve them.
foreach (KeyValuePair<Guid, object> customer in subcustomeritems)
{
Guid Id = customer.Key;
string customername = customer.Value.ToString();
}
// Retrieving all child items (sub customers) of a specific item (customer) with Id field.
Guid itemid = new Guid("87B15910-54F8-4F78-9EA5-3E3FD3AD3286");
Dictionary<Guid, object> subcustomeritems1 = new Dictionary<Guid, object>();
subcustomeritems1 = ldh.GetChildListItemsWithDefaultFieldForId(itemid);
// Looping through individual items to retrieve them.
foreach (KeyValuePair<Guid, object> customer in subcustomeritems1)
{
Guid Id = customer.Key;
string customername = customer.Value.ToString();
}
// Retrieving the Item Id for the customer with the name Kevin.
object cname = "Kevin";
Guid KevinId = ldh.GetListItemIdForDefaultField(cname);
}
}
}
```

Save ListItem

Save a ListItem using list API.

The list API makes use of the following DLL references, namespaces, classes, and methods.

DLL Reference	Namespace
• Workflow.NET.NET2 • Skelta.Forms.Core	• System • Skelta.Repository.List

Class
• Skelta.Repository.List.ListDefinition • Skelta.Repository.List.ListItem

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Methods
• `ListItem.Save()` This method is used to save the ListItem.

Scenario 1

For the **Employee** list mapped to an **Emp** table with the fields such as **Title**, **Name**, **City**, **State**, **Pincode**, and **Age**, save the ListItem by running the following code.

```
using System;
using Workflow.NET;
using Workflow.NET.Engine;
using Skelta.Forms.Core.Controls;
using Skelta.Repository.List;
public class WorkflowScript0144412702584f2a8852a95a0a727737
{
public string Run(int ExecutionId, int ExecutionDetailsId, Workflow.NET.Engine.Context ctx,
Workflow.NET.Action action, string inlink)
{
string applicationname = ctx.ApplicationName;
string listname = "Employee";
ListDefinition lstdef = new ListDefinition(new
Skelta.Core.ApplicationObject(applicationname), listname);
Skelta.Repository.List.ListItem listItem = new Skelta.Repository.List.ListItem(lstdef);
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("Title"))).Value = "T3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("Name"))).Value = "A3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("City"))).Value = "C3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("State"))).Value = "S3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("PinCode"))).Value =
"P3";
((ListIntDataItem)(listItem.ListForm.Records[0].FindControlByName("Age"))).Value = "20";
listItem.Save();
return "saved";
}
}
```

Scenario 2

If we add a new column to the **Emp** Table say **Department**, then we need to call the `ClearTableNameKeyFromTableCollection` API before calling the `listitem.save()`.

```
Skelta.Core.DS.SingleTableObject.ClearTableNameKeyFromTableCollection(ctx.DataHandler.DataSourceTyp
applicationname, "EmpTable");
```

```
using System;
using Workflow.NET;
using Workflow.NET.Engine;
using Skelta.Forms.Core.Controls;
using Skelta.Repository.List;
public class WorkflowScript0144412702584f2a8852a95a0a727737
{
public string Run(int ExecutionId, int ExecutionDetailsId, Workflow.NET.Engine.Context ctx,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
Workflow.NET.Action action, string inlink)
{
string applicationname = ctx.ApplicationName;
string listname = "Employee";
Skelta.Core.DS.SingleTableObject.ClearTableNameKeyFromTableCollection(ctx.DataHandler.DataS
ourceType, applicationname, "Emp");
ListDefinition lstdef = new ListDefinition(new
Skelta.Core.ApplicationObject(applicationname), listname);
Skelta.Repository.List.ListItem listItem = new Skelta.Repository.List.ListItem(lstdef);
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("Title"))).Value = "T3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("Name"))).Value = "A3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("City"))).Value = "C3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("State"))).Value = "S3";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("PinCode"))).Value =
"P3";
((ListIntDataItem)(listItem.ListForm.Records[0].FindControlByName("Age"))).Value = "20";
((ListTextDataItem)(listItem.ListForm.Records[0].FindControlByName("Department "))).Value =
"R&D";
listItem.Save();
return "saved";
}
}
```

Advanced List Features

This section explains the advanced List features available in AVEVA Work Tasks.

List Control Properties

The Tool Box provides a set of controls to design a list. The List Designer provides an easy to use drag and drop interface for designing a list using the various controls. Controls can also be added by clicking the Add New icon on the List Menu.

The following is a reference to all the controls, including their attributes and script properties. The attributes of a list are arranged in two tabs, Basic and Advanced. The Scripts tab attributes are common for all the controls and is explained in "Viewing Scripts in Lists".

List(Parent)

There are two tabs for setting the properties:

Basic

Field	Description
Tag name	Enter a tag name that identifies the list main form name.
Description	Enter a description for the list main form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Mapped XML Node	This field displays the XML node mapped to the list main form. It allows mapping of a List to a Form and the Form to a Workflow. **Note:** Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists and Workflows.
DefaultSortOrder **	This property is used to display the list items in a specified order for the dynamic list. You can specify the comma separated column names for list sorting. Example: In a list, assume that there are columns like OrderNo, Name, Description, Quantity, Price and Total. To sort the list based on the OrderNo, Name and Quantity, specify the Default Sort Order property as OrderNo, Name, Quantity.
Allow Child items?	Select **Yes**, to allow child items, else select **No**.
Show All Standard Columns?	Select **Yes**, to show all standard columns, else select **No**.
Show Specific Standard Columns?	Type the Standard Columns which you want to display in the grid, separated by commas. To show the columns; Title, Description, Owner, Version, Item Status, Work in Progress, Workflow Status, and LastModified Date type **Title**, **Description**, **Owner**, **Version**, **Status**, **WIP**, **WFStatus,** and **LMD** respectively separated by commas.
Table name	This field displays the database table name to which the list main form is bound to. To edit the table, click the **Table** button and select the required table.

The column names specified in the **Default Sort Order** property and the actual columns names should be same. If the column name entered in the **Default Sort Order** property and the actual column name is different then that column will be ignored while sorting.

If a column specified in the **Default Sort Order** property is not available in the grid then the column will be ignored while sorting. For example, if **Default Sort Order** property is specified as **OrderNo**, **Name**, **Quantity** but if the **OrderNo** column is not available in the grid, then the list will be sorted based on the **Name** and **Quantity** columns.

The **Allow Sorted** property and the **Default Sort Order** properties are different. By default, the **Allow Sorted** property of list item is set as **False**. For example, if **DefaultSortOrder** property is specified as **OrderNo**, **Name**, **Quantity** and the **Allow Sorted** property of the **OrderNo** is set as **True** then list items will be displayed with default sort order specified in the **Default Sort Order** property when the dynamic list is loaded for the first time. If you sort by **OrderNo** column then list items are displayed based on the OrderNo column.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For faster sorting, index the columns specified in the default sort order property.

Advanced

Field	Description
Default Field	It is the default field used to hold the item details such as item id, version and item type. The field that is specified in the Default Field will serve the purpose of Title field. The options **Available in grid?** and **Available in lookup?**, should be set as **Yes** in the **List properties** dialog for the field that is set as **Default Field**. Example: If the field **Name** is specified as Default Field, then the control Name should be edited and the options **Available in grid?** and **Available in lookup?** should be set as **Yes**.
Id	The reference identification code in this field is generated automatically.
Unique Title?	Select **Yes** for the title to be unique, else select **No**.
Unique Field	This field works as a unique key to the List. The list item is identified based on the unique key. Every list can have a unique key defined, based on which the list item is identified. Given a scenario where the user creates an Order List with Order Type, Order Category, and Order ID fields. Since the Order ID is the uniquely identifiable key for this list, it becomes the unique key. In addition, the user can assign multiple fields to a unique key. For example, in an 'Employee List' which has First Name, Last Name, Address, Age etc., the user can assign the First Name and the Last Name as the Unique Key for this list item. Note that the user is not restricted to use only string values (first name and last name are a part of the type string). The user may use different combinations of Integer, String fields and date fields etc. For example, the user can use Age and Date of Birth or Address and Telephone number as a Unique Key. Unique Key can be defined to a maximum limit of five fields in a list. **Note:** Unique key can be defined only in a parent table.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Attachment

There are two tabs for setting the properties:

Basic

Field	Description
Tag name	Enter a tag name that identifies the list main form name.
Description	Enter a description for the list main form.
Mapped XML Node	This field displays the XML node mapped to the list main form. It allows mapping of a List to a Form and the Form to a Workflow. **Note:** Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists and Workflows.
Bound to DB Column	Enter the name of the database column to which it is bound.

Advanced

Field	Description
Grid display width	Enter the grid display width.
Available in grid?	Select **Yes**, if you want to make the content available in the grid, else select **No**. This field should be set as **Yes**, if the control is set as Default Field
Available in lookup?	Select **Yes**, if you want to make the control available in any lookup, else select **No**. This field should be set as **Yes**, if the control is set as Default Field
Grid caption	Enter a caption for the grid.
Grid Index	This field is used to display field names on the Lists page in a specific order. The field names will be displayed in an ascending order. By default, the value for **Grid Index** field is 100. You can enter only numerical value for the Grid Index

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	field. Example: Assume that, 'Employee Details' list has the 'Employee Name', 'Employee Id', 'Experience', 'Skills' and 'Salary' fields. If you specify the Grid Index value as follows: Employee Name - 2, Employee Id - 1, Experience - 40 ('Experience' field is '40' in case later you want to add fields before 'Experience' field and after 'Employee Name'), Skills - 3, Salary - 5, Then it will display the fields in the specified order on the Lists page. i.e. Employee Id, Employee Name, Skills, Salary and Experience. **Note:** Make sure that the Grid Index that you have specified is unique for a field.
Id	The reference identification code in this field is generated automatically.
Lookup display index	Enter a numerical value that identifies the Lookup display Index.
Persist file name as Header	It persists the filename uploaded by user as the header information along with the file.
Id of the control which persists filename	If the Persist file name as Header property is set as No, the filename of the uploaded file can also be persisted in another control present at the same hierarchy.
Is a lookup title?	Select **Yes**, if you want to make the control act as a lookup title, else select **No**.
Image Path (Relative URL)	Enter the image relative url, where the image is located.

BooleanInput

The **BooleanInput** control is used to define a Boolean value (Yes or No).

There are two tabs for setting the properties:

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Tag name	Enter a tag name that identifies the control name.
Description	Enter a description for the control.
Mapped XML Node	This field displays the XML node mapped to the control. It allows mapping of a List to a Form and the Form to a Workflow. When you add a control for the first time, the XML Node is dynamically altered to reflect the Tag Name. **Note:** Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists and Workflows.
Bound to DB column	Enter the name of the database column to which it is bound.
Mandatory?	Select **Yes**, if the control has to be a mandatory field, else select **No**.
Allow Sorting	Select **Yes**, if you want to allow the sorting of the list column, else select **No**.

Advanced

Field	Description
Available in grid?	Select **Yes**, if you want to make the content available in the grid, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field.**
Available in lookup?	Select **Yes**, if you want to make the control available in any lookup, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**.
Grid caption	Enter a caption for the grid.
Grid Display Type	Select the appropriate grid display type from the drop-down list. The available types are: • Only Text • Only Image • ImageAndText

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Grid Index	This field is used to display field names on the Lists page in a specific order. The field names will be displayed in an ascending order. By default, the value for Grid Index field is 100. You can enter only numerical value for the Grid Index field. Example: Assume that, 'Employee Details' list has the 'Employee Name', 'Employee Id', 'Experience', 'Skills' and 'Salary' fields. If you specify the Grid Index value as follows: Employee Name - 2, Employee Id - 1, Experience - 40 ('Experience' field is '40' in case later you want to add fields before 'Experience' field and after 'Employee Name'), Skills - 3, Salary - 5, Then it will display the fields in the specified order on the Lists page. i.e. Employee Id, Employee Name, Skills, Salary and Experience. **Note:** Make sure that the Grid Index that you have specified is unique for a field.
Grid display width	Enter the grid display width.
Id	The reference identification code in this field is generated automatically.
Image Path (Relative URL)	Enter the image relative url, where the image is located.
Is a lookup title?	Select **Yes**, if you want to make the control act as a lookup title, else select **No**.
Lookup display index	Enter a numerical value that identifies the Lookup display Index.

DateInput

The **DateInput** control is used to define the date, along with the Calendar pick list.

There are two tabs for setting the properties:

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Tag name	Enter a tag name that identifies the control name.
Description	Enter a description for the control.
Mapped XML Node	This field displays the XML node mapped to the control. It allows mapping of a List to a Form and the Form to a Workflow. When you add a control for the first time, the XML Node is dynamically altered to reflect the Tag Name. **Note:** Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists and Workflows.
Bound to DB column	Enter the name of the database column to which it is bound.
Mandatory?	Select **Yes**, if the control has to be a mandatory field, else select **No**.
Allow Sorting	Select **Yes**, if you want to allow the sorting of the list column, else select **No**.

Advanced

Field	Description
Available in grid?	Select **Yes**, if you want the content to be available in the grid, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**
Available in lookup?	Select **Yes**, if you want the control to be available in any lookup, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**
Grid caption	Enter a caption for the grid.
Grid Display Type	Select the appropriate grid display type from the drop-down list. The available types are: • Only Text • Only Image • ImageAndText

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Grid Index	This field is used to display field names on the Lists page in a specific order. The field names will be displayed in an ascending order. By default, the value for Grid Index field is 100. You can enter only numerical value for the Grid Index field. Example: Assume that, 'Employee Details' list has the 'Employee Name', 'Employee Id', 'Experience', 'Skills' and 'Salary' fields. If you specify the Grid Index value as follows: Employee Name - 2, Employee Id - 1, Experience - 40 ('Experience' field is '40' in case later you want to add fields before 'Experience' field and after 'Employee Name'), Skills - 3, Salary - 5, Then it will display the fields in the specified order on the Lists page. i.e. Employee Id, Employee Name, Skills, Salary and Experience. Make sure that the Grid Index that you have specified is unique for a field.
Grid display width	Enter the grid display width.
Id	The reference identification code in this field is generated automatically.
Image Path (Relative URL)	Enter the image relative url, where the image is located.
Is a lookup title?	Select **Yes**, if you want the control to act as a lookup title, else select **No**.
Lookup display index	Enter a numerical value that identifies the Lookup display Index.

DecimalInput

The **DecimalInput** control is used to define a decimal value.

There are two tabs for setting the properties:

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Tag name	Enter a tag name that identifies the control name.
Description	Enter a description for the control.
Mapped XML Node	This field displays the XML node mapped to the control. It allows mapping of a List to a Form and the Form to a Workflow. When you add a control for the first time, the XML Node is dynamically altered to reflect the Tag Name. **Note:** Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists and Workflows.
Bound to DB column	Enter the name of the database column to which it is bound.
Mandatory?	Select **Yes**, if the control has to be a mandatory field, else select **No**.
Allow Sorting	Select **Yes**, if you want to allow the sorting of the list column, else select **No**.

Advanced

Field	Description
Available in grid?	Select **Yes**, if you want the content to be available in the grid, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**
Available in lookup?	Select **Yes**, if you want the control to be available in any lookup, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**
Grid caption	Enter a caption for the grid.
Grid Display Type	Select the appropriate grid display type from the drop-down list. The available types are: Only Text Only Image ImageAndText
Grid Index	This field is used to display field names on the Lists

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	page in a specific order. The field names will be displayed in an ascending order. By default, the value for Grid Index field is 100. You can enter only numerical value for the Grid Index field. Example: Assume that, 'Employee Details' list has the 'Employee Name', 'Employee Id', 'Experience', 'Skills' and 'Salary' fields. If you specify the Grid Index value as follows: Employee Name - 2, Employee Id - 1, Experience - 40 ('Experience' field is '40' in case later you want to add fields before 'Experience' field and after 'Employee Name'), Skills - 3, Salary - 5, Then it will display the fields in the specified order on the Lists page. i.e. Employee Id, Employee Name, Skills, Salary and Experience. Make sure that the Grid Index that you have specified is unique for a field.
Grid display width	Enter the grid display width.
Id	The reference identification code in this field is generated automatically.
Image Path (Relative URL)	Enter the image relative url, where the image is located.
Is a lookup title?	Select **Yes**, if you want the control to act as a lookup title, else select **No**.
Lookup display index	Enter a numerical value that identifies the Lookup display Index.

GUIDInput

The **GUIDInput** control is used to store the GUID (Globally Unique Identifiers).

There are three tabs for setting the properties:

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Tag name	Enter a tag name that identifies the control name.
Description	Enter a description for the control.
Mapped XML Node	This field displays the XML node mapped to the control. It allows mapping of a List to a Form and the Form to a Workflow. When you add a control for the first time, the XML Node is dynamically altered to reflect the Tag Name. **Note:** Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists and Workflows.
Bound to DB column	Enter the name of the database column to which it is bound.
Mandatory?	Select **Yes**, if the control has to be a mandatory field, else select **No**.
Allow Sorting	Select Yes, if you want to allow the sorting of the list column, else select **No**.

Advanced

Field	Description
Available in grid?	Select **Yes**, if you want the content to be available in the grid, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**
Available in lookup?	Select **Yes**, if you want the control to be available in any lookup, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**
Grid caption	Enter a caption for the grid.
Grid Display Type	Select a grid display type from the drop-down list. The available types are: • Only Text • Only Image • ImageAndText

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Grid Index	This field is used to display field names on the Lists page in a specific order. The field names will be displayed in an ascending order. By default, the value for Grid Index field is 100. You can enter only numerical value for the Grid Index field. Example: Assume that, 'Employee Details' list has the 'Employee Name', 'Employee Id', 'Experience', 'Skills' and 'Salary' fields. If you specify the Grid Index value as follows: Employee Name - 2, Employee Id - 1, Experience - 40 ('Experience' field is '40' in case later you want to add fields before 'Experience' field and after 'Employee Name'), Skills - 3, Salary - 5, Then it will display the fields in the specified order on the Lists page. i.e. Employee Id, Employee Name, Skills, Salary and Experience. Make sure that the Grid Index that you have specified is unique for a field.
Grid display width	Enter the grid display width.
Id	Displays the identification code for the reference.
Image Path (Relative URL)	Enter the image relative url, where the image is located.
Is a lookup title?	Select **Yes**, if you want to make the control act as a lookup title, else select **No**.
Lookup display index	Enter a numerical value that identifies the Lookup display Index.

IntegerInput

The **IntegerInput** control is used to define an integer value.

There are two tabs for setting the properties:

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Tag name	Enter a tag name that identifies the control name.
Description	Enter a description for the control.
Mapped XML Node	This field displays the XML node mapped to the control. It allows mapping of a List to a Form and the Form to a Workflow. When you add a control for the first time, the XML Node is dynamically altered to reflect the Tag Name. **Note:** Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists and Workflows.
Bound to DB column	Enter the name of the database column to which it is bound.
Mandatory?	Select **Yes**, if the control has to be a mandatory field, else select **No**.
Allow Sorting	Select **Yes**, if you want to allow the sorting of the list column, else select **No**.

Advanced

Field	Description
Available in grid?	Select **Yes**, if you want the content to be available in the grid, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**
Available in lookup?	Select **Yes**, if you want the control to be available in any lookup, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**
Grid caption	Enter a caption for the grid.
Grid Display Type	Select a grid display type from the drop-down list. The available types are: • Only Text • Only Image • ImageAndText

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Grid Index	This field is used to display field names on the Lists page in a specific order. The field names will be displayed in an ascending order. By default, the value for Grid Index field is 100. You can enter only numerical value for the Grid Index field. Example: Assume that, 'Employee Details' list has the 'Employee Name', 'Employee Id', 'Experience', 'Skills' and 'Salary' fields. If you specify the Grid Index value as follows: Employee Name - 2, Employee Id - 1, Experience - 40 ('Experience' field is '40' in case later you want to add fields before 'Experience' field and after 'Employee Name'), Skills - 3, Salary - 5, Then it will display the fields in the specified order on the Lists page. i.e. Employee Id, Employee Name, Skills, Salary and Experience. Make sure that the Grid Index that you have specified is unique for a field.
Grid display width	Enter the grid display width.
Id	Displays the identification code for the reference.
Image Path (Relative URL)	Enter the image relative url, where the image is located.
Is a lookup title?	Select **Yes**, if you want to make the control act as a lookup title, else select **No**.
Lookup display index	Enter a numerical value that identifies the Lookup display Index.

UserLookup

The **UserLookup** control is used to select the users for the role.

There are three tabs for setting the properties:

Basic

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Tag name	Enter a tag name that identifies the control name.
Description	Enter a description for the control.
Mapped XML Node	This field displays the XML node mapped to the control. It allows mapping of a List to a Form and the Form to a Workflow. When you add a control for the first time, the XML Node is dynamically altered to reflect the Tag Name. **Note:** Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists and Workflows.
Bound to DB column	Enter the name of the database column to which it is bound.
Mandatory?	Select **Yes**, if the control has to be a mandatory field, else select **No**.
Allow Sorting	Select **Yes**, if you want to allow the sorting of the list column, else select **No**.
Show Default User	Select **Yes** to display the default user to user control and select **No** if you do not want to display the default user.
Mode of text	Select **SingleLine**, to make the user enter single line of text, select **MultLine** to allow the user to enter multiple lines of text.

Appearance

Field	Description
Number of rows	Enter the number of rows in which the control is to be displayed.

Advanced

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Available in grid?	Select **Yes**, if you want the content to be available in the grid, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**
Available in lookup?	Select **Yes**, if you want the control to be available in any lookup, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**
Grid caption	Enter a caption for the grid.
Grid Display Type	Select a grid display type from the drop-down list. The available types are: • Only Text • Only Image • ImageAndText
Grid Index	This field is used to display field names on the List page in a specific order. The field names will be displayed in an ascending order. By default, the value for Grid Index field is 100. You can enter only numerical value for the Grid Index field. Example: Assume that, 'Employee Details' list has the 'Employee Name', 'Employee Id', 'Experience', 'Skills' and 'Salary' fields. If you specify the Grid Index value as follows: Employee Name - 2, Employee Id - 1, Experience - 40 ('Experience' field is '40' in case later you want to add fields before 'Experience' field and after 'Employee Name'), Skills - 3, Salary - 5, Then it will display the fields in the specified order on the List page. i.e. Employee Id, Employee Name, Skills, Salary and Experience. Make sure that, the Grid Index that you have specified is unique for a field.
Grid display width	Enter the grid display width.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Id	Displays the identification code for the reference.
Image Path (Relative URL)	Enter the image relative url, where the image is located.
Is a lookup title?	Select **Yes**, if you want the control to act as a lookup title, else select **No**.
Lookup display index	Enter a numerical value that identifies the Lookup display index.

LookupInput

The **LookupInput** control is used to select the look up input from the drop-down list.

There are two tabs for setting the properties:

Basic

Field	Description
Tag name	Enter a tag name that identifies the control name.
Description	Enter a description for the control.
Mapped XML Node	This field displays the XML node mapped to the control. It allows mapping of a List to a Form and the Form to a Workflow. When you add a control for the first time, the XML Node is dynamically altered to reflect the Tag Name. **Note:** Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists and Workflows.
Bound to DB column	Enter the name of the database column to which it is bound.
Mandatory?	Select **Yes**, if the control has to be a mandatory field, else select **No**.
Lookup Window Title	Enter a title, which is to be displayed above the lookup grid.
Lookup Window Description	Enter a description, which is to be displayed above the lookup grid.
Lookup Window Page Height	Enter an integer value in this field, to set the height of

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	the Lookup Window Page in pixels. A value '0' will set the height of the field in default percentage value.
Lookup Window Page Width	Enter an integer value in this field, to set the width of the Lookup Window Page Width in pixels. A value '0' will set the width of the field in default percentage value.
Show as DropDown?	Select Yes, if you want to show List Lookup control as a drop-down. The **Lookup Window Page Height** and **Lookup Window Page Width** field can be used to change height and width of the **Lookup** control. Select **No**, if you want to show Lookup control.
Allow Sorting	Select **Yes**, if you want to allow the sorting of the list column, else select **No**.
Data Source	Click **Data Source** button to select the Lookup Data Source. The Data source for a lookup can be a Database or a List. Database is used to populate lookup data from a database and List is used to populate data from an existing List. For more details on how to create a new lookup, refer Lookup Settings.

Advanced

Field	Description
Available in grid?	Select **Yes**, if you want to make the content available in the grid, else select **No**. This field should be set as Yes, if the control is set as Default Field
Available in lookup?	Select **Yes**, if you want to make the control available in any lookup, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**.
Grid caption	Enter a caption for the grid.
Grid Display Type	Select a grid display type from the drop-down list. The available types are:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	• Only Text • Only Image • ImageAndText
Grid Index	This field is used to display field names on the Lists page in a specific order. The field names will be displayed in an ascending order. By default, the value for Grid Index field is 100. You can enter only numerical value for the Grid Index field. Example: Assume that, 'Employee Details' list has the 'Employee Name', 'Employee Id', 'Experience', 'Skills' and 'Salary' fields. If you specify the Grid Index value as follows: Employee Name - 2, Employee Id - 1, Experience - 40 ('Experience' field is '40' in case later you want to add fields before 'Experience' field and after 'Employee Name'), Skills - 3, Salary - 5, Then it will display the fields in the specified order on the Lists page. i.e. Employee Id, Employee Name, Skills, Salary and Experience. Make sure that the Grid Index that you have specified is unique for a field.
Grid display width	Enter the grid display width.
Id	Displays the identification code for the reference.
Image Path (Relative URL)	Enter the image relative url, where the image is located.
Is a lookup title?	Select **Yes**, if you want to make the control act as a lookup title, else select No.
Lookup display index	Enter a numerical value that identifies the Lookup display Index.
FormProviderNameProperty	Use of this property is not recommended because it is being deprecated in future releases.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

TextInput

The **TextInput** control is used to define the field required for entering textual content.

There are two tabs for setting the properties:

Basic

Field	Description
Tag name	Enter a tag name that identifies the control name.
Description	Enter a description for the control.
Mapped XML Node	This field displays the XML node mapped to the control. It allows mapping of a List to a Form and the Form to a Workflow. When you add a control for the first time, the XML Node is dynamically altered to reflect the Tag Name. **Note:** Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists and Workflows.
Bound to DB column	Enter the name of the database column to which it is bound.
Mandatory?	Select **Yes**, if the control has to be a mandatory field, else select **No**.
Allow Sorting	Select **Yes**, if you want to allow the sorting of the list column, else select **No**.

Advanced

Field	Description
Available in grid?	Select **Yes**, if you want the content to be available in the grid, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**.
Available in lookup?	Select **Yes**, if you want the control to be available in any lookup, else select **No**. This field should be set as **Yes**, if the control is set as **Default Field**.
Grid caption	Enter a caption for the grid.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Grid Display Type	Select a grid display type from the drop-down list. The available types are: • Only Text • Only Image • ImageAndText
Grid Index	This field is used to display field names on the Lists page in a specific order. The field names will be displayed in an ascending order. By default, the value for Grid Index field is 100. You can enter only numerical value for the Grid Index field. Example: Assume that, 'Employee Details' list has the 'Employee Name', 'Employee Id', 'Experience', 'Skills' and 'Salary' fields. If you specify the Grid Index value as follows: Employee Name - 2, Employee Id - 1, Experience - 40 ('Experience' field is '40' in case later you want to add fields before 'Experience' field and after 'Employee Name'), Skills - 3, Salary - 5, Then it will display the fields in the specified order on the Lists page. i.e. Employee Id, Employee Name, Skills, Salary and Experience. Make sure that the Grid Index that you have specified is unique for a field.
Grid display width	Enter the grid display width.
Id	Displays the identification code for the reference.
Image Path (Relative URL)	Enter the image relative url, where the image is located.
Is a lookup title?	Select **Yes**, if you want the control to act as a lookup title, else select **No**.
Lookup display index	Enter a numerical value that identifies the Lookup display Index.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

List Table

The **List Table** control is used to select a table for storing list data. To use the existing table, you need to create a table before navigating to the List Designer. For more information on creating a table, see Create List.

The available tables are:

- Generic Table
- Existing Table

In the **Select Table** screen, select an option and click **Select**.

If you select the **Generic Table** option, the corresponding **Properties** dialog box opens, which allows you to set its properties.

There are two tabs for setting the properties:

Basic

Field	Description
Tag name	Enter a tag name that identifies the control name.
Description	Enter a description for the control.
Mapped XML Node	This field displays the XML node mapped to the control. It allows mapping of a List to a Form and the Form to a Workflow. When you add a control for the first time, the XML Node is dynamically altered to reflect the Tag Name. **Note:** Editing the Mapped XML Node at a later instance will affect the mapping between Forms, Lists and Workflows.
Table Name	This field displays the database table name to which the list main form is bound to. To edit the table, click the **Table** button and select the required table.

Advanced

Field	Description
Id	The reference identification code in this field is generated automatically.

If you select the **Existing Table** option, then select a table from the available list and click **Select**. The corresponding **Properties** dialog box opens, which allows you to set its properties. You can set the properties as explained above.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: For generic List, ensure that the Bound to DB column Name is not the same for both Parent and Child control.

In addition, if 2 rows have the same data in all the columns displayed in the grid, then the row will not be displayed twice, since we use Distinct queries.

So ensure that at least one column (Display column) is added to distinguish the 2 data rows containing similar data.

XML Variables for Lists

Introduction to XML List Variables

AVEVA Work Tasks introduces XML Variables as a way to provide data inputs to workflows. The various types of XML Variables provided can be configured in the Start activity through the XML Variable property. An XML List type XML Variable can be mapped to any list schema i.e. either to lists provided out of the box or to lists created dynamically using create list option, to perform retrieval of data of a list item.

Using XML list XmlVariable, you can fetch the data from any list. XML Variable for XML list can be mapped to any XML List schema for inserting, updating and retrieving list items. Also you can attach the list to a workflow very easily in a few steps.

Scope of List XML Variable

List XML variable can be set to **Current**, **New** and **List Query** scopes. In that, you can set Current and New scopes at the time of creating XML List type XML variable and all the three scopes including List Query can set up in the List Navigator activity properties.

- **Current** : The XML variable will be pointed to the list item which initiated the workflow.
- **New** : The XML variable will be pointed to add a new list item to the list. Accessing this XML variable before adding an item will not give any details. After adding a list item, this XML variable will have the details of the added item. To add another item, use a list navigator action to re-initialize the scope as New.
- **List Query**: This property allows you to build a query for the list and fetch the items based on the query.

Create a Workflow for a List in a Single Step

The List workflow option in the Workflow Ribbon bar helps to create and associate a workflow to a list in a single step.

To create a workflow for a list

1. Login to the **Enterprise Console** and create a list.
2. Click the **Enterprise Console** menu on the top and select Workflows from the drop-down list. The **Workflows** page appears.
3. Click **List** on the **Ribbon Bar**. The **New List Workflow** dialog box appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
New List Workflow	
Title	Enter a title of workflow for the List in this field.
Description	Enter a description for the workflow in this field.
Associate to List Event - This can also be done using the 'Associate' icon in the Workflow List Menu.	
List	Select a list id by clicking the **List** button. The List Explorer screen opens. Select the list name from the drop-down list and click **Submit**.
Event	Select an event from this drop-down list. The available options are **OnCreate**, **OnDelete**, **OnPublish**, **OnUnPublish**, and **OnVersionChange**.

4. Enter the appropriate details.
5. Click **Save and Continue**. The Workflow Options dialog box appears.

   Note: You can perform four tasks – provision the workflow, start the Workflow Wizard, design the Workflow, or just perform no task at all.
6. Select the required task by clicking the corresponding radio button and click Finish.
7. Using List workflow there are three incidents taken place.

Workflow Creation

When you click **Finish**, a workflow is created for the selected list. You can open this workflow by either by selecting Design the Workflow now from tasks or using Design Workflow in context menu of the workflow item.

List XML Variable

An XML variable of XML List type is also created by default and the list name is used as the variable name. To view the XML Variable, go to the workflow designer screen. Right click the **Start activity**, select **Activity Properties** from the options and then select the **Xml Variables** icon from the **Activity Properties**.

You can notice that, the variable is created with scope as **Current** , which means, this variable is responsible to give the information of the list item which initiates the workflow. The XML variables displayed in tree-view are the XML nodes that are mapped to the controls while creating the list.

System nodes are also created automatically for internal use. This helps you to save the list item in to another version as published or draft and also helps to save the item as folder for the list item.

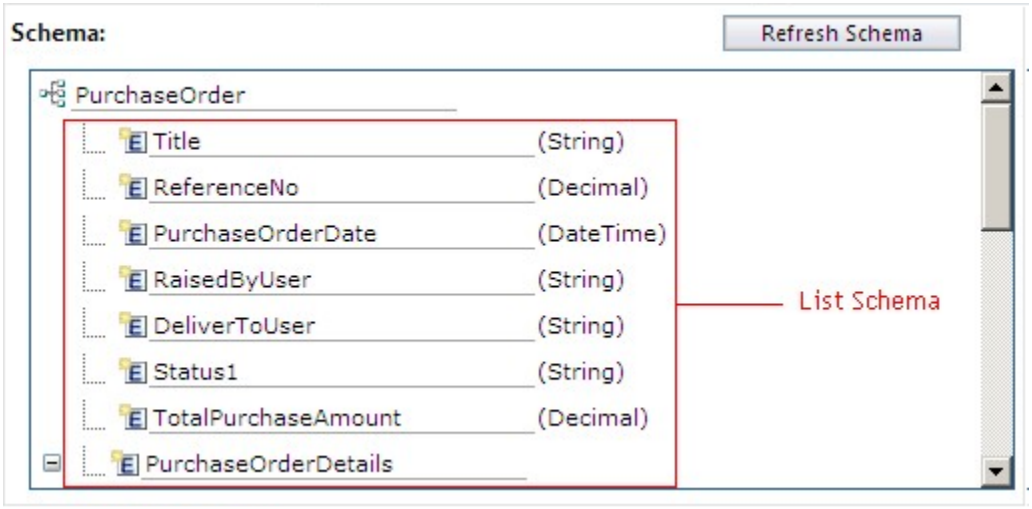
List Event Association

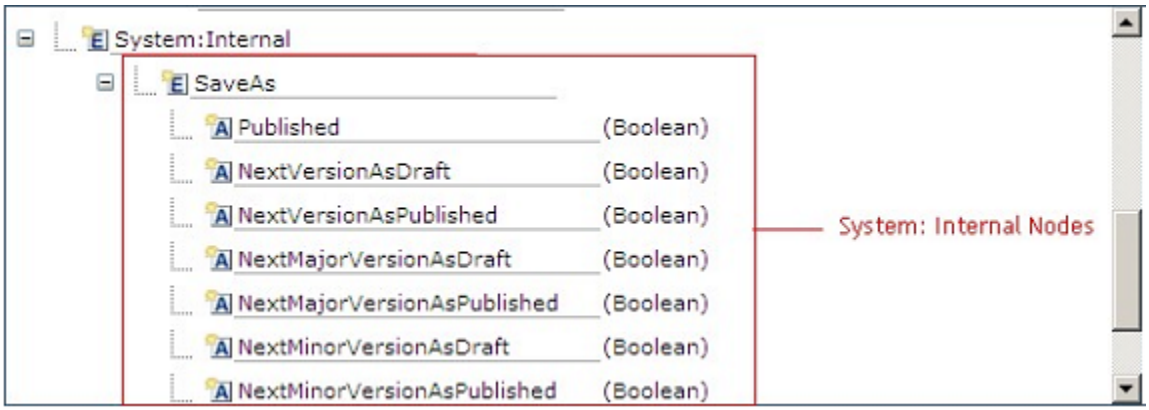
This workflow is associated to the list event that you have selected at the time of execution. To view the associations, select the workflow, right click and select View Associations from the options.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Schema of XML List XML Variable

The schema of XML list includes schema of the list and system internal nodes.





System Internal Nodes

When you create a list XML variable for the list, system internal nodes are also created automatically for internal use. This will help you to save your list item as a different version as **Published** or **Draft** when you update an item. It will also help you to save the item **As Folder** and **Published**.

If you want to add a new list item with **Published** status, select **Published** from the **SaveAs** option under **System:Internal** and then set value as **True**.

Similarly, you can select an appropriate operation from the **Operations** node to **Delete**, **Enable**, or **Disable** folders. Just as saving the folder, you can select these operations and then set the value as **True**.

Create an XML List Type XML Variable

XML Variable of List type can be configured in the Start activity through the XML Variable property. XML Variable of List type can be mapped to any list schema i.e. either to lists provided out of the box or to lists created dynamically using create list option, to perform retrieval of data of a list item.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To create an XML List type XML Variable

1. If you select **Design the Workflow now** option from the Workflow option screen while creating a workflow for the list.
2. Right click the **Start** and select **Activity Properties** from the options.
3. Click the **More** button in the **XMLVariables** field.
4. Click **New Variable** to add a new variable.
5. Type a variable name in the **Variable Name** field.
6. Select type as **XML List** from the **Type** drop-down list.
7. Type the List name in the **Select List** field or click the **List** button to select the list.
8. The XML schema displayed in tree-view are the XML nodes that are mapped to the controls while creating the list. Also you can see the System Internal nodes that are created automatically for internal use. This will help you to save the list item in another version as draft or published and help to create a folder.
9. Select the scope from the **Choose the ListItem to work on** section. The available options are:
   - **Current**: This will help to work with the list item that initiated the workflow.
   - **New**: The XML variable will be pointed to add a new list item to the list.
10. After entering all the required data, click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window. Now you have created an XML List type XML Variable.

Note:
- Later you can modify the variable. Click the **Refresh Schema** to reload the changes made to the list.
- After refreshing the schema, click **Save Variable** to save the changes.

Change the Scope of the XML List XML Variable

The scope (Choose the list item to work on options) of the XML list XML variable can be changed during the runtime. After the scope is changed, the XML list XML variable will be using the changed scope. You can change the scope of the XML list XML Variable using List Navigator activity during the workflow execution. Using this action, you can also navigate the XML list XML variable to another list having the similar schema.

To change the scope of the XML List XML Variable

1. Click **Activities** and select **List Activities** from the options.
2. Drag and drop **List Navigator** activity.
3. Select the **XML variable**, for which the scope needs to be changed, from the **XML Variable** of **List Type** drop down list.
4. Type/select the list name in the **List Name** or **ID** field. This property helps to change the list that needs to be worked on. The list entered here should have the same schema of the list from which the XML variable is created. If you don't want to change the working on list, then leave this field blank. After you change the list, the XML variable will be using the changed one.

   As an example, you have a list called Purchase Orders and another list called Approved Purchase Orders which has the same schema of the Purchase Orders list. If you want to add/update some item in the Approved Purchase Orders, you don't have to create separate XML variables for both the lists. You can use the same XML variables since both lists have the same schema. In this case, create XML variable for Purchase Orders. Enter the List Name as Approved Purchase Orders. And then set the scope as New. This

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

will point to the Approved Purchase Orders list and add/update the item in that list.

5. Click the drop-down and select the scope from the options. The available options are **Current**, **New** and **List Query**. Here you can change the scope.

Note:
- You can also use the list id. Id should be inside curly braces.
- Format to use {ListId}.Tell that this is optional and the list id need not have to be given when the list name is provided and vice versa.
- Also they can provide both as {ListId}ListName format.

Set up the List Query Scope

This property allows you to build the query for the list and retrieve the data based on the query. You need to point to any particular item to perform any task based on that item. This can be done by building the query using **List Query**.

To set up the list query scope

1. Select **List Query** as the scope and click **Save**. Then click the **Query** button of **List Query** property.
2. Click the **Query** button. The **List Query** dialog box appears.
3. The following are the field values for List Query window:

Field	Description
ListName	Select the list from List Name drop-down list.
Table	By clicking on this textbox, a popup allows you to select the corresponding table for query. The respective table name will display in the popup window based on the list selected by the user.
Column	You can select this drop-down list to select the column or field of the corresponding list.
Operator	You can select this drop-down list to select an operator (either one of =,>, <,>=, <=, <>, IN, Not IN, Like) to build the query.
Type	You can select this drop-down list to select a type. The value is automatically displayed in the textbox.
Condition	You can select this drop-down list to select the condition [either of End, And, Or, '),' '('.
Add	You can click this button to add a query in the field.
Clear	You can click this button to clear the query contents.
OK	You can click this button to save this query.
Close	You can click this button to close this window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Enter the details and click **Add** to add the List Query.
5. Click **OK** and then click **Close**.
6. As an example, you have a list named 'Purchase Orders' and you want to update the records that have the **Reference No** value '456789' and '123456' . This query can be built using list query.
7. Click the **Query** button to build the query for the list 'Purchase Orders'.
8. The list name **Purchase Orders** is selected automatically. If the list is not selected, select your list from the **List Name** drop-down list. Click the **Table** field. The pop up window allows you to select the table of the 'Purchase Orders'.
9. Click on **SKEListItemDataNV**. This will display automatically in the text box. Here the table names displayed in the tree view are the parent table (one on the top) and the child table (below one).
10. Click the **Column** drop-down list.
11. Select **int1** from the options since the field **Reference No** is mapped to this column of the database.
12. Select '=' operator from the **Operator** drop-down list.
13. Select **Value** from the **Type** drop-down list.
14. Type '**456789**' in the textbox.
15. Select **OR** from the **Condition** drop-down list to add another query.
16. Click **Add** to add the condition in the textfield.
17. Here, **SKEListItemDataNV** is the table name where the list has been stored and **t1** is the internal name that has been used for the main table. If there is any child table also, it will display as **t2. int1** in the field name, to that you need to point your XML variable and **456789** is the value for that field.
18. Repeat the same steps for the list item having '**123456**' as **Reference No**.
19. Enter the value as '**123456**'.
20. Select **End** from the **Condition** drop-down list.
21. Click **Add**. The build query will be displayed in the text field.
22. Click **OK** and then click **Close**.
23. Click **Update**.
24. You can replace the value for the query, with the variables. To replace, select the text/value to be replaced, and then select the respective field from the drop-down list and click **Append**.
25. A list query that uses a variable will look like this : **(SKEListItemDataNV(t1).int1 = '<%#XmlVariables.purchaseorders.PurchaseOrders.ReferenceNo%>'**.
26. Click **Update** and close the window.

Create an XML List Iterator Type XML Variable

XML List Iterator type XML variable is used to iterate through the list items (parent items) of a list and helps to retrieve/update data from a workflow.

To create an XML List Iterator type XML Variable

1. Right-click the **Start** Activity and select **Activity Properties** from the options. The **Process Designer** page appears.
2. Click the **More** button in the **XMLVariables** field. The existing list of XML variable is displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. You have to create an XML List type XML variable before creating XML List Iterator type XML variable. If you have already created XML List type XML variable, you can proceed to the following step. See Creating an Xml List type Xml variable to create XML List XML variable.
4. Click **New Variable** to add a new variable.
5. Enter a variable name in the **Variable Name** field.
6. Select **Xml List Iterator** from the **Type** drop-down list.
7. Select an XML Variable from the **Xml Variable** drop-down list.

   Note: If you need to loop through the grid items of an item while iterating through the list items, you need to select the respective **Xml List Iterator** type XML variable.
8. Select a repeating node from the **Node Select** drop-down list. Select the XML node which is mapped to the grid in the list.
9. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window.

Create an XML Iterator Type XML Variable for Grid Items

XML Iterator type XML variable is used to iterate though the child items (grid items) of a list item in a workflow.

To create an Xml Iterator type Xml Variable for grid items

1. Right-click the **Start** Activity and select **Activity Properties** from the options.
2. Click the **More** button in the **XMLVariables** field. The existing list of XML variables is displayed.
3. You have to create an XML List Iterator type XML variable before creating XML Iterator type XML variable. If you have already created XML List Iterator type XML variable, you can proceed to the following step. See Create an XML List Type XML Variable to create XML List XML variable.
4. Click **New Variable** to add a new variable.
5. Type a variable name in the **Variable Name** field.
6. Select XML Iterator from the **Type** drop-down list.
7. Select an XML Variable from the **Xml Variable** drop-down list.

   Note: If you need to loop through the grid items of an item while iterating through the list items, you need to select the respective XML List Iterator type XML variable.
8. Select a repeating node from the **Node Select** drop-down list. Select the XML node which is mapped for the grid in the list.
9. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window.

   Note: The user is unable to get the immediate values or attributes for the root node, if it was repeating.

Example

Consider an XML Schema

```
<EmployeeDetails>
<Employee Name='John'>ID123</Employee>
<Employee Name='Ken'>ID222</Employee>
</EmployeeDetails>
```

In this case, the repeating element is Employee. From the workflow, the user was unable to view the Name and ID values in runtime.In the above case, the repeating Element being Employee, from the workflow, the user was

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

unable to view the Name and ID values in runtime. In order to view the values, the user had to build the schema in the following pattern.

```
<EmployeeDetails>
<Employee>
<Emp Name='John'>ID123
</Emp>
</Employee>
<Employee>
<Emp Name='Ken'>ID222
</Emp>
</Employee>
</EmployeeDetails>
```

XML Variable Scenarios

This section explains various scenarios on how to use List XML Variables while:

- Add a New List Item
- Update the List Item that Alerted the Workflow
- Save the List item as Different Version as Draft/Published using XML Variable
- Add New Item to a List Folder using List XML Variable
- Delete an Item from a List Folder using XML Variable
- Move an Item to a Folder using List XML Variable
- Create a New Folder using XML Variable
- Add a New List Item with Grid Items
- Add a New Grid Item to the Existing List Item
- Copy a List Item with Grid Items
- Iterate through the List Item and Grid Item of the List

Add a New List Item

AVEVA Work Tasks allows you to add a new list item to the list using XML List XML variables. Update Variable activity can be used to add/update an item to the list. To add an item you need to select the scope as New from Choose the List Item to Work on options. You can also use List Navigator Activity to set the scope. With the help of a scenario, we will explain this function.

Assume that you have created a list called Purchase Orders with the following fields.

- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

  - Quantity
  - Unit Price
  - Item Total
- Total Purchase amount

Now you want to add items to this list.

To add a new List Item

1. Logon to the **Enterprise Console** and launch **Workflow Designer** page.
2. You need to create an XML List type XML variable for the Purchase Orders list. If the variable is not created, follow the below mentioned procedure to create.
3. Click the **More** button in the **XMLVariables** property from the **Start** Activity Properties.
4. Click **New Variable** to add a new variable.
5. Type the variable name as purchaseorders.
6. Select type as **XML List** from the **Type** drop-down list.
7. Type the List name in the **Select List** field or click the **Query** button to select the list. In this case, select Purchase Orders list.
8. Select **New** as scope from the **Choose the ListItem to work on** options to point to a new item in the Purchase Orders list.
9. Now you can see the XML nodes in a tree-view that are mapped to the controls while creating the list.
10. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window.

    Note: If you set the scope as Current while creating the XML List type XML variable, you can change it in to New using List Navigator activity.
11. Drag and drop **Update Variable** activity to add an item to the list.
12. Click the **More** button in the **Set Variables** property.
13. Click the **Select** link and select the respective field of the XML List type XML Variable created for the list (in this case purchaseorders). The fields displayed in the tree-view are representing the XML nodes of the controls in the list.
14. Select a field for which you want to set the value, from the tree-view and enter the value for the field in the right-hand side text box, or you can use **Value** button to set the value.
15. If you want to set the value, select **enter a value** from the tree-view, and then enter the value. Also you can select variables or contents or any XML Variable nodes directly by selecting the respective node from the tree-view. If you want to combine multiple node values, you can build an expression by selecting **Build Expression**.
16. In the same way, you can set the value for all other fields in the list for the new item. Make sure to specify the values for all mandatory fields of the list.
17. After setting the value for the entire field, click **OK**. When you execute this workflow, a new item will get added based on the value that you have set.

Update the List Item that Alerted the Workflow

AVEVA Work Tasks allows you to update the list item that alerted the workflow using XML List Type XML Variable. Update Variable activity can be used to add/update an item to the list. To update an item you need to select the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

scope as Current from Choose the List Item to Work on options. You can also use List Navigator Activity to set the scope. With the help of a scenario, we will explain this function.

Assume that you have created a list called Purchase Orders with the following fields and alerted the workflow.

- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price
  - Item Total
- Total Purchase Amount

Now you want to update an item in the same list. Follow the procedure.

To update the list item that alerted the workflow

1. Logon to the **Enterprise Console** and go to **Workflow Designer** page.
2. You need to create an XML List type XML variable for the Purchase Orders list. If the variable is not created, follow the procedure mentioned below to create the variable.
3. Click the **More** button in the **XMLVariables** property from the **Start** Activity Properties.
4. Click **New Variable** to add a new variable.
5. Type the variable name as 'purchaseorders'.
6. Select type as **XML List** from **Type** the drop-down list.
7. Type the List name in the **Select List** field or click the **Query** button to select the list. In this case, select 'Purchase Orders' list.
8. Select **Current** as the scope from the **Choose the ListItem to work on** options to point to an item in the Purchase Orders list.
9. Now you can see the XML nodes in a tree-view that are mapped to the controls while creating the list.
10. Then click Save Variable. A message is displayed saying ‘Variable Updated’. Close the window.

    Note: If you set the scope as New while creating the **Xml List** type XML variable, you can change it in to **Current** using **List Navigator** activity.
11. Drag and drop **Update Variable** activity to update an item in the list.
12. Click the **More** button in the **Set Variables** property.
13. Click the Select link and select the respective field of the XML List type XML Variable created for the list (in this case purchaseorders). The fields displayed in the tree-view are representing the XML nodes of the controls in the list.
14. Select a field for which you want to set the value from the tree menu. Then enter the value for the field in the right-hand side text box or click the **Value** button to set the value.
15. If you want to set the value, then select enter a value from the tree-view, and then enter the value. Also you

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

can select variables or contents or any XML variables nodes directly by selecting the respective node from the tree-view. If you want to combine multiple node values, you can build an expression by selecting Build Expression.

16. Similarly, you can set the value for other fields that you want to update. In this case, you are only updating the field Title which has the value Order for Illustrator.
17. Click **OK**. When you execute this workflow, the item gets updated based on the value that you have set.

Save the List item as Different Version as Draft/Published using XML Variable

There are two kinds of lists, version enabled and version not enabled. If your list is version enabled, you can save the item as a different version as published or draft using System Internal variables. This can be done at the time of setting the value for activity.

Assume that you are creating a workflow and it is attached to the out of box list workflow which is version enabled.

To save the list item as different version as draft/published using Xml Variable

1. Right click the **Start** activity and select **Activity Properties** from the options.
2. Click the **More** button in the field.
3. For the version enabled list, you can see the list of schema in a tree-view under **System:Internal**. Nodes include Save As NextVersionAsDraft, NextVersionAsPublished and so on. You can use this nodes to save an item in to another version as draft or published.
4. Use **List Query** and point to an item that you want to update and save as different version. See Setting up the List Query.
5. Then drag and drop **Update Variable** activity. Click **Set Variables** button in the **Set Variables** field.
6. Click the **Select** link and then click the drop-down list that appears.
7. Assume that you want to update the item **Description** having the value 'Workflow Management'. So select **Description** from the drop-down list and enter the value as 'Workflow Management'.
8. Now, you want to save this item as **Next Version** as Draft. For that click the drop-down.
9. Select **NextVersionAsDraft** from the **SaveAs** options under **System:Internal**.
10. Then enter the value as 'True'.
11. Click **OK**. This will update the item Title and also will save the item as next version as draft.

    In the same way you can save any item in to different versions like Next Major Version as Published/Draft, Next Minor Version as Published/Draft while updating the item.

    Note: If you select multiple **SaveAs** conditions and set the value as **True**, the last condition will be taken into account and the item will be saved based on the last condition. Also for the list that is not version enabled, the system will automatically save the item as published.

Add New Item to a List Folder using List XML Variable

A new item can be added to a list folder using list XML Variable. This can be done using Update Variable activity and List Navigator activity.

Follow the steps below to add a new item to a list folder.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Right-click the **Start** activity and select **Activity Properties** from the options.
2. Click the **More** button in the **XMLVariables** field.
3. Create a new variable of type **XML List**. Select the list to which a new item is to be added from the **Select List** drop-down. The XML variable scope should be set as **New**.
4. Drag and drop the List Navigator activity from the List Activities category. List Navigator activity is used to navigate to the folder to which the list item is to be added. You need to set the following properties:
5. **XmlVariable of List Type** - You need to select the XML variable created in the Start activity.
6. **Folder Name** - Select the folder name to which a new list item is to be added.
7. **Scope** - The scope should be set as New.
8. **Enable Errors Encountered Output** - It should be set as **No**.
9. You can set different paths for outputs - Action Completed and Folder Not Found. **Folder Not Found** is given by the activity when the folder is not found in the list.
10. Then drag and drop **Update Variable** activity. Click **Set Variables** button in the **Set Variables** field.
11. Click the **Select** link and then click the drop-down list that appears. Navigate to the Title field of the new variable and set its value. The value for the Title field will be the name of the new item.
12. Click **OK** to save the settings.
13. Publish the workflow. Execute the workflow to create a new item in the folder selected in the **Folder Name** property.

Delete an Item from a List Folder using XML Variable

Items can be deleted from a list folder using the delete option under the Operations node in the System Internal Variables. This can be done in the **Update Variable** activity.

The steps given below uses the same scenario used in the previous topic and considers Purchase Order List for the scenario.

To delete an item using XML Variable

1. In the **Update Variable** activity properties pane, select the **Set Variable** icon.
2. In the Update Variable screen, select the **Delete** option in the Operations node in the System Internal node.
3. Set the Operator Value to **True**.
4. Click **OK** to save it.

To enable an item using XML Variable

1. In the **Update Variable** activity properties pane, select the **Set Variable** icon.
2. In the Update Variable screen, select the **Enable** option in the Operations node in the System Internal node.
3. Set the Operator Value to **True**.
4. Click **OK** to save it.

To disable an item using XML Variable

1. In the **Update Variable** activity properties pane, select the **Set Variable** icon.
2. In the Update Variable screen, select the **disable** option in the Operations node in the System Internal node.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Set the Operator Value to **True**.
4. Click **OK** to save it.

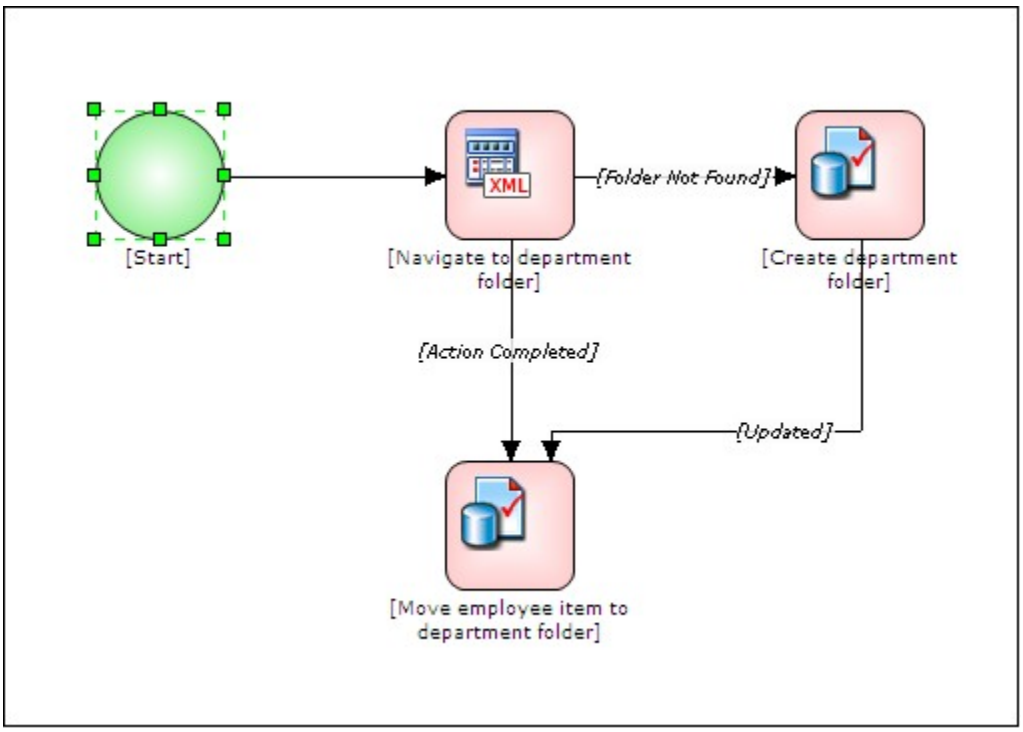
Move an Item to a Folder using List XML Variable

You can move an item to a list folder using list XML variable. This can be done using Update Variable activity and List Navigator activity.

Scenario

An organization has got multiple departments. Each employee information needs to be moved into the respective department folder under the Employee List. If the folder does not exist, the respective department folder needs to be created under the Employee List.

Follow the steps below to move an item to a list folder.



1. Create a list called Employee, with fields as shown below.
   - Title
   - Name
   - Department
   - Experience
   - - Company
   - - Years
2. Go to **Enterprise Console** -> **Workflows**. Create a **List Workflow** and associate it to the **OnCreate** event of Employee List.
3. In the Process Designer, right click the **Start** activity and select **Activity Properties** from the options.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click the **Set Variables** button in the **XMLVariables** field.
5. Create a new variable **EmployeeDepartment** of type **XML List**. Select the **Employee** list from the **Select List** dropdown. The XML variable scope should be set as New. [This XML variable is used in the workflow to check the existence of department folder, and create a new department folder if not found in the Employee List].
6. Drag and drop the List Navigator activity from the List Activities category. List Navigator activity is used to navigate to the folder to which list item is to be moved. You need to set the following properties:

   XmlVariable of List Type - You need to select the XML variable - EmployeeDepartment created in the Start activity.

   Folder Name - Select the department folder name to which the list item is to be moved.

   Scope - The scope should be set as New.

   Enable Errors Encountered Output - It should be set as **No**.
7. You can set different paths for outputs - **Action Completed** and **Folder Not Found**. **Action Completed** output is given by the activity if the XML variable has navigated to the folder specified. **Folder Not Found** is given by the activity when the folder is not found in the list. [**Folder Not Found** output is given by the activity only when the **Enable Error Encountered Output** property is set as **No**].
8. Create the department folder in the list, when the output is **Folder Not Found**. Drag and drop **Update Variable** activity. Click the **Set Variables** icon in the **Set Variables** field.
9. Set the title field of EmployeeDepartment list XML variable to department folder name and System:Internal -> SaveAs -> AsFolder to true.
10. When the folder is found (i.e. Action Completed output), drag and drop Update Variable activity. Click the **Set Variables** button in the Set Variables field.
11. Click the **Select** link and then click the drop-down list that appears. Set the System:Internal -> Operations -> **MoveToFolder** with the department name of the employee. [This is to be done on the list XML variable that points to the employee added].
12. Click **OK** to save the settings.
13. Publish the workflow.
14. Create a new item in the list.
15. The workflow gets triggered. The department name folder gets created in the Employee List and the employee information moves under the respective department folder.

Create a New Folder using XML Variable

Folders can be created in a list using System Internal Variables if the list allows folders to be created. This can be done using **Update Variable** activity.

Follow the steps below to create a folder in Workflow list which allows folders to be created.

To create a new folder using XML Variable

1. Right-click the **Start** activity and select **Activity Properties** from the options.
2. Click the **More** button in the **XMLVariables** field.
3. Create a new variable of type **XML List**. Select list as **Workflow**. The XML variable scope should be set as **New**.
4. Then drag and drop **Update Variable** activity. Click the **Set Variables** button icon in the **Set Variables** field.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Click the **Select** link and then click the drop-down list that appears. Navigate to the Title field of the new variable and set its value. The value for the Title field will be the name of the new folder.
6. To create a new folder, select **AsFolder** from the **SaveAs** options under **System:Internal**. Set its value to true.
7. If your list is version enabled and you want the status of the created folder to be Published, select **Published** from the **SaveAs** options under **System:Internal**. Set its value to true.
8. Click **OK** to save the settings.
9. Publish the workflow. Execute the workflow to create a new folder.

Note: Folder names are not allowed to have special characters like - :"\\"""<>;|*/{}?

Add a New List Item with Grid Items

AVEVA Work Tasks allows you to add list items with grid items to any list using the XML variable. Update Variable activity can be used to perform this task. An XML Iterator type XML variable should be created to add the grid items (child items) and the XML Navigator activity can be used to point to the grid.

This can be explained through the scenario mentioned below.

A list Purchase Orders has been created with the following fields.

- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price
  - Item Total
- Total Purchase amount

From the workflow, you have to add an item along with the child item to the **Purchase Orders** list.

To add a new list item with grid items

1. Logon to the **Enterprise Console** and launch **Workflow Designer** page.
2. You need to create an XML List type XML variable for the Purchase Orders list. If the variable is not created, follow the procedure mentioned below to create the variable.
3. Click the **More** button in the **XMLVariables** property from the **Start** Activity Properties.
4. Click **New Variable** to add a new variable.
5. Type the variable name as 'purchaseorders'.
6. Select type as **XML List** from the **Type** drop-down list.
7. Type the List name in the **Select List** field or click **Select List** button to select the list. In this case select Purchase Orders list.
8. Select **New** as scope from the **Choose the ListItem to work on** options to point to the new item in the list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Now you can see the XML nodes in a tree menu that are mapped to the controls while creating the list. You can also notice that the parent XML nodes and child XML nodes (PurchaseOrderDetails grid) are also displayed.
10. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window.
11. Drag and drop **Update Variable** activity to add an item in the list.
12. Click the **Set Variables** button icon in the **Set Variables** property.
13. Click the **Select** link and select the respective field of the XML List type XML Variable created for the list (in this case purchase orders). The fields displayed in the tree-view are representing the XML nodes of the controls in the list.
14. Select a field for which you want to set the value from the tree- view. Then enter the value for the field in the right-hand side text box. If you want to set the value, select **enter a value** from the tree-view, and then type the value. Also you can select variables or contents or any XML variables nodes directly by selecting the respective node from the tree-view. If you want to combine multiple node values, you can build an expression by selecting **Build Expression**.
15. After setting the value for all the fields in the parent item (controls in the main form) that you want to add, click **OK**. This will add a list item to the list, while executing the workflow, with the fields in the main form.
16. If you want to add an item to the grid (in this case, if you want to add a **Purchase Order Details** grid item), use **Xml Navigator** activity. **Xml Navigator** can be used to point to an item in the grid (**Purchase Order Details** in this case). For this purpose you need to create an **Xml Iterator** type **Xml Variable**. This can be created from the **Start** activity XML variable property. In the XML Variable screen, click **New Variable**.
17. Type the variable name as **purchaseorderdetails**.
18. Select **Xml Iterator** from the **Type** drop-down list.
19. Select **purchaseorders** from the **Xml Variable** drop-down list.
20. Select the repeating node (Qualification) from the **Node Select** drop-down list.
21. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window.
22. Drag and drop **Xml Navigator** activity.
23. Select **purchaseorderdetails** from **Xml Iterator** drop-down list and select **Yes** from **Create New Node** properties to create a new node.
24. Select the appropriate **Location**. The available options are:

    First: Navigates to the first record in the collection.

    Last: Navigates to the last record in the collection.

    Next: Navigates to the next record based on the current location.

    Previous: Navigates to the previous record based on current location.

    Index: Navigates to the record identified in the Index field.
25. Since we have to append the items in **Purchase Order Details** grid, select the option as **Last**.
26. Click **Save**.
27. Drag and drop the **Update Variable** activity to enter the grid field values.
28. Click the **Set Variables** button in the **Set Variables** property.
29. Click the **Select** link and select the respective field of the XML Iterator type XML Variable created for the list (in this case purchaseorderdetails), i.e. grid items. The fields displayed in the tree-view are representing the XML nodes of the controls in the **Purchase Order Details** grid.
30. Select a field for which you want to set the value from the tree- view and type the value for the field in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

right-hand side text box.

31. After setting the value for all the fields in the child item, click **OK**. A new item with grid will be added to the list while executing the workflow.

Note: If you want to add multiple items, use XML Navigator and Update Variable activities for Iterator within the For-Next Loop activities

Add a New Grid Item to the Existing List Item

AVEVA Work Tasks allows you to add list items with grid items to the existing list item using XML variable. Update Variable activity can be used to perform this task. An XML Iterator type XML variable should be created to add the grid items (child items) and the XML Navigator activity can be used to point to the grid. Also you have to use List Query property of List Navigator activity to build the query and to point to the list item to add a new grid.

This can be explained through the scenario mentioned below.

A list Purchase Orders has been created with the following fields.

- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price
  - Item Total
- Total Purchase Amount

From the workflow, now you have to add a grid item to a particular order say Reference No. 456789 in the Purchase Orders list.

To add a new grid item to the existing list item

1. Login to the **Enterprise Console** and go to **Workflow Designer**.
2. You need to create an XML List type XML variable for the Purchase Orders list. If the variable is not created, follow the procedure mentioned below to create a variable.
3. Click the **More** button icon in the **XMLVariables** property from the **Start** Activity Properties.
4. Click **New Variable** to add a new variable.
5. Enter the variable name as **purchaseorders**.
6. Select XML List from the **Type** drop-down list.
7. Enter the List name or click the **Select List** button to select the list. In this case, select Purchase Orders list.
8. Select **New** as scope from the **Choose the ListItem to work on** options to point to the new item in the list.
9. Now you can see the XML nodes in a tree-view that are mapped to the controls while creating the list. You can also notice that the parent XML nodes and child XML nodes (PurchaseOrderDetails grid) are also

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

displayed.

10. Click **Save Variable**. A message is displayed saying 'Variable Updated'. Close the window.
11. Drag and drop the **List Navigator** activity and select **List Query** as the scope.
12. Click **List Query** button in the **List Query** property.
13. Click the **Query** button to build the query for the list Purchase Orders.
14. Select Purchase Orders from the **List Name** drop-down list.
15. Click **Table** field to select the table.
16. Select **SKEListItemDataNV** displayed on the top of the tree-view since you are building a query for an item in the parent table. This will be displayed automatically in the text box.
17. Click the **Column** drop-down list.
18. Select **int1** from the options as it is mapped to the Reference No field.
19. Select **=** operator from the **Operator** drop-down list.
20. Select **Value** from the **Type** drop-down list.
21. Type **456789** in the textbox.
22. Select **End** from the **Condition** drop-down list to end the query.
23. Click **Add** to add the condition in the text field.
24. Here, SKEListItemDataNV is the table name where the list has been stored and t1 is the internal name that has been used for the main table. If there is any child table also, it will display as t2. Int1 is the field name to which you want to point your XML variable and 456789 is the value for that field.
25. Click **OK** and then click **Close**.
26. Click **Update**.
27. If you want to add an item to the grid (in this case, if you want to add a Qualification grid item), use XML Navigator activity. XML Navigator can be used to point to an item in the grid (PurchaseOrderDetails in this case). For this purpose you need to create an XML Iterator type XML Variable. This can be created from the Start activity XML variable property.
28. On the XML Variable screen, click **New Variable**.
29. Type the variable name as purchaseorderdetails.
30. Select **Xml Iterator** from the **Type** drop-down list.
31. Select **purchaseorders** from the **Xml Variable** drop-down list and select the repeating node (Qualification) from the **Node** Select drop-down list.
32. Click **Save Variable**. A message is displayed saying **Variable Updated**. Close the window.
33. Drag and drop **Xml Navigator** activity.
34. Select **purchaseorderdetails** from **Xml Iterator** drop-down list and select **Yes** from **Create New Node** properties to create a new node.
35. Select the appropriate Location. The options available are:

    First: Navigates to the first record in the collection.

    Last: Navigates to the last record in the collection.

    Next: Navigates to the next record based on the current location.

    Previous: Navigates to the previous record based on current location.

    Index: Navigates to the record identified in the Index field.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

36. Since we have to append the items in **Purchase Order Details** grid, select the option as **Last**.
37. Click **Save**.
38. Drag and drop the **Update Variable** activity to enter the grid field values.
39. Click the **Set Variables** button in the **Set Variables** property.
40. Click the **Select** link and select the respective field of the XML Iterator type XML Variable created for the list (in this case purchaseorderdetails), i.e. grid items. The fields displayed in the tree-view are representing the XML nodes of the controls in the PurchaseOrderDetails grid.
41. Select a field for which you want to set the value from the tree-view. Then enter the value for the field in the right-hand side text box. If you want to set the value, then select enter a value from the tree-view, and then enter the value. Also you can select variables or contents or any XML variables nodes directly by selecting the respective node from the tree-view. If you want to combine multiple node values, you can build an expression by selecting **Build Expression**.
42. After setting the value for all the fields in the child item, click **OK**. Now, a grid item will be added to the list while executing the workflow.

    Note: If you want to add multiple items, use XML Navigator and Update Variable for Iterator within the For-Next Loop activities.

Copy a List Item with Grid Items

AVEVA Work Tasks allows you to copy a list item with grid items to any list, but both the lists should have the same schema. You can perform this task with the help of Update Variable activity. To iterate through the grid items, you need to create XML Iterator type XML variable for the grid. This feature can be explained with the help of a scenario.

Telematics is an organization which maintains a Purchase Orders list to keep a track of all the purchase orders raised by the company.

The Purchase Orders contains details like:

- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price
  - Item Total
- Total Purchase Amount

Telematics also maintains another list called **Approved Purchase Order** which has the same schema of the **Purchase Order** list. Now you want to copy an item with grid (Purchase Order approved) to the Approved Purchase Order list when the item is approved.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To copy a list item with grid items

1. Login to the **Enterprise Console** and go to **Workflow Designer**.
2. You need to create an XML List type XML variable for the Purchase Order list. If the variable is not created, follow the procedure mentioned below to create the variable.
3. Click the **More** button in the **XMLVariables** property from the **Start Activity Properties**.

   Here, for Purchase Order list you need to create an XML List type XML variable to work with the current purchase order item and XML Iterator type XML variable to loop through the child/grid nodes of the current purchase order item.
4. Click **New Variable** to add a new variable.
5. Type the variable name as 'purchaseorders'.
6. Select type as **XML List** from the **Type** drop-down list.
7. Type the List name in the **Select List** field or click the **Select List** button to select the list. In this case select **Purchase Orders** list.
8. Select **Current** as the scope from the **Choose the ListItem to work on** options to point to the an item in the list.
9. Now you can see the XML nodes in a tree-view that are mapped to the controls while creating the list. You can also notice that the parent XML nodes and child XML nodes (PurchaseOrderDetails grid) are also displayed.
10. Click **Save Variable**. A message is displayed saying ‘Variable Updated’.
11. Now, you need to create XML Iterator type XML variable to iterate through the child nodes(in this case, PurchaseOrderDetails). For that, click **New Variable** from the XML Variable screen.
12. Enter the variable name as purchaseorderdetails.
13. Select **XML Iterator** as type from the **Type** drop-down list.
14. Select **purchaseorders** from the **XML Variable** drop-down list.
15. From the **Node Select** drop-down, select the repeating node, i.e. PurchaseOrderDetails.
16. Click **Save Variable**. A message is displayed saying ‘Variable Updated’.
17. In the same way you have to create XML variables for Approved Purchase Order list. For Approved Purchase Order list, you need to create XML List type XML variable to add an item and a XML Iterator type XML variable to map to the grid items of the new purchase order item.
18. Click **New Variable** to add a new variable, from the **XML Variable** property of **Start** activity.
19. Enter the variable name as approvedpurchaseorders in the **Variable Name** field.
20. Select XML List as type from the **Type** drop-down list.
21. Enter the List name in the **Select List** field or click the Select List button to select the list. In this case, select Approved Purchase Orders list.
22. Select **New** as scope from the **Choose the ListItem to work on** options to point to the new item in the list.
23. Now you can see the XML nodes in a tree-view that are mapped to the controls while creating the list. You can also notice that the parent XML nodes and child XML nodes (PurchaseOrderDetails grid) are also displayed.
24. Click **Save Variable**. A message is displayed saying ‘Variable Updated’.
25. Now, you need to create **XML Iterator** type XML variable to iterate through the child nodes. For that, click **New Variable** from the XML Variable screen.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

26. Enter the variable name as **approvedpurchaseorderdetails**.
27. Select **XML Iterator** as type from the **Type** drop-down list.
28. Select **approvedpurchaseorders** from the **XML Variable** drop-down list.
29. From the **Node Select** drop-down, select the repeating node, i.e. PurchaseOrderDetails.
30. Click **Save Variable**. A message is displayed saying 'Variable Updated'.
31. Drag and drop **Update Variable** activity to set value for the parent nodes or the item.
32. Click the **Set Variables** button in the **Set Variables** property.
33. Click the **Select** link and select the respective field of the XML List type XML Variable created for the list (in this case approvedpurchaseorders). The fields displayed in the tree-view are representing the XML nodes of the controls in the list.
34. Select a field for which you want to set the value from the tree-view under approvedpurchaseorders. The selected variable for the field **Title** will be displayed in the text box as '**XmlVariables.approvedpurchaseorders.ApprovedPurchaseOrders.Title**.'
35. Enter the value for the field in the right-hand side text box. Select the respective field of the purchaseorders XML variable (purchase order item to be copied). The selected XML variable for the field will be displayed as '**XmlVariables.purchaseorders.PurchaseOrders.Title'**. Set the value for all other fields in the same way.
36. After setting the value for all the fields in the parent form (controls in the main form) that you want to add, click **OK**. This will add a list item to the new list (Approved Purchase Order), while executing the workflow, with the values in the current Purchase Orders item.
37. Next, you need to iterate through the grid/child items of the current purchase order and to add the child items in the new list (Approved Purchase Order). Drag and drop ForEach-Loop activity to loop through the grid items.
38. Select **XML Iterator** type XML variable that you have created for the current Purchase Order from the **XML Iterator** drop-down list (in this case, purchaseorderdetails).
39. If you want to add an item to the grid (in this case, if you want to add a purchase order details grid item), use **XML Navigator activity**. XML Navigator can be used to point to an item in the grid (purchaseorderdetails in this case).
40. Drag and drop **XML Navigator** activity.
41. Select approvedpurchaseorderdetails from **XML Iterator** drop-down list and select Yes from Create New Node properties to create a new node.
42. Select the appropriate **Location**. The available options are:

- First: Navigates to the first record in the collection.
- Last: Navigates to the last record in the collection.
- Next: Navigates to the next record based on the current location.
- Previous: Navigates to the previous record based on current location.
- Index: Navigates to the record identified in the Index field.

    1. Since we have to append the items in **Purchase Order Details** grid, select the option as **Last**.
    2. Click **Save**.
    3. Drag and drop **Update Variable** activity to set the value for the grid item fields of the new list item.
    4. Click the **Set Variables** button in the **Set Variables** property.
    5. Click the **Select** link and select the respective field of the XML List type XML Variable created for the list

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

(in this case ApprovedOrderDetails). The fields displayed in the tree-view are representing the XML nodes of the controls in the list.

6. Select the XML variable **PurchaseOrderDetails** from the tree-view. You can either select PurchaseOrderDetails node or select all child nodes separately. In this case, you are selecting the PurchaseOrderDetails node. The variable will be displayed in the text box as '**XmlVariables.approvedpurchaseorderdetails.PurchaseOrderDetails'**.
7. Set the value in the right-hand side text box. Select the respective field of the purchaseorderdetails XML variable (purchase order item to be copied) from the tree-view. The selected variable will be displayed in the text box as '**XmlVariables.purchaseorderdetails.PurchaseOrderDetails'**.
8. After setting the value for the fields in the child item that you want to add, click **OK**. This will add a grid item to the new list (Approved Purchase Order), while executing the workflow, with the values in the current (Purchase Order) list.

   Note: You can also select each node in the grid separately and set the value for each node.
9. Complete the loop with a **Next** activity and finish the workflow with appropriate link conditions.

Iterate through the List Item and Grid Item of the List

AVEVA Work Tasks allows you to loop through the list items and grid items of a list. An XML List Iterator type XML variable can be used to iterate through the list items, where XML Iterator type XML variable can be used to iterate through the grid items of a list. This feature can be explained with the help of a Purchase Order list scenario.

The Purchase Orders contain details like:

- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price
  - Item Total
- Total Purchase Amount

Now you want to loop through the approved purchase order items (items with status as Approved) as well as its child items.

To iterate through the list item and grid item of the list

1. Login to the Enterprise Console and go to Workflow Designer.

   Note: As a first step, you need to create an XML List type XML variable for the Purchase Orders list. If the variable is not created, follow the procedure mentioned below to create a variable.
2. Click the **More** button in the **XMLVariables** property from the Start Activity Properties.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Click **New Variable** to add a new variable.
4. Enter the variable name as **purchaseorders**.
5. Select XML List from the **Type** drop-down list.
6. Enter the List name or click the **Select List** to select the list. In this case, select Purchase Orders list.
7. Select the scope from the Choose the ListItem to work on options. Later you can change the scope of the variable to List Query to query the approved purchase orders using List Navigator activity.
8. Click **Save Variable**. A message is displayed saying 'Variable Updated'.
9. Now, you need to create XML List Iterator type XML variable to iterate through the approved purchase order items. For that click **New Variable** from the XML Variable screen.
10. Type the variable name as **purchaseorderiterator**.
11. Select **XML List Iterator** from the Type drop-down list.
12. Select **purchaseorders** from the **List XML Variable** drop-down list.
13. Click **Save Variable**. A message is displayed saying 'Variable Updated'.
14. Now, you need to create XML Iterator type XML variable for the purchaseorderiIterator (XML List Iterator type) XML variable to iterate through the child nodes of the currently iterating purchase order item. For that click New Variable from the XML Variable screen.
15. Type the variable name as **purchaseorderdetails**.
16. Select **XML Iterator** from the **Type** drop-down list.
17. Select **purchaseorderiterator** from the **XML Variable** drop-down list.
18. From the **Node Select** drop-down LIST, select the repeating node, i.e. PurchaseOrderDetails.
19. Click **Save Variable**. A message is displayed saying 'Variable Updated'.
20. Use **List Navigator** activity to query the approved purchase order items.

    In this case, the following query is used to retrieve the approved purchase orders. **SKEListItemDataNV(t1).nvarchar3 = 'Approved'**, where **SKEListItemDataNV** is the table used to store the list items and **nvarchar3** is the table column mapped to the Status field.
21. Use ForEach-Loop and Next Loop activities to loop through the approved purchase order items.
22. Select **purchaseordersiterator** from the **XML Iterator** drop-down list of ForEach Loop activity.
23. Use any action as per the requirement to retrieve the data. You need to access the value of the parent nodes from the purchaseordersiterator XML variable.
24. Use another set of ForEach-Loop and Next activities within the existing loop to iterate through the grid items (PurchaseOrderDetails).
25. Select **purchaseorderdetails** from the XML Iterator drop-down list of ForEach Loop activity.
26. Use any action as per the requirement to retrieve the data. You need to access the value of the parent nodes from the purchaseorderdetails XML variable.

List Views

List View provides a host of options to view Lists according to your preferences. It allows you to customize views according to specific requirements. You can create and use views to see the items in a list that are most important to you or that best fit your purpose. For example, you can create views of the list items that were created most recently, or the list items that apply to a specific department, or of the items created by one

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

person. There should be at least one default view created.

You can use this feature to set different views like Filtered view, Read-only View, Group by view etc. You can make these views either public or private by adding the attributes. These attributes allow you to set permissions for the created views, thereby ensuring confidentiality in your organization.

For example, a task list has several views, such as the tasks due today, the tasks assigned to the user, all tasks, and so on.

List View Introduction

List View provides a host of options to view Lists according to your preferences. It allows you to customize views according to specific requirements. You can create and use views to see the items in a list that are most important to you or that best fit your purpose. For example, you can create views of the list items that were created most recently, or the list items that apply to a specific department, or of the items created by one person. There should be at least one default view created.

You can use this feature to set different views like Filtered view, Read-only View, Group by view etc. You can make these views either public or private by adding the attributes. These attributes allow you to set permissions for the created views, thereby ensuring confidentiality in your organization.

For example, a task list has several views, such as the tasks due today, the tasks assigned to the user, all tasks, and so on.

Work with List Views

You can create different list views, add filters, and manage views.

About Creating List Views

This section provides detailed procedures for creating a new list view. After creating a new list view, you can start customizing the view according to your preferences. All the views can be selected from the drop-down list in the right-hand top corner of the List View screen.

In order to understand the different options in List View, let's use a working scenario.

Scenario

The Telematics Company is a Product Support Service Provider and uses Lists in their service for Product Support. They have ten long-standing clients to whom they have to cater product support services through Email, Phone, and Web. Some of these clients are tagged as 'Gold' clients to whom, Telematics readily provides support over the phone. While some other clients are tagged as 'Silver' clients to whom Telematics provides support through Email. Similarly, there is a host of classification based on various parameters to categorize the customer related information. In order to manage the customer related information, Telematics has created **Support Incident** lists using Lists in which, the organization categorizes the Support Incidents information under specific columns. Here is a table that gives you the column name and its description:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Column Name	Description
Number	Specific number assigned to the incident
Title	Title of the Support incident
Service User	Name of the user who is using this service
Service	The name of the product used by the organization – BPM.NET 2006, BPM.NET 2007, BPM.NET 2009, BPM.NET 2009 August Update
Service Level	The type of customer - Gold or Silver
Incident Status	Status of the Support Incident which is displayed as - In progress, Resolved, Closed, New, and Level 2
Support Person	Personnel working on this issue
Business Unit	Name of the customer organization
Incident Type	Type of Support Incident - Query or Development
Incident Origin	Origin of the Support Incident - By Email, Web, Phone

Telematics team likes to use List View feature to view all the information related to these clients in customized views.

Each of the upcoming topics gives you a walk-through on how Telematics achieves these tasks.

Create a New View

This section provides detailed procedures for creating a new view. The Telematics Representative working with lists of Support Incidents wants to create a view to see incidents which have originated through emails. The representative can create this view, by selecting the Create View option.

Note: The Support Incidents lists here refers to the group of lists created by the Telematics organization for the purpose of this tutorial. Support Incident is not the system generated list item.

To add a new view

1. In the **Enterprise Console**, click menu icon, click **Lists**, and then click **Support Incidents**. The **Support Incidents** list items page appears. By default, the All Items option is displayed in the View drop-down field and hence all the list items related to Support Incidents are displayed in this view.
2. In the **View** drop-down list field, select **Create View**, and then click **Go**. The **Create View** screen appears.

   Note: For every list created through List Infrastructure, a default field must be configured. For example, the Order ID field becomes the default field in an Order list and the Department Name becomes the default field in a Department list. While creating a list view, in the available columns, the user will not have an option to uncheck these fields as these are the default fields. Here, we are using the **Title** field as the default field for the **Support Incidents** lists. Hence, the list view users cannot deselect or select this display column name.
3. Type the name **By Email** for this view.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Check the **Make this the default view** check box. This option can be used to make this view as the default view for your Support Incidents lists.
5. Select the **Public** option if you want to display this view to all the users.
6. In the **Grid Columns**, check the columns **Number**, **Service User**, **Incident Status**, **Incident Type**, **Incident Origin**, **Created By**, **Created DateTime**, **Owner**, and **Status** from the **Display Column Name** field to display specific information related to these column headings in this view.
7. Similarly, select the order of the display columns from the **Display Order Number** field to display the columns in the required order.

Adding Filters

1. You can even add filters to get specific information under the selected columns. In this scenario, we will add a filter to get the incidents originated from emails. In the Filter Columns field, the Show all items in this view is selected by default. If you want to add any filters, then you will have to click the **Show items only when the following is true** option. Only then the Filter options are displayed.
2. Click the **Show items only when the following is true** radio button in the Filter Columns field. You can see the options appearing in the field.
3. Select the **Incident Origin** column from the column drop-down list.
4. Select the **is equal to** filter condition from the filter condition drop-down list.
5. Select **Value** from the Value field.

   If the variable option is selected from the Value column, the user should ensure to specify the appropriate data type in the condition field. If the user specifies inappropriate data type in the condition field, a validation message appears asking the user to modify the data type.
6. Type **Email** in the condition field.
7. Click **Create View**.

   Note: While creating a list view, the user may come across certain exceptions owing to reasons like assigning inappropriate values, repeating the same column in the sort expression, database connectivity errors etc. An exception message appears in a pop-up window highlighting the possible reasons for the exception.

To view the lists under the By Email View

- Select the **By Email** view from the View drop-down list in the Support Incidents list screen. You can see that the information has been arranged according to the column specifications and filter conditions assigned by you. All the incidents originated from Email have been displayed in the Incident Origin column.

  You can see that all the different columns specified by you are displayed in this view. You can also see that the Title column appears in this view. This is because, the Title column is one of the default list column in the Support Incident list items.

Note: Item selection is not retained, when a view is applied to the list items and the position of the list item is changed because of the applied view. For example: If the view applied is sorted by Date and Time, then the position of the selected list item will change as the date modified for the list item also changes.

Create a Default View

This section provides detailed procedures for creating a default view.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Default view can be created to view the same view every time the user logs into the portal.

Note: There can be only one default view for the Public view and one for the Private view. If there are any existing private default views and the user wants to create a new default private view, then the user can either overwrite this new view as a default view or create a new view without selecting the default view.

Scenario

The Telematics Support representative wants to see upon login all the support issues related to the 'Gold' customers, since they are given top priority. The representative can create a default view with the following specifications and view the same every time they log into the system.

To create a default view

1. In the **Enterprise Console**, click the **Menu** button, click **Lists**, and then click **Support Incidents**. The **Support Incidents** list items page appears.
2. Go to the **View** drop-down list field and select **Create View**.
3. Type the name **Gold** for this view of the list.
4. In the **Create New View** screen, check the **'Make this the default view'** check box.
5. Since, you had created the By Email view as default view, the system asks you if you want to overwrite it. Select **OK**.
6. Select the **Public** option if you want to display this view to all the users.
7. In the **Grid Columns**, check the columns **Number**, **Service Level**, **Incident Status**, **Support Person**, **Business Unit**, **Service**, **Incident Type**, **Incident Origin**, and **Created By** from the **Display Column Name** field to display specific information related to these column headings in this view.
8. Similarly, select the order of the display columns from the **Display Order Number** field to display the columns in a particular order.

To add Filters

You can add filters to get specific information under the selected columns. In this scenario, we will add a filter to get the support issue list related to the **Gold** customers under the Service Level .

1. Click the **Show items only when the following is true** radio button in the 'Filter Columns' field. You can see the options appearing in the field.
2. Select the **Service Level (SupportIncidents)** column, **Is equal to**, and **Value** from the drop-down lists.
3. Type **Gold** in the filter condition field.
4. Click **Create View**.

To view the Default view

- Select the **Gold** view from the **View** drop-down list in the Support Incidents list screen. All the list items under the Service Level related to the **Gold** customers will be listed.

Create a Public View

This section provides detailed procedures for creating a Public view.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Public view option can be chosen when you want to allow all the users to view the list items.

Scenario

Consider the case of Support Incident List items. The Telematics representative wants to create a view that can be displayed to all the users. The Administrator can select the Public View option to make all the three views accessible to the end users.

To add a public view, refer to Creating a New List View.

Create a Private View

This section provides detailed procedures for creating a Private view.

The Private view option can be used if you want to create a view according to your personal choice. After this option is selected, the other users will not be able to see this view option in the View drop-down list.

Scenario

Consider the case of Support Incident List items. The Telematics representative wants to create a view consisting of the incidents which are of the **In Progress** status. The representative can select the Private view option to achieve this task.

To add a private view

1. In the **Enterprise Console**, click the **Menu** button, click **Lists**, and then click **Support Incidents**. The **Support Incidents** list items page appears.
2. Go to the **View** drop-down list field and select **Create View**.
3. Type the name **In Progress** for this view of the list.
4. Click the **Private** radio button in the **Scope of the View** to make this view inaccessible to other users.
5. Select the **Grid Columns Service User**, **Incident Status**, **Business Unit**, **Service**, **Service Level**, **Incident Type**, **Incident Origin**, and **Modified By**.
6. In the **Filter Column** field, select **Incident Status**, **Is Equal To**, and **Value** from the filter option drop-down lists.
7. Type **In Progress** in the filter condition field.
8. Click **Create View**.

To view the Private view

Login as the user who has created the private view and select the **InProgress** view from the **View** drop-down list in the Support Incidents list screen. All the list items which have the incident status as In Progress will be listed in this view. If you login as a different user then you can see all the other views except this view.

Add Filters

This section provides detailed procedures to add filters to the views.

Filters can be added to further classify the information in the lists. The user can create filters using the filter expressions.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scenario

Consider the case of the Support Incident list items. The Support Representative wants to create a **High Priority** view to view all the support incidents related to the Business Unit, where the Incident Origin is Email and the Service Level is Gold. To achieve this, the representative can create a view and build a filter expression. When creating a view, the user has an option to create filters for any of the user defined and system generated fields. There are different filter conditions that can be used for creating the filters. Below are the filter conditions that can be used.

- is equal to
- is not equal to
- is greater than
- is less than
- is greater than or equal to
- is less than or equal to
- contains

To add filters to the views

1. In the **Enterprise Console**, click the **Menu** button, click **Lists**, and then click **Support Incidents**. The **Support Incidents** list items page appears.
2. Go to the **View** drop-down list field and select **Create View**.
3. Type the name by **High Priority** for this view of the list.
4. In the **Filter Columns** field, click **Show items only when the following is true** radio button.
5. Select **Business Unit** from the **Show the items when column** drop-down list.
6. Select the **is equal to** filter condition from the filter condition drop-down list.
7. Select the **Value** from the **Value** drop-down list.
8. Type **Telematics** in the empty field.
9. Select the **and** radio button to add additional filters. By default, the Or button is selected.
10. Add another filter by selecting **Incident Origin** from the **When column** drop-down list.
11. Select the **is equal to** filter condition from the filter condition drop-down list.
12. Select the **Value** from the **Value** drop-down list.
13. Type **Email** in the empty field.
14. Click the **Show More Columns** to add more filters.
15. Another set of filter conditions appears in the filter field.
16. Click the **And** radio button.
17. In the new filter rows, select **Service Level** from the **When column** drop-down list.
18. Select the **is equal to** filter condition from the filter condition drop-down list.
19. Select the **Value** from the Value drop-down list.
20. Type **Gold** in the empty field.
21. Click **Create View**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To view the filtered view

- Select the **High Priority** view from the **View** drop-down list in the Support Incidents list screen. You can see the grid columns according to your filter specifications.

Filter Grouping

This option is used to execute the filter conditions in a specific order by applying the parenthesis at the appropriate position in the filter conditions.

Scenario

Consider that the Telematics Support Manager wants to create a view for "Open Incidents for GOLD AMC" with the following conditions.

- ServiceLevel = 'GOLD'
- Incident Status either "Waiting" or "InProgress"

The condition can be as given below:

ServiceLevel = 'GOLD' and Incident Status either Incident Status is "Waiting" or "InProgress"

In this filter, we first need to evaluate the condition Incident Status is "Waiting" or "InProgress". Later, we need to join it with the ServiceLevel = 'GOLD'. The filter condition is given below:

((ServiceLevel = 'GOLD') and (Incident Status = "Waiting" or Incident Status = "InProgress"))

If we do not specify the order of execution (by putting the parenthesis) the condition might get evaluated as follows and may result in invalid data:

First the filter gets executed for the condition given below :

ServiceLevel = 'GOLD' and Incident Status = "Waiting"

And then it gets executed for the condition given below:

Incident Status ="InProgress"

Hence, it is very important to add the parenthesis to get the required grouping.

To create a Group Filter

1. Create a new view with the name **Open Incidents for Gold ABC**. Follow steps 1- 4 from the previous section to create a new view.
2. In the **Filter Columns** field, click **Show items only when the following is true** radio button.

   The Show items only when the following is true fields appear on the screen.
3. Select **Business Unit** from the **Show the items when column** drop-down list.
4. Select the **is equal to** filter condition from the drop-down list.
5. Select the **Value** from the drop-down list.
6. Type **ABC Corporation** in the empty field.
7. Select the **and** radio button to add additional filters. By default, the **Or** button is selected.
8. Add another filter by selecting **Service Level** from the **When column** drop-down list.
9. Select the **is equal to** filter condition from the drop-down list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

10. Select the **Value** from the drop-down list.
11. Type **Gold** in the empty field.
12. Add another filter by selecting **Incident Status** from the **When column** drop-down list.
13. Select the **is equal to** filter condition from the drop-down list.
14. Select the **Value** from the drop-down list.
15. Type **Waiting** in the empty field.
16. Click the **Show more Columns** to get more columns.
17. Click **Or** and add another filter for the **Incident Status**, with value as **Inprogress**.
18. Click the **Set Filter Order** tab to get all the filter conditions.
19. To group the filter conditions of the Incident Status, add an open parenthesis '(' after And and a close parenthesis ')' at the end of the filter conditions.

    For example: **Business Unit = ABC Corporation And (Incident Status = Waiting Or Incident Status = Inprogress)**
20. Click the **Validate** button to validate the expression. If any parenthesis mismatch is found, an error message pops up. If no errors are found, then the Validation success message pops up on your screen.
21. Click **Create View** to view the Incident Status of the ABC Corporation.
22. To see the view, choose the **Open Incidents for Gold ABC** view from the Create View drop-down list.

Use Date Time Variable to Filter Columns

Let us understand this by using a scenario. The Telematics Support Manager wants to create a view to get all the in progress and closed incidents created during the current year. To achieve this, the Manager has to follow the procedure given below:

To use Date System Variable to Filter Columns:

1. Create a new view with the name **In Progress and Closed Issues**. Follow steps 1- 4 from the previous section to create a new view.
2. In the **Filter Columns** field, click **Show items only when the following is true** radio button.
3. Select the **and** radio button to add additional filters. By default, the **Or** button is selected.
4. Add another filter by selecting **Incident Status** from the **When column** drop-down list.
5. Select the **is equal to** filter condition from the drop-down list.
6. Select the **Value** from the drop-down list.
7. Type **In Progress** in the empty field.
8. Select the **and** radio button to add additional filters. By default, the Or button is selected.
9. Add another filter by selecting **Incident Status** from the **When column** drop-down list.
10. Select the **is equal to** filter condition from the drop-down list.
11. Select the **Value** from the drop-down list.
12. Type **Closed** in the empty field.
13. Select the **And** radio button to add additional filters.
14. Add another filter by selecting **CreatedDateTime** from the When column drop-down list.
15. Select the **is equal to** filter condition from the drop-down list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

16. Select the **Variable** from the drop-down list.
17. Select the **Variable.Sys.This Year** from the variable drop-down.
18. Click the **Create View** button at the end of the screen.
19. To see the view, choose the **In Progress and Closed Issues** view from the Create View drop-down list.

Note: The date is stored in the UTC format in the database. Hence, the user needs to provide the date in UTC format in the filter area. Since, UTC date contains time too, the user should provide the exact time to filter with 'Equals to' operator. The date format for Microsoft SQL Server should be mm/dd/yyyy.

Sort the Columns

This section provides detailed procedures to sort the order of the grid columns.

This option can be used to sort the columns according to your preference.

Scenario

Consider the case of the Support Incident List View. The Telematics Support Manager wants to create a tasks view by sorting the incident numbers in descending order. The Manager can use the Sort Columns option to get this view.

To sort the columns in the views

1. In the **Enterprise Console**, click menu icon, click **Lists**, and then click **Support Incidents**. The **Support Incidents** list items page appears.
2. Go to the **View** drop-down list field and select **Create View**.
3. Type the name **Tasks** for this view of the list.
4. In **Sort Columns**, select the **Number** option from the **Sort by** field.
5. Select **Descending** order radio button.
6. If you want to sort other columns, you can click the **Show More Columns** and make the necessary selections.
7. Click **Create View**.

Note: Sorting can be done to a maximum of 10 columns.

To view the Tasks view

- Select the **Tasks** view from the **View** drop-down list in the Support Incidents list screen. You can see the grid columns according to your specifications.

System Variables

List infrastructure provides default system variables that can be used in Filters while creating a view. They can be categorized as below:

- Date Time Variables:

These variables can be used to compare or filter on DateTime Columns.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- User variables:

These variables can be used to compare filter on loggedin user ID context.

- Workflow Status:

These variables can be used to filter list items based on workflow instance status.

Date Time Variables

The following datetime variables allow the user to filter datetime columns of the list:

All the datetime variables in this topic have been explained assuming the present date as 19th, month as January, and year as 2011.

Variable to filter datetime columns for Today

This variable is used to compare datetime columns for today. For example, the Support Manager wants to get all the incidents created today. To achieve this the Manager has to specify the **Created DateTime** variable equal to **Variable.Sys.Today** in the view creation window.

By specifying this variable, the user will be able to view all the support incidents created today.

Variable System Current Week

This variable is used to compare datetime columns for the current week. For example, the Support Manager wants to get all the incidents created today. To achieve this the Manager has to specify the **Created DateTime** variable equal to **Variable.Sys.Current Week** in the view creation window.

By specifying this variable, the user will be able to view all the support incidents created during the current week.

Variable System Last 10 days

This variable is used to compare datetime columns for the last 10 days. For example, the Support Manager wants to get all the incidents created during the last 10 days. To achieve this the Manager has to specify the **Created DateTime** variable equal to **Variable.Sys.Last 10 Days** in the view creation window.

By specifying this variable, the user will be able to view all the support incidents created during the last 10 days.

Variable System Current Month

This variable is used to compare datetime columns for the current month. For example, the Support Manager wants to get all the incidents created this month. To achieve this the Manager has to specify the **Created DateTime** variable equal to **Variable.Sys.Current Month** in the view creation window.

By specifying this variable, the user will be able to view all the support incidents created during this month.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Variable System Lasts 2 Months

This variable is used to compare datetime columns for the last 2 months. For example, the Support Manager wants to get all the incidents created during the last two months. To achieve this the Manager has to specify the **Created DateTime** variable equal to **Variable.Sys.Last 2 Months** in the view creation window.

By specifying this variable, the user will be able to view all the support incidents created during the last 2 months.

Variable System This Year

This variable is used to compare datetime columns for the current year. For example, the Support Manager wants to get all the incidents created this year. To achieve this, the Manager has to specify the Created DateTime variable equal to Variable.Sys.This Year in the view creation window.

By specifying this variable, the user will be able to view all the support incidents created this year.

Date Range logic for each Date System variable

Variable	From Date	To Date
Variable.Sys.Today		Current date.
Variable.Sys.Current Week	Start date of current week. (Start of the week is as per the culture.)	End date of current week.
Variable.Sys.Last 10 Days	10 days prior to current date.	Current date.
Variable.Sys.Current Month	First day of current month.	Last day of current month.
Variable.Sys.Last 2 Months	First day of the month which is two months prior to the current month.	Last day of the month which is one month prior to the current month.
Variable.Sys.This Year	First day of the current year.	Last day of the current year.

Workflow Variables

The following variables allow the user to filter the columns according to the workflow instance status of incidents:

Variable System Workflow Status Completed

This variable is used to compare the completed workflow instance status. For example, the Support Manager wants to get all the incidents which contain workflows with completed status. To achieve this the Manager has to specify the **Workflow Status** variable equal to **Variable.Sys.WFStatus.Completed** in the view creation window.

By specifying this variable, the user will be able to view all the support incidents which contain workflows with completed status.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Variable System Workflow Status InProgress

This variable is used to compare the InProgress workflow instance status . For example, the Support Manager wants to get all the incidents which contain workflows which show in the progress status. To achieve this, the Manager has to specify the **Workflow Status** variable equal to **Variable.Sys.WFStatus.InProgress** in the view creation window.

By specifying this variable, the user will be able to view all the support incidents which contain workflows which show in the progress status.

Variable System Workflow Status Errors

This variable is used to compare the Errors workflow instance status. For example, the Support Manager wants to get all the incidents which contain workflows which contain errors in them. To achieve this, the Manager has to specify the **Workflow Status** variable equal to **Variable.Sys.WFStatus.Errors** in the view creation window.

By specifying this variable, the user will be able to view all the support incidents which contain workflows which are erroneous.

User Variables

This variable allows the user to filter the columns according to the logged in user.

Variable System LoggedInUser

This variable is used to filter the columns based on the logged in user. For example, the Support Manager wants to get all the incidents created/owned by a particular logged in user. To achieve this, the Manager has to specify the Logged In User variable equal to **Variable.Sys.LoggedInUserId** in the view creation window.

By specifying this variable, the user will be able to view all the support incidents related to the logged in user. For this the user has to log in with the logged in user credentials.

Group Columns

This section provides detailed procedures to group the columns in a particular order in the view.

Columns can be grouped for better organizing of your list item columns. You can group columns of similar or related information. You can group the columns in expanded mode or collapsed mode.

Scenario

Consider the case of the Support Incident list items. When the Telematics Support Manager logs in and navigates to the support incident list, there can be many records in the list. The Manager can group similar columns of information for display either in the collapsed mode or in expanded mode.

To group columns in the view

1. In the **Enterprise Console**, click the **Menu** button, click **Lists**, and then click **Support Incidents**. The **Support Incidents** list items page appears.
2. Go to the **View** drop-down list field and select **Create View**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Type the name **Status** for this view of the list.
4. In the **Group Columns** field, select the **Collapsed** group mode. You can see that the Group Columns field gives a provision to give three group column conditions. To give more conditions, click **Show More Columns**.
5. Select the **Incident Status** from the **Group Column** by drop-down list.
6. Select the **Ascending** or **Descending** option.
7. In the **then group by** drop-down list, select **Incident Type** and select **Descending order**.
8. Click the **Show More Columns** button to include another group condition **Incident Origin** and select the **Ascending order**.
9. Click **Create View**.

To view the group columns view

- Select the **Status** view from the **View** drop-down list in the Support Incidents list screen. You can see the specified grid columns grouped according to your specifications.

  If you had selected the **Expanded** mode, then the Incident Information would be in hierarchical order.

Display All Items

This section provides detailed procedures to display all the items in the list view.

This option can be used to allow all the list items to be displayed to the user. Lists can be grouped in a folder and upon opening the lists, only the lists outside the folder are displayed by default. By checking this option, even the lists within a folder can be viewed upon opening the lists.

To display all items in the view

1. In the **Enterprise Console**, click the **Menu** button, click **Lists**, and then click **Support Incidents**. The **Support Incidents** list items page appears.
2. Go to the **View** drop-down list field and select **Create View**.
3. Type the name **All Support Issues** for this view of the list.
4. Select the **Display All Items** check box.
5. Click **Create View**.

To view the displaying all items view

- Select the **All support issues** view from the **View** drop-down list in the Support Incidents list screen. You can see all the lists, including the ones within a folder.

Create a Read-Only View

This section provides detailed procedures to create a read only view.

If you want to restrict the users from editing or deleting the list items, then you can check this option. After this option is selected, the other users will not be able to edit or delete the list items in this view.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scenario

The Telematics Representative wants to make the Status view as the Read-only view in order to restrict the users from editing or deleting the list items in this view.

To create read only view

1. In the **Enterprise Console**, click the **Menu** button, click **Lists**, and then click **Support Incidents**. The **Support Incidents** list items page appears.
2. Go to the **View** drop-down list field and select **Create View**.
3. Type the name **Status** for this view of the list.
4. Select the **Read Only** check box to disable the editing and deleting list view options for the end-user.
5. Click the **Create View**.

To view the read-only view

- Select the **Incident Information** view from the **View** drop-down list in the Support Incidents list screen. This view will be a read-only list which cannot be edited or deleted.

Arrange Grid Columns

This section provides detailed procedures for arranging grid columns.

The appearance of the grid columns in the list view can be arranged according to your preference. You can specify the names of the grid columns and the order of their appearance in the list view.

Scenario

Consider the case of the Support Incident List View. The user wants the columns in the list view to be displayed in a different order. The user can change the position or order of the columns to be rendered in the list grid. This helps the user to organize the information in the view.

The default system fields displayed while creating the view are listed below. These fields can be used for sorting or filtering purposes and also to display it in the grid.

- CreatedBy
- CreatedDateTime
- ModifiedBy
- Owner
- WorkflowStatus

To arrange the grid columns in the views

1. In the **Enterprise Console**, click the **Menu** button, click **Lists**, and then click **Support Incidents**. The **Support Incidents** list items page appears.
2. Go to the **View** drop-down list field and select **Create View**.
3. Type the name **By Email** for this view of the list.
4. In the **Grid Columns** field, select the **Display Column Name** and **Display Order Number** in the required order.
5. Click **Create View**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To view the By Email view

- Select the **By Email** view from the **View** drop-down list in the Support Incidents list screen. You can see all the grid columns according to your specifications.

Copy from Existing Views

This section provides detailed procedures for copying from existing views.

While creating a new view which is similar to an existing view, but with few differences, you can copy the template of an existing view and continue creating a new view.

Scenario

Consider the case of the Support Incident View. The user has created some of the public views to get easy access to all the support incident list items. However, the user now wants to create a view to get only the Query incidents which have originated through Email. To achieve this, the user can create a new view while selecting the existing view (By Email), that was created previously. This feature can help the user to create a new view by using the template of an existing view.

To copy from pre-existing views

1. In the **Enterprise Console**, click the **Menu** button, click **Lists**, and then click **Support Incidents**. The **Support Incidents** list items page appears.
2. Go to the **View** drop-down list field and select **Create View**.
3. Type the name **Email Query** for this view of the list.
4. Select the **By Email** template from the pre-existing views in the **Copy From** drop-down list.

   You can see that the display column selections made in the By Email template are displayed here because of selecting the By Email in the Copy From field.
5. Click the **Show items only when the following is true** radio button in the **Filter Columns** field. You can see the By email filter conditions appearing in the field.
6. Add another filter condition by selecting **Incident Type**, **is equal to**, and **Value**.
7. Type **Query** in the condition field.
8. Click **Create View**.

To view the Email Query view

- Select the Email Query view from the **View** drop-down list in the Support Incidents list screen. All the list items which have originated by email and are of the query incident type are listed.

Manage Views

This section provides detailed procedures for managing and securing the views created using the List Views feature. From this page, you can create, edit/update, or delete a view. You can even modify the security settings for a view.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Update a View

This section provides detailed procedures for updating a list view.

Scenario

The Telematics representative wants to update the **Email Query** view by excluding some of the grid columns in the view. To update a view the user has to select the Update View option in the Update View screen.

To update a view

1. In the **Enterprise Console**, click menu icon, click **Lists**, and then click **Support Incidents**. The **Support Incidents** list items page appears.
2. Go to the **View** drop-down list field and select **Email Query** view.
3. Click **Go**. The **Update View** screen appears.
4. Uncheck the **Number**, **Support Person**, and **Created By** check boxes and reassign the column order numbers.
5. Click the **Update View** button at the end of the screen.

   Note: While updating a list view, the user may come across certain exceptions owing to reasons like assigning inappropriate values, repeating the same column in the sort expression, database connectivity errors etc. An exception message appears in a pop-up window highlighting the possible reasons for the exception.

To view the updated Email Query screen

- Select the **Email Query View** from the View drop-down list in the Support Incidents screen.

Delete a View

This section provides detailed procedures for deleting a list view.

Scenario

The Telematics representative wants to delete the Incident Information Read-Only view. To delete a view the user has to select the Delete View option in the Update View screen.

To delete a view

1. In the **Enterprise Console**, click the **Menu** button, click **Lists**, and then click **Support Incidents**. The Support Incidents list items page appears.
2. Go to the **View** drop-down list field and select **Incident Information** view.
3. Click **Go**. The **Update View** screen appears.
4. Click the **Delete View** button at the end of the screen.

To view the result of deleting the view

- On clicking the View drop-down list you will not be able to find the Incident Information view option in the list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Secure a View

AVEVA Work Tasks security feature allows the administrator to assign rights for groups and map users to the groups. This option helps you to define security for a view.

The **List View Securi**ty consists of **Standard Settings** and **Advanced Settings**.

- **Standard Settings** is a reporting function that allows you to view the Security for User/Role or the Security Mapping at List Level.
- **Advanced Settings** allows you to modify the existing mapping, add a new resource mapping or delete the existing mapping by enabling mapping at List level.

There are two ways to enable security at list level:

- Enabling Resource Mapping
- Customizing the Security Group Definitions

To access Standard Settings

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage Views**. The **Manage Views** page appears.
2. Click the **Security** icon on the **Ribbon Bar**. The **Security Settings** page appears.

Field	Description
View Effective Security For User/Role	Click this link to select a role/user and view the effective security group and rights for the selected role/user.
View Security Mapping at List Level	Click this link to view the security mapping at the list level. It provides details of the Security Groups, Roles and Users having access to the List.

To access Advanced Settings

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage Views**. The **Manage Views** page appears.
2. Click the **Security** icon on the **Ribbon Bar**.
3. Select the **Advanced Settings** tab. The **Security Settings** page appears.
4. Click the **Enable List specific Security** link. The **Enable Security wizard** is displayed.

Field	Description
Security Group Customization	Select this option to complete the wizard and display the Security Group screen. Security Group is a list which lets the user to create a new security group, modify, view and delete an

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	existing security group. It also lets the user to view the effective mapping for a particular security group, which displays the information about the security group and its mappings, it shows where the entire security group is mapped to the resource and/or modified.
Resource Mapping	Select this option to complete the wizard and display the Security Group Mapping screen. It allows the user to set the priority, if the user is part of multiple roles and if these roles are mapped to different security groups. While defining the Security Group to Role mapping there will be an option to set a priority number. So in case a user is part of multiple roles and if these roles are mapped to different Security Groups then based on the Lowest priority number given to the roles the security will take effect.

5. Select the required option(s) and click **Next** to complete the wizard.
6. Select the appropriate Security Group and click **Edit** on the Ribbon bar. If you select the Administrator group security and click **Edit**, the **Modifying an Existing Security Group** dialog box appears.
7. Select the appropriate options and click **Save**.

Applying security for a list view

- AVEVA Work Tasks allows you to apply the security for a specific list view. Then this security will not affect any other views in the Manage View page.

To apply the security for specific view

1. On the **Manage View**, select a View and right click.
2. Select **Security Settings** from the right click options. The **Security Settings** page appears.
3. Select the **Advanced Settings** tab.

Field	Description
Security Group- Resource Mapping for this List Item	Click this link to add, modify and delete Resource Mapping at List Item level.
Customize Security Group at List Item level	Click this link to customize Security Group at List Item level
Remove Security from this ListItem	Click this link to remove security for deleting the Resource Mapping /Security Group Customization from the intermediate above level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click the required link. If you select **Customize Security Group for this List Item**, the **Security Group** page appears.
5. Select appropriate **Security Group** and click **Edit** on the Ribbon bar. If you select the Administrator group security and click Edit the Modifying an Existing Security Group dialog box appears.

   Note: You can restrict the display of specific view in the List View drop-down by de-selecting the **Can Consume Item** right of the security group. To display the specific view in the List View drop-down, select the **Can Consume Item** right of the security group.
6. Select the appropriate options and click **Save**.

Note: If you apply the security at View level, it will not affect any other View present in the Manage View page. All the security settings that you have been configured specifically for a view will be applicable only for that particular view.

List Lookups

Lookup is a powerful feature in AVEVA Work Tasks that allows you to fetch data from a List or a SQL Database. Lookup Settings is a separate module that allows you to create lookups and maintain it as a Lookup list.

The Data source for a lookup can be a Database or List. Database is used to populate lookup data from a database and List is used to populate data from an existing List. When you create a List or Form, the Lookup Input control can be easily linked to the desired lookups available in the Lookup list. Consider a scenario of creating a new user for the "User list". Details of the user like name, password, email, department and manager fields are required. In this scenario, Managers can be the existing users from the "Manager list" for which the lookup can be created.

Supported Data Types

Supported Data Types

The supported data types for SQL include bigint, binary, bit, char, date, datetime, decimal, float, int, money, nchar, numeric, nvarchar, real, smalldatetime, smallint, smallmoney, tinyint, uniqueidentifier, varbinary, varchar, and xml. This is applicable for database queries, stored procedures, and list lookups.

BigInt, Decimal, Float, and Money data types in the List control is limited to the range -999999999999999 to 999999999999999. This limited range is due to the limitation of a number in the underlying JavaScript technology used in Forms.

SmallMoney data types in the List control is limited to the range -214748.3648 to 214748.3647. Also, the Number control in the associated form generated by the List is limited to 2 decimal places. So, the Number control range would be from -214748.36 to 214748.36.

The supported data types for Oracle include bfile, binary_double, binary_float, blob, char, clob, date, float, interval day, interval year, long, long raw, nchar, nclob, number, nvarchar2, raw, rowid, urowid, varchar, and varchar2. This is applicable for database queries and stored procedures.

Note:
- Each user in the "Manager list" will have a unique id but can have similar names.
- Lookup control has properties **Value column** and **Display column**.
- The value for the **Value** column will be persisted in the form instance XML of the "User list". However, the value for the "display column" is retrieved from the "Manager list" based on the value of the **Value** column when the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

form is rendered. Hence its mandatory that the **Value** column selected for the lookup control has unique values.
- For the **Display** column, it is recommended to use the **String** data type. In the case of Database Lookup, the Display column should be of a data type that can be implicitly converted to a string during execution. For other types of lookups, consider using general data types such as **String**, **Number**, or **DateTime**. Avoid using data types like **Binary**, **Large Text Columns**, or **Arrays**. In summary, a **String** data type is the preferred choice for the Display column.

Start Lookup Settings

This section explains how to start the Lookup Settings.

To start Lookup Settings

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Lookup Settings**. The Lookup Settings page appears.
2. The **Lookup Settings** page shows the title, description, lookup type, owner, item status, work in progress, workflow status, and last modified date for the lookup lists.

Lookup Settings Interface

A proper understanding of the Lookup Settings Interface allows you to effectively use the various features of the application.

The **Lookup List** is the default screen that is displayed when you start **Lookup Settings**. It allows you to create new database lists, lookup lists, and organize them.

Lookup Settings Interface

Lookup Settings interface has a breadcrumbs, an action bar, and a list of lookups.

Breadcrumbs

Breadcrumbs display the path to the child lookup or folder, and has commands to customize and view the list of lookups.

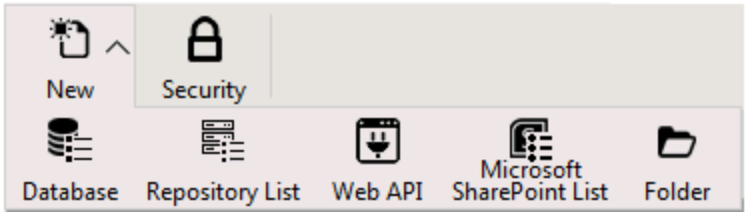
Command	Description
All Items	Default view to display the list of lookups.
Create View	Create a new view to display the list of lookups.

Action Bar

Action bar has commands to manage lookups and folders. Standard commands are available all the time whereas contextual commands are available only when a lookup is selected.

Standard Commands

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



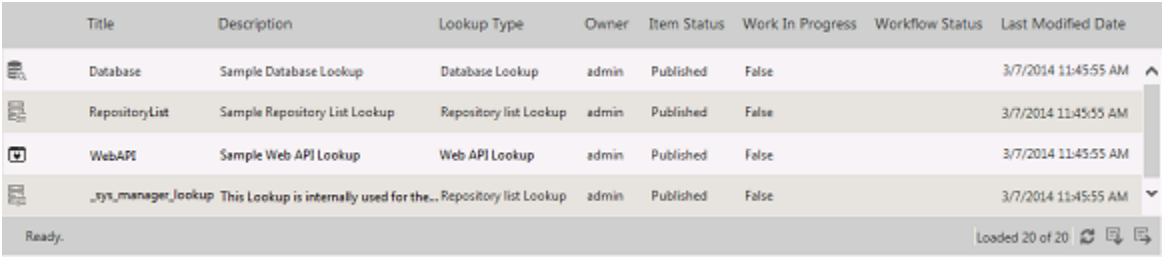
Menu/Command	Command	Description
New	Database	Create a new database lookup.
	Repository List	Create a new repository list lookup.
	Web API	Create a new web API lookup.
	Folder	Create a new folder to group and organize lookups.
Security		View or customize the security level for the lookups.

Contextual Commands

Command	Description
Edit	Edit the lookup.
Delete	Delete the lookup.

List

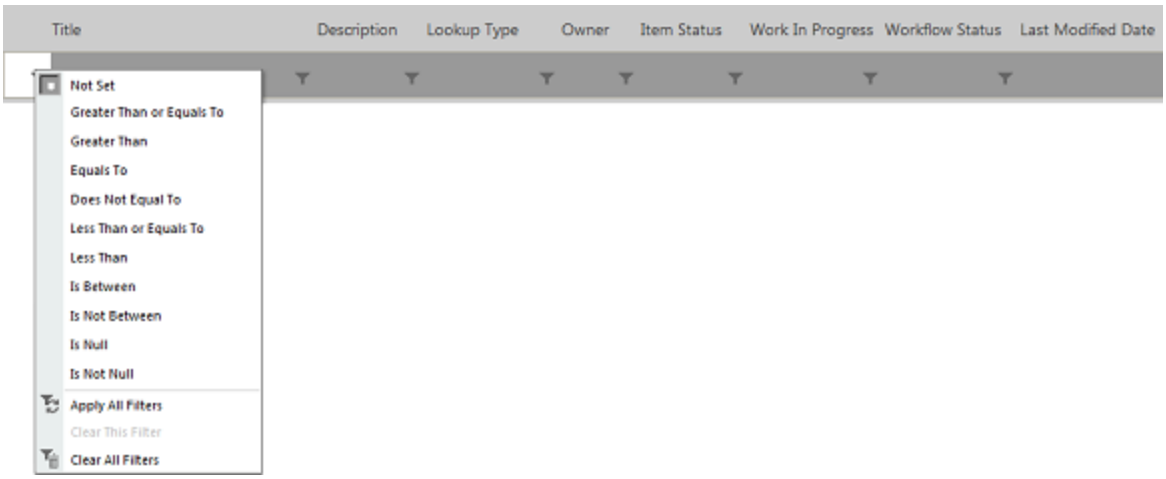
List of lookups shows the title, description, lookup type, owner, item status, work in progress, workflow status, and last modified date for the lookups. Click on a column to sort the lookups.



Filter Bar

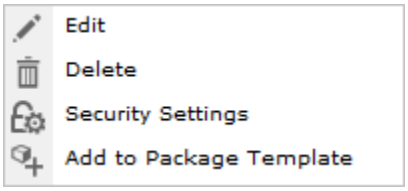
Filter bar can help to quickly find the desired lookup from the list of lookups. Filter bar has a funnel button and a parameter area for every column of the list of lookups. The filter button shows the available filter options and the parameter area is for entering the filter criteria. The filter bar is hidden by default. Right-click on the column header, and then click **Filter Bar**, to access the filter bar.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



List Shortcut Menu

List shortcut menu has commands to manage the lookup. Right-click on a lookup to access the shortcut menu.



Command	Description
Edit	Edit the lookup.
Delete	Delete the lookup.
Security Settings	View or customize the security level for the lookups.
Add to Package Template	Add the lookup to a package template.

Lookup Settings

This section provides information on the following:

- Add a New Repository List Lookup
- Add a New Database Lookup
- Add a New Web API Lookup
- Edit a List Lookup
- Organize Lookup Lists
- Delete a List Lookup

Add a New Repository List Lookup

This section provides detailed procedures for creating a new Repository List lookup.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To add a new Repository List lookup

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Lookup Settings**. The **Lookup Settings** page appears.
2. Click the **New Lookup List** icon on the **Ribbon Bar**. The **New Repository List Lookup** dialog box appears.

Field	Description
Lookup Name	Enter the appropriate title for the lookup.
Description	Enter the appropriate description for the lookup.
List Name	Select the List name to be associated with the lookup.
Get all records	Select the required option. • Select **True**, if you would like to display all the records from the list. • Select **False**, if you would like to display only selected records from the list. When you select **False**, the List Query field is displayed below this field.
List Query	Click the **Query** button to build a filter query to get the required records. This field is enabled only if the **Get all records** option is selected as "False". Query can be built using Value, Parameter or Table Fields. For more information on building a List Query, see Add a List Query.
Order By Query	Click the **Query** button to build a query to sort the list items based on the given condition (Similar to Order by clause of the sql queries). Data can be either in ascending or descending order. For more information on building a Order By Query, see Add an Order By Query.
Table	Click this text box to select the required table. The tables that are associated to a List are displayed in a tree menu as shown below. The tree menu displays the hierarchy of tables.
Available Columns	This field displays all the columns associated to the selected table.
Selected Columns	This field displays all the selected columns of the selected table. You can select or deselect the columns.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Display Column	This field displays the column to be displayed in the lookup. For example, Employee Name column.
Value	Select the required option. The options are Auto, Custom. • If **Auto** option is selected, the Id column of the table selected is considered as value field. For Example, for an Employee list, the Employee Name will be display column and Employee ID will be Value column. • If **Custom** option is selected, you can specify the desired value column.. **Note:** The column specified in the **Value** field should identify a unique row in the table.
Display value column in grid	Select the required option. The options are Yes, No. • If **Yes** option is selected, the selected value column is displayed in the lookup. • If **No** option is selected, the selected value column is not displayed in the lookup.
Form	This field does not support any functionality for the current release.

Note:
- The column "Title" will always be a part of the Available Columns and Display Column fields. The "Title" column should not be selected if it is not a part of the List Definition.
- The **Title** of repository list lookups cannot have special characters and blank space. The only allowed special character is _ (underscore). Also, the Title starting with _SYS_ is not allowed.

3. Type the lookup name in the **Lookup Name** field and enter its description in the **Description** field.
4. Select the List name, and then select the required columns.
5. Click **Update**.

Please note the following important things about the **Dynamic Query Builder**.

The following operators are available for creating queries using the Dynamic Query Builder:

- =
- >

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- <
- >=
- <=
- <>
- IN
- Not In
- Like
- Is Null
- Is Not Null

Queries cannot have any Special Characters except '@' for Parameter Name.

Add a List Query

Using List Query, you can built a filter query for the lookup to fetch records. This field is enabled only if the **Get all records** option is selected as **False** in the **New Repository List Lookup** dialog box.

To build a query

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Lookup Settings**. The **Lookup Settings** page appears.
2. Click the **New Lookup List** icon on the **Ribbon Bar**. The **New Repository List Lookup** dialog box appears.
3. Click the **Query** icon against **List Query** to build a filter query to get the required records.

   The **Dynamic Query Builder** dialog box opens.

Field	Description
Condition (*Prefix*)	Grouping prefix for the condition.
Table	Click this text box to select the required table. The tables that are associated to a List are displayed in a tree menu. The tree menu displays the hierarchy of tables.
Column	Select the column associated to the selected table.
Operator	Select the available operator for creating Query.
Type	• Select the Type to display the filtered records. The available Type options are Value, Table Field, and Parameter.
Value	• When the Type is **Value**, the Value can be given as a string with Single quotes for a string or a number for Number type as per the Column

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	selected. • When Type is **Table Field**, in the Value, the table and the column can be specified. • When Type is **Parameter**, the Value can be given as @<parameter name>.
Condition (*Suffix*)	Grouping suffix for the condition with the option of specifying AND and OR operators to build multiple conditions based query.

4. Click **Add** to add the query.

   Multiple conditions can be specified using the Condition (*Prefix*) and Condition (*Suffix*) options.
5. Click **Save** to save the query.
6. Click **Clear** to remove the data entered in the Query Builder.

Add an Order By Query

Using Order By Query, you can build a query to sort the list items based on the given condition (similar to Order by clause of the sql queries). Data can be either in ascending or descending order.

To build an Order Byquery

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Lookup Settings**. The **Lookup Settings** page appears.
2. Click the **New Lookup List** icon on the **Ribbon Bar**. The **New Repository List Lookup** dialog box appears.
3. Click the **Query** icon against **Order By Query** to build a query to sort the list items.

   The **Dynamic Order By Query Builder** dialog box opens.

Field	Description
Table	Click this text box to select the required table. The tables that are associated to a List are displayed in a tree menu. The tree menu displays the hierarchy of tables.
Column	Select the column associated to the selected table.
Sort	Select **ASC** to list the items in ascending order or **DESC** to list in the descending order.

4. Click **Add** to add the query.

   Multiple columns can be specified for the Order By query.
5. Click **Save** to save the query.
6. Click **Clear** to remove the data entered in the Query Builder.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Add a New Database Lookup

Add a **Database Lookup** to use the result set in the lookup.

To add a Database Lookup

1. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Lookup Settings**. The **Lookup Settings** page appears.
2. Click **New**, and then click **Database**. The **New Database Lookup** dialog box appears.
3. In the **Title** box, type a name for the database lookup, and in the **Description** box, type a description for the database lookup.
4. Do one of the following:

Task	Action
• Design the database lookup using a query.	a. In the **Command Type** list, click **Query**, and then click **Continue**. The **New Database Lookup** dialog box appears with additional options. b. Follow the steps for Add a New Database Lookup Using Query.
• Design the database lookup using a stored procedure.	a. In the **Command** Type list, click **Stored Procedure**, and then click **Continue**. The **New Database Lookup** dialog box appears with additional options. b. Follow the steps for Add a New Database Lookup Using Stored Procedure.

Note: Lookup title cannot have special characters and blank space. The only allowed special character is _ (underscore). Also, the Title starting with _SYS_ is not allowed.

Default Search Behaviour of Database Lookups

The default search behaviour of database lookups is to include the **Display Value** column and the **Value** column. If there is only a single **Value** column, then that **Value** column is considered, otherwise the first column returned from the results of the select query is considered provided that column is part of the **Value** column.

For example, If the **Display Value** is **EmpDetails**, the **Value** column contains multiple columns – **Emp.Address**, **Emp.Name**, and **Emp.Age**, and the select query is `SELECT * from Emp;` then **EmpDetails** from the **Display Value** column and **Emp.Name** from the **Value** column is considered for the search as the **Emp.Name** column is the first column in the database table.

Add a New Database Lookup Using Query

Add a **Database Lookup** using query to use the result set in the lookup.

- Follow the steps for Add a New Database Lookup till you a get the **New Database Lookup** dialog box.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To add a Database Lookup using a Query

1. Select the Database connection. Enter the required query. Data will be retrieved based on the query from the database.

   Note:
   - You can write queries to fetch data from tables, synonyms and views.
   - The use of aliases, aggregate functions, joins in database lookup query is supported through the use of views.
   - Click here for a scenario on handling aliases, aggregate functions, joins in the database lookup query.
   - The Database Lookup only supports the dbo schema by default. However, you can create views to support any other custom
   schemas.
   -The database lookup query supports the use of CASE expression. Click here for a scenario on using CASE expression in the
   query.
   - Square brackets should not be used in the SQL query for Oracle database. In case the Square brackets are used as part of a string, then the user must use **CONCAT** or **||** and **CHR** functions instead of Square brackets. For example, if the string used in the query is 'Home [test] Page', then it can be given in either of the following two ways:
   -- CONCAT('Home ', CONCAT(CHR(91), CONCAT('test', CONCAT(CHR(93), ' Page'))))
   -- 'Home ' || CHR(91) || 'test' || CHR(93) || ' Page'

2. Click **Load Parameters** to load parameters from the query, and then select a data type for each listed parameter.

   Note:
   - For SQL database, the supported data types include Big Integer, Boolean, Byte, Char, Date, DateTime, Decimal, Double, Guid, Integer, Single, Small Integer, and String.
   - For Oracle database, the supported data types include Big Integer, Byte, Char, Date, DateTime (in correct format), Decimal, Double, Integer, Single, Small Integer, and String. For Guid data, use nvarchar2 instead of raw.

3. Enter the column name to be used as **Value** field in the lookup input control. The value field column(s) must be one of the selected columns in the Query field. Example: Employee.EmployeeId denotes Employee table and EmployeeId column.

   Ensure that the case of the table name used in the **Value** field matches with the case of the table name in the query. For example, if the table name given in the query is **EmployeeDepartment**, then an entry **employeeDepartment** in the Value field, is considered invalid.

   Note:
   - If required, you can enter multiple column names separated by commas.
   - Please ensure that the column names are always preceded by table names.
   - Also, the column name(s) should identify a unique row in the table.
   For example, TableName.Id,TableName.title,TableName.description

4. Enter the column name to be used as display field in the lookup input control. Example: FirstName.
5. Click **Validate & Preview**. It validates all the entries made for the creating the database lookup. If all the entries are valid, a preview of the data fetched by the database lookup is displayed.

   Also, if the database lookup query is with parameters, you can preview the data after providing valid sample inputs. Click here for a database lookup scenario based on query with parameters.

   Note:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Following are some of the important checks to be done to ensure that the entries for the database lookup are valid:
- The **Title** of database lookups cannot have the following special characters: `~! @#$%^&*()+|`-=\\{}:\"<>?[];',./ - (Including blank space)`. The only allowed special character is _ (underscore).
- The **Title** of database lookups cannot begin with a number. For example, 12EmployeeDetailsDBLookup is an invalid entry.
- The query entered in the **Query** field must be syntactically correct.
- Also, the query validation will fail when the query is executed on a invalid database connection or the table name or column names used in the query are invalid. For example, if the query entered is **"Select * Employee Where Id=@id"**, on click of **Validate & Preview**, an error message is displayed as the query is syntactically incorrect.
- All the columns entered in the **Value** field must belong to the same table. For example, **Employee.EmployeeId**,**Department.DeptId** is an invalid entry.
- The columns entered in **Value** field must not repeat. For example, **Employee.EmployeeId**,**Employee.EmployeeId** is an invalid entry.
- The columns entered in the **Value** and **Display Name** fields must be a part of the query. For example, the query is **"Select Id, Dept, Designation From Employee"**. **Value** field entered is **Employee.DeptId** and **Display Name** is **EmpName**. This is an invalid scenario as the columns entered for the **Value** and **Display Name** fields are not a part of the query.
- The column specified in the **Value** field should identify a unique row in the table.

6. Click **Close** in the **Preview** screen. Click **Save** in the database lookup screen to save the changes to the database lookup.

Please note the following important points about Database Lookup:

- If Lookup fails to resolve the Display Column value based on the Lookup value, then the Lookup value will be shown as the display text for that particular Lookup.
- For lookups created on case sensitive databases, ensure that the queries entered are also case sensitive.
- Do not use an image field (type: Byte Array) as the value field for the Database lookup
- Display Name of database lookups which have been created on external (i.e. non repository) databases, cannot be resolved when viewed from the Lists grid.
- The parameters used in the database lookup query should not have any special characters, should not be blank and should be in `columnname=@parametername` format.
- Also, note the following important points while creating Database lookups to be associated with the Grid Configuration:
- If any of the columns selected in the database lookup used in a Grid Configuration are image fields of type Byte Array, the Data Grid on rendering does not show the images. The image columns will be shown as System Byte. Also, these columns cannot be used in workflows.
- If the column names in the selected data source lookups have any special characters other than _, the user will not be allowed to select them in the Persist Columns field. Also, if the column names have space or begin with numbers the user will not be allowed to select them in the Persist Columns field.
- If the value field in the lookup which is consumed in a Grid Configuration has a special character, space or begins with a number, the Grid Configuration will be considered invalid.
- Filter on Display Value for Data Lookup control is through SQL Like operator. Therefore, we recommend setting the Display Name column, which is of string data type. For more information, refer to Supported Data Types.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Add a New Database Lookup Using Stored Procedure

Add a **Database Lookup** using stored procedure to use the result set in the lookup.

- Follow the steps for Add a New Database Lookup till you a get the **New Database Lookup** dialog box.

To add a Database Lookup using a Stored Procedure

1. In the **Database Connection** list, click the name of the database connection.
2. Click **Configure** for the **Procedure Details**. The **Procedure Details** dialog box appears.
3. In the **Procedure Name** box, type the name of the stored procedure, and then click **Load Details**.
4. If there are parameters for the stored procedure, then the parameters appear. In such cases, do the following:
   a. By default, all parameters are selected. Click to clear parameters that are not required to be passed when the stored procedure is called, if required.
   b. Type values for the parameters, and then click **Load Columns**. The **Value Column** and the **Display Name Column** are populated.
5. In the **Value Column** list, click the column names to be used as the value fields.
6. In the **Display Name Column** list, click a column name to be used as the display field.
7. Click **Save**. The procedure details configuration is saved and the **Procedure Details** dialog box is closed.
8. Do one of the following:

Task	Action
• Show the value field in the grid.	• Click **Yes** for the **Show Value Column**.
• Show the display field in the grid.	• Click **No** for the **Show Value Column**.

9. Click **Save**. The database lookup is created and the **New Database Lookup** dialog box is closed.

Note:
- The **Title** of database lookups cannot have special characters and blank space. The only allowed special character is _ (underscore). Also, the Title starting with _SYS_ is not allowed.
- Stored procedures from Oracle databases are not supported, and therefore any database connection to Oracle database will not be available in the **Database Connection** list.
- Only the user defined stored procedures that return only the result set are supported.
- Only the first result set is supported for stored procedures that return multiple result sets.
- Only the columns from the result set, not the out parameters, are supported for the value and display name columns.
- Multiple value columns are currently not supported.
- The supported data types for the value column include bit, datetime, decimal, double, guid, int32, int64, nvarchar, single, uint16, uint32, uint64, and varchar.
- The supported data types for the parameters include bigint, bit, char, date, datetime, datetimeoffset, decimal, float, int, money, nchar, nvarchar(max), numeric, nvarchar, real, rowversion, smalldatetime, smallint, smallmoney, varchar(max),
time, tinyint, uniqueidentifier, and varchar.
- Only the mandatory parameters, that is the parameters that are selected, are made available in **Data Lookup** and **Data Grid** controls.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The column specified in the **Value** field should identify a unique row in the table.
- While passing the parameters for lookups and data grid controls, ensure that the value for the **Boolean** data type is either **True** or **False**.
- While passing the parameters for lookups and data grid controls, ensure that the value for the **Date Time** control is converted from the UTC format to the local time format. For more information about the Date Time control, see Date Time.

Adding parameters for stored procedures

For string parameters, empty string value is passed, and not null values.

To check whether the string parameter value passed to the stored procedure is empty, add the following condition in the **Where** clause:

```
NULLIF(@EmpName,'') is null
```

Where, **@EmpName** is a string parameter in the stored procedure and **NULLIF** is an SQL function.

You can also check whether the string parameter is empty by adding the following condition in the **Where** clause:

```
@EmpName=‘’
```

Also while defining the parameters, ensure that the parameters are given default values as shown below:

```
CREATE PROCEDURE [dbo].EmpDetailsProc
-- Add the parameters for the stored procedure here
@EmpID decimal(18, 0) = null,
@EmpName nvarchar(MAX) = null,
@IsMarried bit = null,
@EmpDOJ datetime = null,
@Guid uniqueidentifier = null,
@Salary float = null
:
:
```

Add a New Web API Lookup

Web APIs are frameworks to build HTTP services for browsers and mobile devices. Use Web APIs to customize your business applications.

To add a Web API Lookup

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Lookup Settings**. The **Lookup Settings** page appears.
2. Click **New**, and then click **Web API**. The **New Web API Lookup** page appears.
3. In the **Name** box, type a name for the API.
4. In the **Description** box, type a description for the API.
5. In the **Web API** list, select a Web API connection.

   Based on the Web API connection selected, enter the URI in the **Request URI** box.

   For example, **api/product/1** for **http://localhost/MySite/api/product/1** (for Request URI without parameters) or **api/{product}/{ID}** for **http://localhost/MySite/api** (for Request URI with parameters).

   - For the Web API not configured with open API specifications, the **Load Columns** option is available to get

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the columns data. When you click **Load Columns** button, the Web API will be executed in the Web API server. Based on the Web API functionality, there is a possibility of data update or delete. For more information, see Configure Web API

- For the Web API configured with open API specifications, specify the **Operation Group** and **Operation** for the selected Web API to get the columns data. For more information on configuring open Web API connection, see Invoke Web API: Configure OpenAPI.

6. Select column names for the **Selected Columns**. The selected columns are displayed in the lookup. Ensure that the value column contains unique data.
7. Select the data type for the selected column name for the **Selected Columns**, if required.

Note: The data type mentioned for the selected columns is used while fetching the data. Make sure that the data type specified for the selected column matches with the source data.

8. In the **Display Column** drop-down list, select the column name.
9. In the **Value Column** drop-down list, select the column name.
10. In the **Encoding** drop-down list, select the encoding type. The supported encoding types are **Default**, **UTF-7**, **UTF-8**, and **UTF-16**. The **Default** encoding type is used for backward compatibility.
11. Select **Yes** for **Advanced Settings**, if you want to add the Header value or display the value of request URI.

    The Header value is displayed if the Web API has the Header value. You can also add a new Header value.

    Note:
    - In the case of Web API, you cannot set the **Headers** as optional i.e. the **Optional** property of the Header cannot be set to **YES**. For each Header, the Key and Value are mandatory properties.
    - In the case of OpenAPI, you cannot set the **Optional** property of the **Header**. You cannot provide the Key, but you need to provide the Value. If the **Optional** property is **YES** and the Value is empty, the value will not be sent to the Web API and will be considered null. If the **Optional** property is **NO**, you need to provide the Value.
12. In the **JSON Path for Child Item** box, provide the JSON path to retrieve child nodes in a JSON having multiple nodes.
13. Select **Yes** for **Display Value Column in Grid**, if you want to display the value column.
14. In the **Display Value Request URI** box, type a URI. For example, **api/DataTable/{Id} for http://localhost/MySite/api/DataTable/**.
15. Click **Save**.

    The Web API lookup is saved and the **Web API** dialog box is closed.

    Note:
    - The **Title** of Web API lookups cannot have special characters and blank space. The only allowed special character is _ (underscore). Also, the Title starting with _SYS_ is not allowed.

    - The Web API for the **Display Value Request URI** helps in resolving the display value directly from the specific row of data. This helps to maintain consistency in the data displayed, save the network bandwidth, and improve the speed of execution.

    - The value of the **Owner** for the Web API Lookup in lists is not resolved by default as it involves connecting to a Web API and thereby consuming network bandwidth and reducing the speed of execution. A workaround is to add an additional field in the list and a corresponding hidden field in the form to map the value with the display value, and show only the display value field in the list grid.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Web API Lookup settings can be deleted even if they are being used.

- The column specified in the **Value** field should identify a unique row in the table.

- The Web API response should be in the JSON format.

- The columns containing JSON objects should be mapped only to string data types. Further, these columns should not be configured for **Display Column** or **Value**.

Edit a List Lookup

This section provides detailed procedures for editing the Lookup.

To edit a lookup

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Lookup Settings**. The **Lookup Settings** page appears.
2. Select the appropriate lookup and click **Edit**
   OR
   Right click the appropriate lookup and select **Edit**. The **Edit Repository List Lookup** dialog box appears.
3. **Edit** the appropriate details and click **Update**.

Organize Lookup Lists

Organize a lookup list by performing one of the following:

- Moving the lookup list to a folder.
- Making the lookup list a child item of another lookup list.

To organize a lookup list by moving the lookup list to a folder

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Lookup Settings**. The **Lookup List** page appears.
2. On the action bar, click **Organize**. The **Organize Lookup List** dialog box appears.
3. Drag a lookup list to a folder. The lookup list is moved to the folder.
4. Click **Close**. The **Organize Lookup List** dialog box is closed.

To organize a lookup list by making the lookup list a child item of another lookup list

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Lookup Settings**. The **Lookup List** page appears.
2. On the action bar, click **Organize**. The **Organize Lookup List** dialog box appears.
3. Drag a lookup list to another lookup list. The lookup list is now a child item of another lookup list.
4. Click **Close**. The **Organize Lookup List** dialog box is closed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Delete a List Lookup

This section provides detailed procedures for deleting a Lookup.

To delete a lookup

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Lookup Settings**. The **Lookup Settings** page appears.
2. Select the appropriate lookup and click **Delete**. A warning message is displayed.
3. Click **Yes**.

List Lookup Scenarios

This section explains some of the scenarios on using the List Lookup features:

- Create a Lookup for a Database Table
- Create a Lookup for a Database Table with Composite Primary Key
- Create Dependent Lookup from Database Tables
- Create a Lookup for a List
- Search or Sort within a Lookup Data
- Add an Item from Lookup Control
- Use ExtendedInformation Property for List Lookups
- Use ExtendedInformation Property for Database Lookups
- Use Aliases, Joins, Aggregate Functions in Database Lookup Query
- Use CASE Expression in the Database Lookup Query

Create a Lookup for a Database Table

AVEVA Work Tasks allows you to create a database lookup and list lookup to retrieve data from a database. This allows the user to select from a set of options, for example list of states, countries etc. To create a database lookup, you need to connect to a database.

Before starting with the scenario, create a database connection.

To connect to a database

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Database Connections**. The **Database Connections** page appears.
2. On the **Ribbon bar**, click **New**. The **New Connection** dialog box appears.
3. In the **Title** box, type the appropriate name for the connection. In this case, enter **formslistscenario**.
4. In the **Description** box, type any description. This is not mandatory.
5. Select the **Connect to repository datasource** check box to connect to the current repository database. A database connection for the repository datasource is successfully created and is displayed in **Connection List** page.

   Or

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Clear the **Connect to repository datasource** check box to connect to an external database. Perform the steps mentioned in the following tables.

Field	Description
Database Type	Select appropriate database type from the options. Available options are SQL Server and Oracle. In this case, select **SQL Server**.
Connection Type	Select the appropriate connection type from the options. Available options are Connection String and Connection Provider. Select **Connection String**.
Server Name	Click the **Refresh** button and then select the appropriate server that you are using, from the drop-down list. In this case, you will select **Server1\SQLEXPRESS**.
Authentication Type	Select the appropriate authentication type from the drop-down list. Since, you are using windows for authentication, select **Windows** from the drop-down list.
User Id	Enter your user Id, if you have selected any other authentication type than Windows.
Password	Enter your password, if you have selected any other authentication type than Windows.
Advanced Settings	Enter any advanced settings, if required.
Database	Click the **Refresh** button and then select the appropriate database that you are using, from the drop-down list. In this case, you will select **FormsScenarioDB**.

6. Enter the details as mentioned in the field description table.
7. Click **Test Connection**.

   A message appears indicating that the connection is successful.
8. Click **Close**.
9. Click **Save**.

   The database connection is listed the **Connection List** page.

Now, you have created a database connection. Next, you will create a database lookup for Countries. The Country lookup will retrieve the data from the database table and display the list of countries along with the other details like country code.

To create the lookup

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Lookup Settings**. The **Lookup Settings** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. On the **Ribbon bar**, click **New**, and then click **Database**. The **New Database Lookup** window appears.
3. In the **Title** box, type a name for the database lookup.
4. In the **Description** box, type a description for the database lookup.
5. In the **Command Type** drop-down list, select **Query**.
6. Click **Continue**.
7. Enter the details as mentioned in the field description table.

Field	Description
Database Connection	Select an appropriate database connection from the drop-down list. You have created the database connection **formslistscenario**, so select that database connection.
Query	Enter the query to execute and display the lookup input control. Data will be retrieved based on the query from the database. Example: **SELECT * from Country**.
Value	Enter the column name to be used as value field in the lookup input control. Example: **Country.CountryId**. If required, you can enter multiple column names separated by commas. Please ensure that the column names are always preceded by table names. For example, TableName.Id, TableName.title, TableName.description. Also, the column name(s) should identify a unique row in the table.
Display Name	Enter the column name to be used as display field in the lookup input control. Example: **CountryName**.
Display value column in grid	Show or hide the value column in the grid. • Select **True** to show the value column in the grid. • Select **False** to hide the value column in the grid. The default value is **True**.

8. Click **Save**. The details are saved and the lookup is listed in the **Lookup List** page.

Note: You have created the **Country** lookup. This lookup can be used to retrieve data in the form.

To design a form

1. Design a new form. For more information about designing a form, see Designing Forms.
2. Drag and drop the Data Lookup control in the form.
3. Edit the properties for the Data Lookup control.
4. Click the **Lookup Source** icon. The **Lookup Source Configuration** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Select the **Database Lookup** in the **Data Source Type**.
6. Select **Country** in the **Data Source**.
7. Click **Save & Continue**.
8. Click **Apply** in the **Property** window.
9. Click **Preview**.
10. Enter the first three characters for the Country in the **Data Lookup** control. The countries whose title contains the specified characters are displayed in the drop-down list.
11. Select the appropriate country from the drop-down list.
12. Click **Finish**.

Create a Lookup for a Database Table with Composite Primary Key

The Database lookup supports creating lookups for tables which have more than one column defined as the primary key.

Consider a **Job-Steps** scenario from a typical manufacturing process. A **work order** is a collection of jobs that produce an item. A **job** is a list of steps or procedures that are executed to produce an item or version of item.

Let us say, the work order details are stored in the **Work_Order** table with **wo_id** as the work order id. Jobs involved for producing an item are stored in the **Job** table with **oper_id** as the job id. The various steps that make up one specific job are stored in the **Job_Steps** table. To uniquely identify any row in the **Job_Steps** table we need **wo_id**, **oper_id**, **seq_no** and **step_no** columns.

In this scenario, we will create a lookup for **Jobs** and a lookup for **Job Steps**. The user selects a Job from the Job Lookup. Job Step lookup gives the list of steps involved in completion of the selected Job.

Before Starting with the scenario, create a database connection. The table schemas for the tables involved are as follows:

Job Table Schema

Column Names
wo_id
oper_id
seq_no
job_desc
state_cd
job_priority

Job_Step Table Schema

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Column Names
wo_id
oper_id
seq_no
step_no
step_seq
step_name
step_desc
action_type

To connect to a database

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Database Connections**. The **Database Connections** page appears.
2. On the **Ribbon bar**, click **New**. The **New Connection** dialog box appears.
3. In the **Title** box, type the appropriate name for the connection. In this case, enter **MESWorkOrderConnection**.
4. In the **Description** box, type any description. This is not mandatory.
5. Select the **Connect to repository datasource** check box to connect to the current repository database. A database connection for the repository datasource is successfully created and is displayed in **Connection List** page.

   Or

   Clear the **Connect to repository datasource** check box to connect to an external database. Perform the steps mentioned in the following tables.

Field	Description
Database Type	Select appropriate database type from the options. Available options are SQL Server and Oracle. In this case, select **SQL Server**.
Connection Type	Select the appropriate connection type from the options. Available options are Connection String and Connection Provider. Select **Connection String**.
Server Name	Click the **Refresh** button icon and then select the appropriate server that you are using, from the drop-down list. In this case, you will select **Server1\SQL2008STANDARD**.
Authentication Type	Select the appropriate authentication type from the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	drop-down list. Since, you are using SQL Server for authentication, select **SQL Server** from the drop-down list.
User Id	Enter your user Id, if you have selected any other authentication type than Windows.
Password	Enter your password, if you have selected any other authentication type than Windows.
Advanced Settings	Enter any advanced settings, if required.
Database	Click the **Refresh** button and then select the appropriate database that you are using, from the drop-down list.

6. Enter the details as mentioned in the field description table.
7. Click **Test Connection**. A message appears indicating that the connection is successful.
8. Click **Close**.
9. Click **Save**. The database connection is listed in the **Connection List** page.

   Now, you have created a database connection.

   Next, you have to create a database lookup for Job and Job Steps.

To create the lookup

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Lookup Settings**. The **Lookup Settings** page appears.
2. On the **Ribbon** bar, click **New**, and then click **Database**. The **New Database Lookup** window appears.
3. In the **Title** box, type a name for the database lookup. In this case, enter **JobLookup**.
4. In the **Description** box, type a description for the database lookup.
5. In the **Command Type drop**-down list, select **Query**.
6. Click **Continue**.
7. Enter the details as mentioned in the field description table.

Field	Description
Database Connection	Select an appropriate database connection from the drop-down list. You have created the database connection **MESWorkOrderConnection**, so select that database connection.
Query	Enter the query to execute and display the lookup input control. Data will be retrieved based on the query from the database. Example: **SELECT * from**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	job
Value	Enter the column names to be used as value field in the lookup input control. Example: **job.wo_id, job.oper_id, job.seq_no** **Note:** The column specified in the **Value** field should identify a unique row in the table.
Display Name	Enter the column name to be used as display field in the lookup input control. Example: **job_desc**.

8. Click **Validate & Preview**. If the validation is successful and the preview has been generated, click **Save**. The details are saved and the Lookup List page appears.

Note: You have created the JobLookup. This lookup can be used to retrieve data in the form.

Next, you need to add another lookup for Job_Steps table.

9. Follow the steps above for creating a Database lookup. Enter the details for creating the lookup.

Field	Description
Title	Enter the appropriate title for the lookup. In this case, enter **JobStepLookup**.
Description	Enter any description. This is not mandatory.
Database Connection	Select an appropriate database connection from the drop-down list. You have created the database connection **MESWorkOrderConnection**, so select that database connection.
Query	Enter the query to execute and display the lookup input control. Data will be retrieved based on the query from the database. Example: **SELECT * from job_step where wo_id=@woid and oper_id=@operid and seq_no=@seq_no** **Note:** The parameters used in the database lookup query should not have any special characters, should not be blank and should be in columnname=@parametername format.
Value	Enter the column names to be used as value field in the lookup input control. **Example:Job_Step.Wo_Id, Job_Step.Oper_Id, Job_Step.Seq_No, Job_Step.Step_No** **Note:** The column specified in the **Value** field should identify a unique row in the table.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Display Name	Enter the column name to be used as display field in the lookup input control. Example: step_desc.

10. Click **Validate & Preview**. The **Preview** dialog box for **Resolve Parameter** appears. Enter valid sample inputs for the parameters and then click Proceed.

The **Preview** screen appears with the preview of the data fetched by the database lookup.

11. Click **Close**.
12. Click **Save**. The details are saved and the lookups are listed in the **Lookup List** page.
13. Design a new form.
14. Drag and drop the Data Lookup control in to the form.

1. Edit the properties for the Data Lookup control.
2. Click the **Lookup Source** icon. The **Lookup Source Configuration** page appears.
3. Select the **Database Lookup** in the **Data Source Type**.
4. Select **JobLookup** in the **Data Source**.
5. Click **Save & Continue**.
6. Click **Apply** in the **Property** window.
7. Drag and drop another Data Lookup control in to the form.
8. Edit properties for the Data Lookup control.
9. Click the **Lookup Source** icon. The **Lookup Source Configuration** page appears.
10. Select the **Database Lookup** in the **Data Source Type**.
11. Select **JobStepLookup** in the **Data Source**.

    Note: Since, you have set the 'where' condition for the **JobStepLookup**, the **Parameters** field is displayed in the Lookup.

12. Click **Save & Continue**.
13. Click **Apply** in the **Property** window.
14. Click in the text box next to the **Controls** drop-down box.
15. Select **wo_id** from the tree structure. Also set the other two parameters using values from the SKRootDefinition of JobsLookup.
16. Click **Submit**.
17. In the **Lookup Input Property** window, click **Save**. The JobStepLookup control is added to the form.
18. Click **Preview** to preview the form.
19. Click the **Search** button corresponding to the Jobs Lookup field. Select any job by double-click or by the **Select** button.
20. Click the **Search** button corresponding to the Job Steps Lookup field. It displays the list of job steps involved in the selected job.
21. Select a Job Step.

    Note: Multiple values set in the **JobStepsLookup** allow you to select a Job Step.

22. Click **Finish**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create Dependent Lookup from Database Tables

AVEVA Work Tasks allows its users to create dependent lookups, i.e. the second lookup will display the values based on the value selected from the first lookup. This functionality is very important in forms and list.

In this scenario, you will learn how to create a dependent lookup with the help of two tables; Country and State. The Country table consists of two columns, **CountryID** and **CountryName**. The State table consists of three columns, **StateID**, **StateName**, and **CountryID**.

We will then create two lookups, Country Lookup and State Lookup. The Country lookup is similar to the one we have created in the scenario, Create a Lookup for a Database Table.

In this scenario, we will use the 'where' condition in the query builder while creating the State lookup.

To create a dependent lookup from database tables

1. Create a lookup with the name **State** with the details as mentioned in the field description table.

Field	Description
Database Connection	Select appropriate database connection from the drop-down list. For more details on how to create the database connection, refer to the scenario, Create a Lookup for a Database Table.
Query	Enter the query to execute and display the lookup input control. Data will be retrieved based on the query from the database. Here, you need to specify the 'Where' condition also. Then the value will be retrieved based on the where condition. Example: **SELECT * from State where CountryId=@CountryId**
Value	Enter the column name to be used as value field in the lookup input control. Example: **State.StateId** **Note:** - If required, you can enter multiple column names separated by commas. - Ensure that the column names are always preceded by table names. For example, **TableName.Id,TableName.title,TableName.description**. - Also, the column name(s) should identify a unique row in the table.
Display Name	Enter the column name to be used as display field in the lookup input control. Example: **StateName**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Create another lookup with the name **Country** with the details as mentioned in the field description table.

Field	Description
Database Connection	Select appropriate database connection from the drop-down list. For more details on how to create the database connection, refer to the scenario, Create a Lookup for a Database Table.
Query	Enter the query to execute and display the lookup input control. Data will be retrieved based on the query from the database. Example: **SELECT * from Country**
Value	Enter the column name to be used as value field in the lookup input control. Example: **Country.CountryId** **Note:** - If required, you can enter multiple column names separated by commas. - Ensure that the column names are always preceded by table names. For example, **TableName.Id,TableName.title,TableName.description**. - Also, the column name(s) should identify a unique row in the table.
Display Name	Enter the column name to be used as display field in the lookup input control. Example: **CountryName**

To design a form

1. Design a new form. For more information about designing a form, see Designing Forms.
2. Drag and drop the Data Lookup control in to the form.
3. Edit the properties for the Data Lookup control.
4. Click the **Lookup Source** icon in the **Data Source** field. The **Lookup Source Configuration** page appears.
5. Select the **Database Lookup** in the **Data Source Type**.
6. Select **Country** in the **Data Source**.
7. Click **Save & Continue**.
8. Click **Apply** in the **Property** window.
9. Drag and drop the Data Lookup control in the form.
10. Edit the properties for the Data Lookup control.
11. Click the **Lookup Source** icon. The **Lookup Source Configuration** page appears.
12. Select the **Database Lookup** in the **Data Source Type**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

13. Select **State** in **Data Source**.

    The **Parameters** field is displayed.

    Note: In the case of parameterized lookups, when the parent lookup selection is changed the value selected in the child lookup is not getting cleared off.
    In this scenario, consider that the user selects a Country and then selects a State for that Country. When the user changes the Country selection, the State lookup value remains the same. The value selected in the State lookup is not being cleared off.

14. For the parameter **@CountryId**, select the first Data Lookup control from the drop-down list.
15. Click **Save & Continue**.
16. Click **Apply** in the **Property** window.
17. Click **Preview**.
18. Enter the first three characters for the Country in the Data Lookup control. In the drop-down list, the countries with the title containing the three characters are displayed.
19. Select the appropriate country from the drop-down list.
20. Enter the first three characters for the State in the Data Lookup control. In the drop-down list, the states with their title containing the three characters and belonging to the selected country in the Country Data Lookup control, are displayed.
21. Select the appropriate state from the drop-down list.

Create a Lookup for a List

AVEVA Work Tasks has the provision to create a Database lookup and List lookup to retrieve data. In the Database lookup the data is retrieved from database tables whereas in a List lookup, data is retrieved from List.

In this scenario, you will create a lookup for the **Employee Details** list. Consider a form which is used for filling in an employee's annual rating with two fields, **Employee** and **Rating**. Employee field will be a drop-down (lookup) only for the active Employees. In this scenario we will try to understand, how to build an active employee lookup.

Before we start with the scenario, create an **Employee Details** list with three fields Title, Employee Number and Active.

To create a lookup for a list

1. Create an **Employee Details** list.
2. Run the script mentioned below in the repository database.

```
CREATE TABLE [dbo].[EmployeeDetails](
[Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF__EmployeeDetails__Id__1E6F845E] DEFAULT (newid()),
[Title] [nvarchar](250) NOT NULL,
[EmployeeNumber] [nvarchar](150) NULL,
[Active] [bit] NULL,
[Application] [nvarchar](150) NULL,
[CreatedBy] [uniqueidentifier] NULL,
[CreatedDateTime] [datetime] NULL,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
[LastUpdatedDateTime] [datetime] NOT NULL,
[WIP] [bit] NULL,
[WIPItemId] [uniqueidentifier] NULL,
[MainItemId] [uniqueidentifier] NULL,
[ParentItemId] [uniqueidentifier] NULL,
[RecordParentItemId] [uniqueidentifier] NULL,
[TableId] [uniqueidentifier] NULL,
[ItemRowIndex] [int] NULL,
[IsLatest] [bit] NULL,
[Version] [nvarchar](25) NULL,
[VersionHistory] [nvarchar](max) NULL,
[Status] [int] NULL,
[LockedBy] [uniqueidentifier] NULL,
[LockedOn] [datetime] NULL,
[ListID] [uniqueidentifier] NULL,
[SecuritySettings] [nvarchar](max) NULL,
[ItemId] [uniqueidentifier] NULL,
[ItemType] [int] NULL,
[SecurityItemId] [uniqueidentifier] NULL,
[Owner] [uniqueidentifier] NULL,
[ModifiedBy] [uniqueidentifier] NULL,
[AuditTrail] [nvarchar](max) NULL CONSTRAINT [DF_EmployeeDetails_AuditTrail] DEFAULT
('),
[IsDisabled] [bit] NULL CONSTRAINT [DF_EmployeeDetails_IsDisabled] DEFAULT ((0)),
[WorkflowStatus] [varchar](50) NULL,
[SecurityCustomizationId] [uniqueidentifier] NULL,
CONSTRAINT [PK_EmployeeDetails] PRIMARY KEY NONCLUSTERED
(
[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

Note: Add two additional columns to the standard column as follows:

```
[EmployeeNumber] [nvarchar] (150) NULL,
[Active] [bit] NULL,
```

3. Make an entry in the SKListTables as mentioned below.

```
Insert into SKListTables (Application,TableName)
Values ('<Repository Name>','EmployeeDetails')
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Go

Note: Replace <Repository Name> with the repository you are using.

4. Create a new List named, **Employee Details** for the above table.
5. The **Title** control is of type TextInput to capture the employee name.
6. The **Employee Number** control is of type TextInput to capture employee number.
7. The **Active** control is of type BooleanInput to identify if the employee is active or not.

   Note: For simplicity in this scenario we will use the form auto generated by the list.
8. Input the employee records as follows.

   - John (Active Employee)

   - Tim (Active Employee)

   - Mike (In Active Employee)

   Note: Our next step would be to create a lookup, which will later be used in a Form.
9. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Lookup Settings**. The **Lookup Settings** page appears.
10. Click **New Lookup List**. The **New List Lookup** window appears.
11. Type the **Lookup Name** as "Employees".
12. Type the description to identify the lookup.
13. Select the **Employee Details** list from the **List Name** drop-down list.
14. Select the "False" option for the **Get all records** option.

    Note: Selecting "False" will enable the List Query. In this scenario we have selected "False", since we require active users only. If you select "True" all the employees will be selected.
15. Click **More** button corresponding to the **List Query** field. The **Dynamic Query Builder** dialog box appears.

    Note: Our aim is to build a query to select all employees with column Active with value as "1".
16. Click the **Table** text box. The tables are displayed in a tree menu.
17. Select "EmployeeDetails".
18. Select "Active" from the **Column** drop-down list.

    Note: Active is the column used in the WHERE condition.
19. Select "=" from the **Operator** drop-down list.
20. Select "Value" from the **Type** drop-down list.
21. Type "1" in the parameter field next to the **Type** drop-down list.

    Note: This is used to specify the value when it is populated from another field in the form.
22. Click **Add**. The **Dynamic Query Builder** dialog box appears.
23. Click **Save**.
24. Select the **No** option in the **Select distinct records** field.

    Note: Next we navigate to the Display Columns section. It is used to select the columns to be displayed in the lookup window.
25. Click the **Table** text box.
26. Select "EmployeeDetails" from the tree menu. The columns in the **EmployeeDetails** list is displayed in the **Available Columns** field.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

27. Select Title, EmployeeNumber, Active and click the **Add** button. These fields are selected in the **Selected Columns** field.
28. Select "EmployeeDetails.Title" in the **Display Column** drop-down list.

    Note: It is very important to understand the purpose of the fields, Display Column and Value.

    When the user selects an employee from the Lookup, the Employee name is displayed in the field, but the value that gets stored in the database would be the Employee Number.

    - If you select the **Auto** option in the Value field, the ID column is stored in the database.
    - If you select the **Custom** option in the Value field, you can select the column to be stored in the database as the value.
29. Select the "Custom" option in the **Value** field. A text box and a drop-down list will appear adjacent to the "Custom" option.
30. Click on the Text box and select "EmployeeDetails" from the tree menu.
31. Select "EmployeeDetails.EmployeeNumber" from the drop-down list.
32. Select "Yes" for **Display value column in grid** option.
33. Click **Save**. The **Employee** lookup is created and listed in the Lookup List.

    Note: This lookup can now be used in multiple forms. The following section explains, how the lookup will be used in a Form.

34. In the **Enterprise Console Menu**, click menu icon, and then click **Forms**. The **Forms** page appears.
35. Create a new form named "Employee Rating".
36. Drag and drop the **Lookup Input** control in the **Form Canvas**.
37. Edit the control properties, type the name as "Employee".
38. Click the **Data Source** button corresponding to the **Data Source** field. The **Look up list Explorer** dialog box appears.
39. Click the **Lookup List** text box.
40. Expand the tree menu and select "Employees".
41. Click **Submit**. The "Employee" lookup is selected in the **Look up list Explorer** dialog box appears.
42. Click **Save**. The **Forms Designer** window will be displayed with the "Employee" Lookup Input control.
43. Click the **Preview** button. The **Employee Rating** form is appears in the preview mode.
44. Click the **Search** button. A popup window of all active employees are displayed.
45. Select the employee "John" from the list and click **Select**. The name is populated in the form.
46. Click **Finish**. You can view the generated XML.

The following observations were made in a List lookup scenario where the List lookup is created on a List with child table as shown below.

1. Add countries and states to the list. Also, add a country without adding the respective state.
2. Create a list lookup for the above list with Display Column as "State" and Value Column as "StateId".
3. Consume this lookup in the form's lookup control.

In the lookup which has been rendered as Popup or drop-down, if the user selects a country without a State (child record), the pop up window is not getting closed(popup) or on submit the lookup value is being cleared off(drop-down).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Search or Sort within a Lookup Data

AVEVA Work Tasks has the provision to search or sort within a lookup data. Lookup controls provide the essential function of helping you maintain data integrity in your database environment.

Assume that, you have created a lookup called **PurchaseOrder**. See Create a Lookup for a Database Table and Create a Lookup for a List to know how to create a lookup.

In this scenario we will learn how to sort the lookup data in the ascending order of **ReferenceNo**. In the second part, we will see how to search for a purchase order for an item with ItemName that contains the term **DEL**. This can be achieved by setting the property accordingly while designing the form.

Note: To search or sort within a lookup data, you have to set the following property in the Lookup Input Property window.Select **Yes** for the **Display Filter** option or select **No** for the **Show as Drop down** option.

Having set the property to search or sort the lookup data while designing the form, now let us examine how it works:

To sort the lookup data in the ascending order of ReferenceNo

1. Preview your form.
2. Click the the **Search** button icon near the **PurchaseOrder ReferenceNo** field. All the items in the Purchase Orders list are displayed in the grid form.
3. Click the **ReferenceNo** column heading to sort the lookup data in the ascending order of ReferenceNo.

To search for a purchase order for an item with ItemName that contains the term DEL

1. Click the **Filter** button on the **Item Name** column header to see all the filter options that are available for that column's cells displayed in the filter list.
2. Click **Contains**.
3. Click the text box and type the parameter as **DEL**.
4. Again click the **Filter** button on the **Item Name** column header and select the **Apply All Filters** option. The results of the filtering are displayed.

Add an Item from Lookup Control

AVEVA Work Tasks has the provision to add a new item for the source list from Lookup control. To achieve this, the **Show add new item option for lookup** property of the **LookupInput** control should be set to **Yes**.

The user can add new items for the source list means the list for which the lookup is created using **Add new item** button.

Note: The user who fills the form must have "Can Create Item" right for that list (Source list). Apart from the preview form, in all other forms this option is available if this property is configured as "Yes". By default, this property will be set to "No". This option will be available only if the source list is a dynamic list.

The above feature is explained through a **Vendors list**(Source list) and **Purchase Receiving list**(Destination list) scenario form. The 'Vendors' list has all the vendors and created a lookup for the vendor list. This lookup is used in 'Purchase Receiving' list to get the vendor information. Suppose if the vendor is not there in vendors list, then you can add a new vendor to the 'Vendors' list which will be used in the 'Purchase Receiving' list using this feature.

Before starting with the scenario, create **Vendor list** and **Purchase Receiving list** and map custom tables

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

of **Vendor list** and **Purchase Receiving list**.

For 'Vendors' Lookup Input Control , set **Show add new item option for lookup** property to **Yes**.

To add a new item from Lookup:

1. Select **Purchase Receiving** list from the Enterprise Console. The 'Purchase Receiving' page is displayed.
2. Click on the **New** icon. The **Purchase Receiving** form is displayed.
3. Click on **Vendor** lookup.
4. The **Vendor** lookup dialog box appears. Search for a required user, if not found, add a new vendor using **Add new item**.
5. Click **Add new item** to add a new vendor in the 'Vendors' list. The 'Vendors' list is displayed. Add the required information.
6. Click **Submit**. The newly added vendor is displayed in the 'Purchase Receiving' list.
7. Click on **Vendor** lookup control. The newly added item appears

Use ExtendedInformation Property for List Lookups

This scenario demonstrates the use of "ExtendedInformation" property for List Lookups to populate the employee details.

Scenario

In this scenario, the administrator wishes to see the employee details such as Employee ID, Email, Phone, Mobile, Department and Gender on selection of the employee's name.

Steps to achieve the scenario

1. Create a List lookup **EmployeeListLookup** through **Lookup Settings**. For more details on creating List Lookups, click here.
2. Create a **Employee Form** through **Enterprise Console -> Forms** which consumes the **EmployeeListLookup**.

   The highlighted **Select Employee** control is a Lookup Input control through which the **EmployeeListLookup** is being consumed by the form.
3. Set the **On Data Change** property under the **Advanced** tab for the **Select Employee** Lookup Input control. Add the following script to this property.

```
var extendedInformation = control.extendedInformation;
if(extendedInformation !== null || extendedInformation !== undefined)
{
control.findById("T1").value = extendedInformation["EmployeeId"];
control.findById("T2").value = extendedInformation["LastName"] + " " +
extendedInformation["FirstName"];
control.findById("T3").value = extendedInformation["Phone"];
control.findById("T4").value = extendedInformation["Mobile"];
control.findById("T5").value = extendedInformation["Email"];
control.findById("T6").value = extendedInformation["Department"];
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
control.findById("T7").value = extendedInformation["Gender"];
}
```

4. Click **Save** to save the changes to the properties. Preview the form.
5. Select the required employee from the list of employees.

Use ExtendedInformation Property for Database Lookups

This scenario demonstrates the use of "**ExtendedInformation**" property for Database Lookups to populate the employee details.

Scenario

In this scenario, the administrator wishes to see the employee details such as **Employee ID**, **Email**, **Phone**, **Mobile**, **Department** and **Gender** on selection of the employee's name. Following is a snapshot of the employee details of the organization.

	EmployeeId	FirstName	LastName	Email	Phone	Mobile	DepartmentName	Gender
▶	1	Guy	Gilbert	guy1@ad.com	123456	1234567890	Production	M
	2	Kevin	Brown	kevin@ad.com	234567	2345678901	Marketing	M
	3	Roberto	Tamburello	rob1@a.com	345678	3456789012	Engineering	M
	4	Jolynn	Dobney	jou@ad.com	456789	4567890123	Production	F
	5	Ruth	Ellerbrock	ruth@ad.com	567890	5678901234	Production	F
	6	Gail	Erickson	gail@ad.com	678901	6789012345	Engineering	F

Steps to achieve the scenario

1. Create a Database lookup **EmployeeDBLookup** through **Lookup Settings**.
2. Create a Form **Employee Form** through **Forms** page which consumes the **EmployeeListLookup**.

   The **Select Employee** control is a Lookup Input control through which the **EmployeeDBLookup** is being consumed by the form.
3. Set the **On Data Changed Script** property under the **Advanced** tab for the **Select Employee** Lookup Input control. Add the following script to this property.

```
var extendedInformation = control.extendedInformation;
if(extendedInformation !== null || extendedInformation !== undefined)
{
control.findById("T1").value = extendedInformation["EmployeeId"];
control.findById("T2").value = extendedInformation["LastName"] + " " +
extendedInformation["FirstName"];
control.findById("T3").value = extendedInformation["Phone"];
control.findById("T4").value = extendedInformation["Mobile"];
control.findById("T5").value = extendedInformation["Email"];
control.findById("T6").value = extendedInformation["Department"];
control.findById("T7").value = extendedInformation["Gender"];
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

}

4. Click Save to save the changes to the properties. Preview the form.
5. Select the required employee from the list of employees.

Use Aliases, Joins, Aggregate Functions in Database Lookup Query

The database lookup query supports the use of aliases, joins and aggregate functions etc. through the use of views. Following is a scenario to demonstrate the use of views to handle aliases, joins and aggregate functions.

Scenario

The administrator wishes to see the list of all the departments in the organization and also the number of employees in each department. Following is a snapshot of the Employee details and Department details of the organization.

Employee

	EmployeeKey	FirstName	LastName	MiddleName	DepartmentName
1	43	Peng	Wu	J	Quality Assurance
2	79	Sean	Alexander	P	Quality Assurance
3	81	Diane	Margheim	L	Research and Development
4	114	Mark	Harrington	L	Quality Assurance
5	117	Gigi	Matthew	N	Research and Development
6	149	Andreas	Berglund	T	Quality Assurance
7	162	Dylan	Miller	A	Research and Development
8	204	Hazem	Abolrous	E	Quality Assurance
9	221	Michael	Raheem	NULL	Research and Development
10	257	Sootha	Charncherngkha	T	Quality Assurance

Department

	DepartmentGroupKey	DepartmentGroupName
1	1	Corporate
2	2	Executive General and Administration
3	3	Inventory Management
4	4	Manufacturing
5	5	Quality Assurance
6	6	Research and Development
7	7	Sales and Marketing

Achieving this scenario requires the use of joins, aliases and aggregate functions in the query as shown below:

```
SELECT DepartmentGroupKey AS DepartmentId,
DepartmentGroupName AS DepartmentName, COUNT(EmployeeKey) AS [Number of Employees]
FROM DimDepartmentGroup INNER JOIN DimEmployee
ON DepartmentName=DepartmentGroupName
GROUP BY DepartmentGroupKey,DepartmentGroupName;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to achieve the scenario

Joins and aliases are supported in the database lookup query through the use of views. Follow the steps give below to achieve the above scenario:

1. Create a view in the database, say **DepartmentEmpCount** as shown below:

```
CREATE VIEW DepartmentEmpCount
AS
SELECT DepartmentGroupKey AS DepartmentId,
DepartmentGroupName AS DepartmentName, COUNT(EmployeeKey) AS [NumberOfEmployees]
FROM DimDepartmentGroup INNER JOIN DimEmployee
ON DepartmentName=DepartmentGroupName
GROUP BY DepartmentGroupKey,DepartmentGroupName;
```

2. Create a database lookup, **DeptEmployeeCount** as shown below. Enter the query in the Query field as "**Select * From DepartmentEmpCount**", where DepartmentEmpCount is the view name.
3. Click **Validate & Preview**. The count of employees for each department is displayed.

Use CASE Expression in the Database Lookup Query

The database lookup query supports the use of CASE expression. The following scenario demonstrates the use of CASE expression in the database lookup query.

Scenario

Consider a work order - job scenario where the supervisor wants to identify all the "**Blending**" and "**Pre-mixing**" jobs with the job quantity rejected greater than 5.

JOB table schema

Column Names
wo_id
oper_id
seq_no
job_desc
state_cd
job_priority
item_id
qty_rejected

Follow the steps given below to achieve this scenario.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to achieve the scenario

1. Create a database lookup, say JobQtyRejectedLookup.

   A CASE expression is added to the query to achieve this scenario. The query is as follows:

   ```
   Select wo_id,oper_id,seq_no,job_desc,item_id,qty_rejected,
   Case
   When job_desc='Blending' and qty_rejected>5 then 1
   When job_desc='Pre-mixing' and qty_rejected>5 then 2
   Else 3
   End as Job_Qty_Rejected
   from job
   ```

   Note: Ensure that the columns being used in the CASE expression are a part of the SELECT clause. For example, in the above query job_desc and qty_rejected are a part of the SELECT clause. The column Job_Qty_Rejected cannot be used as a part of display column or a value column.

2. Click **Validate & Preview**. The column Job_Qty_Rejected is displayed which helps the user to identify the **Blending and Pre-mixing jobs with quantity rejected greater than 5**.

List Scenarios

This section explains some of the scenarios on using the List features:

- Create a Leave Application List
- Create a List Mapped to Multiple Tables
- Modify the Default List Form
- Sort List Data
- Export List Item Data to an Excel Sheet
- Enable Versioning for List Item
- Sort List items using DisplaySortOrder Property
- Set Custom View for a List Control from a Custom ASPX Page using API

Create a Leave Application List

AVEVA Work Tasks allows you to create a list and store the data inside the list. You have the provision to map your list to a single table or multiple tables and also to map to an existing table or a generic table which is provided by AVEVA Work Tasks.

In this scenario, you will create a typical Leave Application list and map to a generic table. Our sample leave application list will have the following fields.

- Employee Name
- Employee No: A unique number that identifies the Employee
- Absence Type: The type of absence (whether Sick Leave, Casual Leave, or Privileged Leave)
- Starting date of leave

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Ending date of leave
- And whether the leave is for a complete day or half a day

You will design a leave application list with the above mentioned fields. This can be created and designed from the Manage List page.

To create the Application List

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. On the **Ribbon Bar**, click **New**. The **Create List** window is displayed.
3. Enter the **Title** as **Leave Application**.
4. Enter the description to identify your list.
5. Select **Yes** for **Is Folder Supported?** option to make your list folder supported. Then, you can organize your list in folders.

   Note: If the **Title** field is removed from the base List, Folder is not supported for the list.
6. Click **Save** and Continue. The **Select Table** window is displayed.
7. In this case, you will map your list to the Generic table which is provided by AVEVA Work Tasks. So select Generic Table option, which is selected by default.
8. Click **Next**. The **List Designer** page is displayed.
9. The **Title** field is displayed automatically. The first field in your sample list is **Name of Employee**. So drag and drop a **TextInput** control in to the orange bar in the list name area, from the **Tool Box**. The property window of the **TextInput Control Property** window is displayed.

Field	Description
Tag name	Enter a tag name as **Name of Employee**.
Description	Enter the description for the control. This is not mandatory.
Mapped XML Node	Click anywhere after entering the tag name. The Mapped XML Node field is filled automatically.
Bound to DB Column	This field specifies to which database column you want to map this field.
Mandatory	Select **Yes** radio button to make this field mandatory.
Allow Sorting	Selecting **Yes** will enable sorting by this field.

10. Enter the details and then click **Save**.

    Note: Since, you are in a beginning stage; you will not go in to the settings like, Appearance, Validation, Advanced and Scripts.
11. Next, you need to design the **Employee Number** field. Drag and drop **TextInput** control in to the orange bar in the list name area from the **Tool Box**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Tag name	Enter a tag name as **Employee No**.
Description	Enter the description for the control. This is not mandatory.
Mapped XML Node	Click anywhere after entering the tag name. The Mapped XML Node field is filled automatically.
Bound to DB Column	Specifies to which database column you want to map this field.
Mandatory	Select **Yes** radio button to make this field mandatory.
Allow Sorting	Selecting **Yes** will enable sorting by this field.

12. Enter the details as explained in the field description table, then click **Save**.
13. Next field in your leave form is **Absence Type (Leave Type).** You need to include the options like **Casual Leave**, **Sick Leave** and **Privileged Leave**. Here, we do not have a drop-down control. So select any control and name it as Leave Type. Later when designing the form for the list, you can replace this control with a Dropdown control. In this case you will drag and drop a TextInput control. The **TextInput Control property** window is displayed.
14. Enter the **Tag name** as **Leave Type** and enter the description for the control. This is not mandatory.
15. Click anywhere after entering the tag name. The **Mapped XML Node** field is filled automatically.
16. Select **Yes** radio button to make this field mandatory.
17. After entering all the necessary details, click **Save**.
18. An applicant should mention the leave duration in his/her leave application form. For that there should be the leave starting date and ending date fields in the list. So, next, you will create leave starting date field. For that drag and drop **DateInput** in to the orange bar in the list name area from the **Tool Box** options. The **Date Input Properties** window is displayed.

Field	Description
Tag name	Enter a tag name as **Leave Start Date**.
Description	Enter the description for the control. This field is not mandatory.
Mapped XML Node	Click anywhere after entering the tag name. The Mapped XML Node field is filled automatically.
Bound to DB Column	Specify to which database column you want to map this field. This field is not mandatory. Type **LeaveStartDate**.
Mandatory	Select **Yes** radio button to make this field mandatory.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Allow Sorting	Selecting **Yes** will enable sorting by this field.

19. After entering all the required information as mentioned in the field description, click **Save**.
20. Next, you will design an end date for the leave. For that drag and drop the **DateInput** control in to the orange bar in the list name area. The **Date Input Properties** window is displayed.

Field	Description
Tag name	Enter a tag name as **Leave End Date**.
Description	Enter the description for the control. This field is not mandatory.
Mapped XML Node	Click anywhere after entering the tag name. The Mapped XML Node field is filled automatically.
Bound to DB Column	Specify to which database column you want to map this field. This field is not mandatory. Type **LeaveEndDate**.
Mandatory	Select **Yes** radio button to make this field mandatory.
Allow Sorting	Selecting **Yes** will enable sorting by this field.

21. After entering all the needed information as mentioned in the field description, click **Save**.
22. Next, you need to design a field to mention whether the leave is for a complete day or a half day. For that drag and drop **BooleanInput** control in to the orange bar in the list area. The **Boolean Input Property** window is displayed.

Field	Description
Tag name	Enter a tag name as **Half day?.**
Description	Enter the description for the control. This field is not mandatory.
Mapped XML Node	Click anywhere after entering the tag name. The Mapped XML Node field is filled automatically. You need to delete the '?' mark from the node as XML Node should not have any space or special characters.
Bound to DB Column	Specify to which database column you want to map this field. This field is not mandatory. Type **Halfday**.
Mandatory	Select **Yes** radio button to make this field mandatory.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
Allow Sorting	Selecting **Yes** will enable sorting by this field.

23. Enter the essential data in the respective fields as explained in the field description table. Then click **Save**.
24. Now, you have designed all the fields required for your leave application. Click **Submit** to save the list.
25. A form for your list is auto generated. Click **Yes** if you want to edit the auto generated list form. Clicking **No** will take you to the List Manage page, There you can see the Leave Application list.

    Note: When you submit a list, a form is automatically created for the list. You can customize the control properties or modify the properties in the forms designer page.
26. Next, you will see how to create items for the Leave Application list that you have created now.

To create Leave Application list items

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. When you click **Leave Application**, the corresponding list item window is displayed.
3. Since, this list is just created now, you will not find any items. You will notice that the column headings are same of your Leave Application list. On the Ribbon bar, click **New** to create new item. The **New List Item** window is displayed.
4. Enter the details. Ensure that all the mandatory fields are entered correctly.
5. After entering all the mandatory fields, click **Submit**.

Now, you can see that a Leave Application list item is added. In the same way, you can create any number of list items for any list that you have created.

Create a List Mapped to Multiple Tables

AVEVA Work Tasks allows you to create a list and to map your list to multiple tables. You can select an existing table or a generic table which is provided by AVEVA Work Tasks to map the table.

In this scenario, you will create a **Purchase Order** list and map to multiple tables. You will map to custom tables **Purchase Order** and **Purchase Order Details** which are created in your database server. Our sample **Purchase Orders** list will have the following fields.

- Reference Number
- Purchase Order Date
- Raised By User
- Deliver To User
- Status
- Purchase Order Details
  - Item Name
  - Quantity
  - Unit Price

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Item Total
- Total Amount

Note: Both the parent and child should have the standard columns apart from those listed above.

You will design a **Purchase Orders** list with the above mentioned fields. This can be created and designed from the **Manage List** page.

Before starting with the scenario, you need to create **Purchase Order** and **Purchase Order Details** tables in your database.

To create a list mapped to multiple tables

1. Login to your SQL Server Database.
2. Copy the script mentioned below to your server and execute the script.

```
USE [FormsScenarioDB]
GO
/****** Object: Table [dbo].[PurchaseOrder] Script Date: 12/01/2008 18:50:42 ******/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATE TABLE [dbo].[PurchaseOrder](
[Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF__PurchaseOrder__Id__1E6F845E] DEFAULT
(newid()),
[Title] [nvarchar](250) NOT NULL,
[Application] [nvarchar](150) NULL,
[CreatedBy] [uniqueidentifier] NULL,
[CreatedDateTime] [datetime] NULL,
[LastUpdatedDateTime] [datetime] NOT NULL,
[WIP] [bit] NULL,
[WIPItemId] [uniqueidentifier] NULL,
[MainItemId] [uniqueidentifier] NULL,
[ParentItemId] [uniqueidentifier] NULL,
[RecordParentItemId] [uniqueidentifier] NULL,
[TableId] [uniqueidentifier] NULL,
[ItemRowIndex] [int] NULL,
[IsLatest] [bit] NULL,
[Version] [nvarchar](25) NULL,
[VersionHistory] [nvarchar](max) NULL,
[Status] [int] NULL,
[LockedBy] [uniqueidentifier] NULL,
[LockedOn] [datetime] NULL,
[ListID] [uniqueidentifier] NULL,
[SecuritySettings] [nvarchar](max) NULL,
[ItemId] [uniqueidentifier] NULL,
[ItemType] [int] NULL,
[SecurityItemId] [uniqueidentifier] NULL,
[Owner] [uniqueidentifier] NULL,
[ModifiedBy] [uniqueidentifier] NULL,
[AuditTrail] [nvarchar](max) NULL CONSTRAINT [DF_PurchaseOrder_AuditTrail] DEFAULT
(''),
[IsDisabled] [bit] NULL CONSTRAINT [DF_PurchaseOrder_IsDisabled] DEFAULT ((0)),
[WorkflowStatus] [varchar](50) NULL,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
[SecurityCustomizationId] [uniqueidentifier] NULL,
[ReferenceNo] [nvarchar](50) NULL,
[PurchaseOrderDate] [datetime] NULL,
[RaisedBy] [nvarchar](50) NULL,
[DeliverTo] [nvarchar](50) NULL,
[TotalAmount] decimal(12,2) NULL,
CONSTRAINT [PK_PurchaseOrder] PRIMARY KEY NONCLUSTERED
(
[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
Insert into SKListTables (Application,TableName)
Values ('FormsScenarioRepo','PurchaseOrder')
/***** Note: You need to replace the repository name ‘FormsScenarioRepo’ with your
repository name. Also replace the database ‘FormsScenarioDB’ with your
database name *******/
Go
```

To create purchase order details:

- Copy the script given below to your database server and execute it.

```
USE [FormsScenarioDB]
GO
/****** Object: Table [dbo].[PurchaseOrderDetails] Script Date: 12/01/2008 18:49:10
******/
SETANSI_NULLSON
GO
SETQUOTED_IDENTIFIERON
GO
SETANSI_PADDINGON
GO
CREATE TABLE [dbo].[PurchaseOrderDetails](
[Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF__PurchaseOrderDetails__Id__1E6F845E]
DEFAULT (newid()),
[Title] [nvarchar](250) NOT NULL,
[Application] [nvarchar](150) NULL,
[CreatedBy] [uniqueidentifier] NULL,
[CreatedDateTime] [datetime] NULL,
[LastUpdatedDateTime] [datetime] NOT NULL,
[WIP] [bit] NULL,
[WIPItemId] [uniqueidentifier] NULL,
[MainItemId] [uniqueidentifier] NULL,
[ParentItemId] [uniqueidentifier] NULL,
[RecordParentItemId] [uniqueidentifier] NULL,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
[TableId] [uniqueidentifier] NULL,
[ItemRowIndex] [int] NULL,
[IsLatest] [bit] NULL,
[Version] [nvarchar](25) NULL,
[VersionHistory] [nvarchar](max) NULL,
[Status] [int] NULL,
[LockedBy] [uniqueidentifier] NULL,
[LockedOn] [datetime] NULL,
[ListID] [uniqueidentifier] NULL,
[SecuritySettings] [nvarchar](max) NULL,
[ItemId] [uniqueidentifier] NULL,
[ItemType] [int] NULL,
[SecurityItemId] [uniqueidentifier] NULL,
[Owner] [uniqueidentifier] NULL,
[ModifiedBy] [uniqueidentifier] NULL,
[AuditTrail] [nvarchar](max) NULL CONSTRAINT [DF_PurchaseOrderDetails_AuditTrail]
DEFAULT (''),
[IsDisabled] [bit] NULL CONSTRAINT [DF_PurchaseOrderDetails_IsDisabled] DEFAULT ((0)),
[WorkflowStatus] [varchar](50) NULL,
[SecurityCustomizationId] [uniqueidentifier] NULL,
[ItemName] [nvarchar](50) NULL,
[Quatitiy] decimal(12,2) NULL,
[UnitPrice] decimal(12,2) NULL,
[ItemTotal] decimal(12,2) NULL,
CONSTRAINT [PK_PurchaseOrderDetails] PRIMARY KEY NONCLUSTERED
(
[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
Insert into SKListTables (Application,TableName)
Values ('< FormsScenarioRepo>','PurchaseOrderDetails')
/***** Note: You need to replace the repository name ‘FormsScenarioRepo’ with your
repository name. Also replace the database ‘FormsScenarioDB’ with your
database name *******/
Go
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To create the Purchase Order:

1. On the **Ribbon** bar, click **New**. The **Create List** window is displayed.
2. Enter the **Title** as **Purchase Orders**.
3. Enter any description to identify your list.
4. Select **Yes** for **Is Folder Supported?** option to make your list folder supported. Then you can organize your list in folders.

   Note: If the **Title** field is removed from the base List, Folder is not supported for the list.
5. Click **Save and Continue**. The **Select Table** window is displayed.
6. In this case, you will map your list to the custom table. So select **Existing Table** option. When you select this option, the **Select Table** field is displayed.
7. Click the **Refresh** button. The tables that you have created in your database server are displayed in the **Select Table** field.
8. Select **Purchase Orders** table from the list and then click **Next**. The **List Designer** page appears.

   All the fields that you have created in your database are displayed in the list. Now you want to add one child item i.e. **Purchase Order Details** to the list. Assume that you want to add the grid just before the **Total Purchase Amount** field.
9. Drag and drop **List Table** from the **Tool Box** to the green area of the **Total Amount** field. The **Select Table** window is displayed.
10. Select **Existing Table** option to map your child table to the custom table.
11. Click the **Refresh** button. The tables that you have created in your database server are displayed in the **Select Table** field.
12. Select **Purchase Order Details** table from the list and then click Next. The **ListTableForm** property window is displayed.
13. Enter the **Tag name** as **Purchase Order Details**.
14. Click **Save**.
15. Now, you have designed all the fields required for your **Purchase Order** list. In this case you have two lists -parent list and child list- and both are mapped to different tables. Click **Submit** to save the list.
16. A form for your list is auto generated. Select **Yes**, if you want to edit the auto generated list form. Selecting **No** will take you to the **Manage List** page, There you can see the **Purchase Order** list.

    Note: When you submit a list, a form is automatically created for the list. You can customize the control properties or modify the properties in the forms designer page.

    Through **Forms Designer** the child table can be replaced with **Grid**. Refer to the scenario Modify the Default List Form.

    Next, you will create **Purchase Order** list items.

To create Purchase Order list items

1. In the **Enterprise Console**, click ☰ menu icon, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Click **Purchase Order**. The corresponding list item window is displayed.

   Since, this list is just created now, you will not find any items. You will notice that the column headings are

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

same of your **Purchase Order** list. On the **Ribbon** bar, click **New** to create a new item. The **New List Item** window is displayed.

3. Enter the details. Ensure that all the mandatory fields are entered correctly.
4. After entering all the mandatory fields, click **Submit**.

Now, you can see that a **Purchase Order** list item is added. In the same way, you can create any number of list items for any list that you have created.

Modify the Default List Form

AVEVA Work Tasks allows its end users to modify the default list forms using **Edit** option. When you submit a list, a form is automatically generated for that list and it will be saved in the **Forms** page. You can find the generated form inside the **Dynamic List Forms** folder in the Forms page.

In List Designer, you have very limited controls and that are not sufficient to design a sophisticated form. So you need to design a list with available controls. After creating a list a form is auto-generated for the list. This auto-generated form can be modified later.

In the previous scenario, you have created a Purchase Order list and a form was auto generated for the list. In this scenario, you will modify the Purchase Orders form to make it more functional.

The objective is to replace the control form with the grid and place all the child items in the grid.

To modify the default list form

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The Forms page appears.
2. In the **Dynamic List Forms**, you will see a form with the same name of the list i.e **Purchase Order**. Select that list and on the Ribbon bar click **Edit**. The **Forms Designer** page is displayed.

   Note: We are now replacing the Purchase Order Details form with grid.
3. Click the **Replace Control** button corresponding to the **Purchase Order Details** control. The **Replace Control** window is displayed.
4. Click the drop-down list and select Grid option.
5. Select **Yes** for the field, **Copy similar properties?** This is to maintain the same properties for the control.
6. Click **Replace**. The form control is replaced with the Grid control.
7. Click **Submit**.

   Note: Similarly you can replace List controls with Form controls like Radio Buttons, Lookup Input etc.

   If you have deleted any field by mistake, you can add the deleted field by dragging and dropping the appropriate control. But ensure to use the same XML node bound to the table of the deleted control.

   Assume that you have deleted the field **Purchase Order Date** from your **Purchase Order** list. But after deleting it, you have realized that you require the same field and you want to create the field.

To add the Purchase Order Date

1. Drag and drop the **Date Input** in to the orange bar of the form area. This will add the control at the end of the form. If you want to add the control in the previous location, drop the control in to the green bar of the control which was just after the **Purchase Order Date** in your previous form. The Date Input Property window is displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Enter the **Tag name** as **Purchase Order Date**.
3. Enter the same Mapped XML Node of the field in the form that you have deleted. Your XML node was **PuchaseOrderDate**. So enter this in the Mapped XML Node field.
4. Then click **Save**. The date field is added to the form.

Note: In this scenario, you have added the Date Input. If you wish, you can add DateTime input control also. Ensure that you have entered the same XML node of the deleted field.

Sort List Data

AVEVA Work Tasks allows you to sort the list items in the ascending order (A to Z) or in the descending order (Z-A), Oldest to Newest, Newest to Oldest, Smallest to Largest and Largest to Smallest orders depending on the nature of the field. If you have thousands of list items in your console, you can locate a particular record by sorting the list items.

Before we start with this scenario, create an **Employee Details** list with three list items, Employee ID, Employee Name and Employee Address using TextInput controls.

Our objective is to enable sorting for the Employee Name column.

To sort list data

1. Launch the **Employee Details** list in the List Designer.
2. Click the **Properties** button corresponding to the Employee Name control. The **Properties** dialog box appears.
3. Select **Yes** for the Allow Sorting option and click **Save**. The List Designer page is displayed.
4. Click **Submit**. The **Re-Generate Form** dialog box is displayed.
5. Select **No**, **do not regenerate the default form** and click **OK**. The Manage List page is displayed.
6. On the **Console Menu**, select **Lists** and click **Employee Details**.
7. Right click the **Employee Name** column header.
8. Apply the required sort type. For Example, if you select **Sort AtoZ**, the **Employee Names** will be displayed in the alphabetic order A to Z.

Note: The sorting order will vary depending on the field you have selected. If you select the **Employee ID** field, sorting order will be **Smallest to Largest** and **Largest to Smallest**.

In the above example, we have enabled sorting only for the list item **Employee Name**. When you click the column header of **Employee ID** or **Employee Address**, the sort option is disabled.

Export List Item Data to an Excel Sheet

AVEVA Work Tasks provides an option to export list item data as an Excel Sheet.

Before we start with this scenario, create an Employee Details list with three list items, Employee ID, Employee Name and Employee Address using TextInput controls.

Populate the **Employee Details** list with some valid data.

Our objective is to export the data entered to an MS Excel (.xls) format.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To export list item data to an excel sheet

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **Manage List**. The **Manage List** page appears
2. Click **Employee Details**. The Employee Details list items are displayed.
3. Click the **Export** button on the right bottom corner of the Status bar.
4. Select **Excel** option and click **Export now**. The **File Download** window is displayed.
5. Click **Save**. The **Save As** window is displayed.
6. Select the appropriate location, type the file name and click **Save**.

Enable Versioning for List Item

AVEVA Work Tasks allows its users to create different versions for a list item which is mapped to multiple tables (parent and child). The information in the parent table is considered as an identifier and is not versioned. All child table information is versioned.

In case of Employee Details, the parent table contains the parent items, Employee Name and Employee No. This list is not version enabled until a child list is added.

Versioning can be enabled for a child table with a single row of data and for a child table with multiple rows of data. In this scenario, first you will see how a child table with one record can be version enabled and the second scenario deals with a child table with multiple records.

Assume that you are creating an **Employee Details** list with the following table schema.

Table **Employee Details** with fields

- EmployeeName
- EmployeeNo

Table **Address** with fields

- CurrentAddress
- PermanentAddress

Also assume that, to map the list you have created two tables Employee Details and Address in your database.

If you have not created the tables in your database, use the script given below to create the Employee Details table.

To create an Employee Details table

1. Login to your **SQL Server**.
2. Copy the script given below to your server and execute.

```
USE [formsandlist]
GO
/****** Object: Table [dbo].[EmployeeDetails] Script Date: 12/15/2008 15:47:44 ******/
SET ANSI_NULLS ON
GO
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[EmployeeDetails](
 [Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF__EmployeeDetails__Id__1E6F845E] DEFAULT (newid()),
 [Title] [nvarchar](250) NOT NULL,
 [Application] [nvarchar](150) NULL,
 [CreatedBy] [uniqueidentifier] NULL,
 [CreatedDateTime] [datetime] NULL,
 [LastUpdatedDateTime] [datetime] NOT NULL,
 [WIP] [bit] NULL,
 [WIPItemId] [uniqueidentifier] NULL,
 [MainItemId] [uniqueidentifier] NULL,
 [ParentItemId] [uniqueidentifier] NULL,
 [RecordParentItemId] [uniqueidentifier] NULL,
 [TableId] [uniqueidentifier] NULL,
 [ItemRowIndex] [int] NULL,
 [IsLatest] [bit] NULL,
 [Version] [nvarchar](25) NULL,
 [VersionHistory] [nvarchar](max) NULL,
 [Status] [int] NULL,
 [LockedBy] [uniqueidentifier] NULL,
 [LockedOn] [datetime] NULL,
 [ListID] [uniqueidentifier] NULL,
 [SecuritySettings] [nvarchar](max) NULL,
 [ItemId] [uniqueidentifier] NULL,
 [ItemType] [int] NULL,
 [SecurityItemId] [uniqueidentifier] NULL,
 [Owner] [uniqueidentifier] NULL,
 [ModifiedBy] [uniqueidentifier] NULL,
 [AuditTrail] [nvarchar](max) NULL CONSTRAINT [DF_EmployeeDetails_AuditTrail] DEFAULT (''),
 [IsDisabled] [bit] NULL CONSTRAINT [DF_EmployeeDetails_IsDisabled] DEFAULT ((0)),
 [WorkflowStatus] [varchar](50) NULL,
 [SecurityCustomizationId] [uniqueidentifier] NULL,
 [EmployeeName] [nvarchar](50) NULL,
 [EmployeeNo] [nvarchar](50) NULL,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 CONSTRAINT [PK_EmployeeDetails] PRIMARY KEY NONCLUSTERED
 (
 [Id] ASC
 )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
 ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
 GO
 Insert into SKList Tables (Application,TableName)
 Values ('<SketlaListandForms>','EmployeeDetails')
 /***** Note: You need to replace the repository name ‘<SkeltaListandForms>’ with your
repository name. Also replace the database ‘formsandlist’ with
 your database name *******/
 Go
```

To create an Address table

- Copy the script given below to your database server and execute it.

```
USE [formsandlist]
GO
/****** Object: Table [dbo].[Address] Script Date: 12/15/2008 15:49:29 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Address](
 [Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF__Address__Id__1E6F845E] DEFAULT
(newid()),
 [Title] [nvarchar](250) NOT NULL,
 [Application] [nvarchar](150) NULL,
 [CreatedBy] [uniqueidentifier] NULL,
 [CreatedDateTime] [datetime] NULL,
 [LastUpdatedDateTime] [datetime] NOT NULL,
 [WIP] [bit] NULL,
 [WIPItemId] [uniqueidentifier] NULL,
 [MainItemId] [uniqueidentifier] NULL,
 [ParentItemId] [uniqueidentifier] NULL,
 [RecordParentItemId] [uniqueidentifier] NULL,
 [TableId] [uniqueidentifier] NULL,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
[ItemRowIndex] [int] NULL,
[IsLatest] [bit] NULL,
[Version] [nvarchar](25) NULL,
[VersionHistory] [nvarchar](max) NULL,
[Status] [int] NULL,
[LockedBy] [uniqueidentifier] NULL,
[LockedOn] [datetime] NULL,
[ListID] [uniqueidentifier] NULL,
[SecuritySettings] [nvarchar](max) NULL,
[ItemId] [uniqueidentifier] NULL,
[ItemType] [int] NULL,
[SecurityItemId] [uniqueidentifier] NULL,
[Owner] [uniqueidentifier] NULL,
[ModifiedBy] [uniqueidentifier] NULL,
[AuditTrail] [nvarchar](max) NULL CONSTRAINT [DF_Address_AuditTrail] DEFAULT (''),
[IsDisabled] [bit] NULL CONSTRAINT [DF_Address_IsDisabled] DEFAULT ((0)),
[WorkflowStatus] [varchar](50) NULL,
[SecurityCustomizationId] [uniqueidentifier] NULL,
[CurrentAddress] [nvarchar](150) NULL,
[PermanentAddress] [nvarchar](150) NULL,
CONSTRAINT [PK_Address] PRIMARY KEY NONCLUSTERED
(
[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
Insert into SKListTables (Application,TableName)
Values ('SketlaListandForms','Address')
/***** Note: You need to replace the repository name ‘SkeltaListandForms’ with your
repository name. Also replace the database ‘formsandlist’ with
your database name *******/
Go
```

To create the Employee Details list

1. On the **Manage List** page, click **New**. The **Create List** window is displayed.
2. Enter the **Title** as **Employee Details**.
3. Enter any description to identify your list.
4. Select **Ye**s for **Is Folder Supported?** option to make your list folder supported so that you can organize your

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

list in folders.

Note: If the **Title** field is removed from the base List, the Folder is not supported for the list.

5. Click **Save and Continue**. The **Select Table** window is displayed.
6. In this case, you will map your list to the custom table. Hence, select the **Existing Table** option. Then, the **Select Table** field is displayed.
7. Select **EmployeeDetails** table from the list and then click **Next**. The **List Designer** page appears.

Note: All the fields that you have created in your database are displayed in the list. The parent items like **Title**, **EmployeeName** and **EmployeeNo** are displayed in the list.

8. Click **Submit**.
9. A form with the same name as that of the list is auto-generated when you submit the list. Select Yes if you want to regenerate the form, else select **No**. You can also edit the form using the **Edit** option from the **Forms** page.

This list data is not version enabled. Let us see how non-versioned data is managed in a list.

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Click **Employee Details**. The Employee Details items list is displayed.
3. Click **New** on the Ribbon bar to create a new list item for the **Employee Details** list.
4. Type the list item details as follows.

Field	**Data**
Title	Mr
EmployeeName	Alex
EmployeeNo	A00012

5. Click **Submit**. The list item is saved.
6. Select the list item and right click the mouse. Since the Employee Details list does not have a child list, it is not version enabled. So you cannot see the **Save As** context menu in the right click options.

Next, add one child table **Address** to the **Employee Details** list to make this list version enabled.

To add an Address list

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Select the **Employee Details** list and click **Edit** on the **Ribbon bar**. The **Create List** window is displayed.
3. You can edit the **Description** and Folder Supported option. The **Title** field will be disabled for editing. Click **Save & Continue**. The **Forms Designer** page is displayed with the **Employee Details** list.
4. Drag and drop the **List** table in to the orange bar of the list name area. The **Select Table** window is displayed.
5. Select the **Existing Table** option to map your child table to the custom table.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Select the **Address** table from the list and then click **Select**. The **Form Property** window is displayed.
7. Enter the **Tag name** as **Address**.**List Table**
8. Click **Save**.
9. Now, a child list table is also added to the Employee Details list and this data will be versioned.
10. Click **Submit**. The **Re-Generate Form** dialog box appears.
11. A form with the same name as that of the list is auto-generated when you submit the list. Select **Yes, regenerate the default form** and click **OK** if you want to regenerate the form, else select **No, do not regenerate the default form** option and click **OK**. Select **Yes** and regenerate the default form with the new fields (Current Address and Permanent Address).
12. Remove the **Title** Field from the **Address** child form. See Also Deleting a Control.
13. Next, disable the repeating property of the **Address** since a person will have only one current address and permanent address. For that, click the **Properties** button for the **Address** control. The **Address Properties** window is displayed.
14. Delete the **Mapped XML Node**.
15. Select **False** from the **Is repeating** drop-down list.
16. Click **Save**.
17. Click **Submit**.

Now, you can create list items for this and save it as a different version. To verify this:

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Click **Employee Details**. The Employee Details items page is displayed.
3. Click **New** on the Ribbon bar to create a new list item for the Employee Details list.
4. Type the list item details.

Field	**Data**
Title	Mr
EmployeeName	John
EmployeeNo	J012
CurrentAddress	21, HSR Layout, Bangalore
PermanentAddress	2nd Cross Rd, Powai, Mumbai

5. Click **Submit**. The list item is saved.
6. Then select the list item and right-click it to open the context menu. Since the Employee Details list is version enabled, you can see the **Save As** context menu in the right-click options. This menu will help you to create different versions for the list item.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To save an item as next major version:

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Select **Lists** and click **Employee Details**.

   Note: Our objective is to create a next major version for the item.
3. Right click the required item list.
4. Click **Next Major Version**. The **Save List Item** dialog box appears. You can specify your comments in the text box provided. Let us leave the field blank as it is not a mandatory field.
5. Click **Save New Version**. An information message appears.
6. Click **OK**. An information message appears.
7. Click **OK**.
8. Right-click the same item list and select **Show All Versions**.

   Note: In the same way, you can create Next Minor Version, Specific Version and Next version of the list item. If you make any changes in the main list i.e. Employee Details, this will reflect in all the versions. If you make changes in the Address items, that will not affect any other items or other versions.

Version enabling for a child table with multiple records

Next you will see how a child table with multiple records can be version enabled. For this purpose, add one repeating child item **Qualification** to the existing list.

The **Qualification** table has the following schema:

- Title
- University
- Year of Pass

Before starting with this scenario, create a table **Qualification** in your database.

To create a Qualification table

1. Login to your **SQL Server**.
2. Copy the following script to your server and execute it.

```
CREATE TABLE [dbo].[Qualification](
 [Id] [uniqueidentifier] NOT NULL CONSTRAINT [DF__Qualification__Id__1E6F845E] DEFAULT (newid()),
 [Title] [nvarchar](250) NOT NULL,
 [Application] [nvarchar](150) NULL,
 [CreatedBy] [uniqueidentifier] NULL,
 [CreatedDateTime] [datetime] NULL,
 [LastUpdatedDateTime] [datetime] NOT NULL,
 [WIP] [bit] NULL,
 [WIPItemId] [uniqueidentifier] NULL,
 [MainItemId] [uniqueidentifier] NULL,
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
 [ParentItemId] [uniqueidentifier] NULL,
 [RecordParentItemId] [uniqueidentifier] NULL,
 [TableId] [uniqueidentifier] NULL,
 [ItemRowIndex] [int] NULL,
 [IsLatest] [bit] NULL,
 [Version] [nvarchar](25) NULL,
 [VersionHistory] [nvarchar](max) NULL,
 [Status] [int] NULL,
 [LockedBy] [uniqueidentifier] NULL,
 [LockedOn] [datetime] NULL,
 [ListID] [uniqueidentifier] NULL,
 [SecuritySettings] [nvarchar](max) NULL,
 [ItemId] [uniqueidentifier] NULL,
 [ItemType] [int] NULL,
 [SecurityItemId] [uniqueidentifier] NULL,
 [Owner] [uniqueidentifier] NULL,
 [ModifiedBy] [uniqueidentifier] NULL,
 [AuditTrail] [nvarchar](max) NULL CONSTRAINT [DF_Qualification_AuditTrail] DEFAULT
('''),
 [IsDisabled] [bit] NULL CONSTRAINT [DF_Qualification_IsDisabled] DEFAULT ((0)),
 [WorkflowStatus] [varchar](50) NULL,
 [SecurityCustomizationId] [uniqueidentifier] NULL,
 [University] [nvarchar](50) NULL,
 [YearofPass] [int] NULL,
 CONSTRAINT [PK_Qualification] PRIMARY KEY NONCLUSTERED
 (
 [Id] ASC
 )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
 ) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
 GO
```

To add a Qualification child table to the Employee Details list

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Select the **Employee Details** list and click **Edit** on the **Ribbon** bar. The **Create List** window is displayed.
3. You can edit the **Description** and **Folder Supported** options. The **Title** field will be disabled for editing. Click **Save & Continue**. The **Forms Designer** page is displayed with the **Employee Details** list.
4. Drag and drop the **List Table** in to the orange bar of the list name area. The **Select Table** window is displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Select the **Existing Table** option to map your child table to the custom table.
6. Select the **Qualification** table from the list and then click **Select**. The **List TableForm Property** window is displayed.
7. Enter the **Tag** name as **Qualification**.
8. Click **Save**. The **List Designer** page appears. Now, a child list table is also added to the **Employee Details** list and this data will be versioned.
9. Click **Submit**. The **Re-Generate Form** dialog box appears.
10. A form with the same name as that of the list is auto-generated when you submit the list. Select **Yes, regenerate the default form** and click **OK** if you want to regenerate the form, else select **No**, **do not regenerate the default form** option and click **OK**. On regenerating the form, all the changes made to the form will be lost. So it is recommended that you do not select this option. You can also edit the form using the **Edit** option from the **Forms** page.

To edit the list form

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. Select the **Employee Details** form and click **Edit** on the Ribbon Bar.
3. Replace the **Qualification** Form with the **Grid** control.
4. Click **Save**.
5. Click **Submit**.

Now, you can create list items for this and save it as a different version. In this scenario, you will add multiple records in the **Qualification** grid and save it as different version. To verify this:

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **Manage List**. The **Manage List** page appears.
2. Click **Employee Details**. The **Employee Details** items page is displayed.
3. Click **New** on the Ribbon bar to create a new list item for the Employee Details list.
4. Type the list item details.

Field	**Data**
Title	Mr
EmployeeName	Alex
EmployeeNo	A00352
CurrentAddress	21, BTM Layout, Bangalore
PermanentAddress	706, Oxford, 21, Thane

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Data
Qualification 1	
Title	MCA
University	University of Bangalore
YearofPass	2002
Qualification 2	
Title	ME
University	University of Mumbai
YearofPass	2005

5. Click **Submit**. The list item is saved.
6. Select the list item and right-click the mouse. Since the **Employee Details** list is version enabled, you can see the **Save As** context menu in the right-click options. This menu will help you to create different versions for the list item.
7. Our objective is to create a next version for the item.
8. Click **Next Version**. The **Save List** Item dialog box appears. You can specify your comments in the text box provided. Let us leave the field blank as it is not a mandatory field.
9. Click **Save New Version**. An information message appears.
10. Click **OK**. An information message appears.
11. Click **OK**.
12. Right-click the same item list and select **Show All Versions**.

Note: In the same way, you can create Next Minor Version, Specific Version and Next version of the list item. If you make any changes in the main list i.e. Employee Details, this will reflect in all the versions. If you make changes in the Address items, that will not affect any other items or versions.

Sort List items using DisplaySortOrder Property

In a **Process Order** list, assume that there are columns like **OrderNo**, **Name**, **Description**, **Quantity**, **Price** and **Total**. By default, the user wants to display the list items in a sorted order whenever a list loads at first time.

Steps

1. Create a **Process Order** list as follows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Control	Name
TextInput	Title
IntegerInput	OrderNo
TextInput	Name
IntegerInput	Quantity
DecimalInput	Price
DecimalInput	Total

2. Add a few items in the Process Order list using **New** option.
3. Open Process Order list. Click **Edit** to set the **DefaultSortOrder** property.
4. Set the **DefaultSortOrder** property as **OrderNo**, **Name**, **Quantity**.
5. Click **Save** and then click **Submit**. The **Re-Generate Form** dialog box appears.
6. Select **Yes**, regenerate the default form and click **OK**.
7. The list items are displayed in a sorted order after the **Process Order** list is loaded.

Set Custom View for a List Control from a Custom ASPX Page using API

The following code demonstrates how to set a custom view for a List control from a custom ASPX page using API.

```
//Create a ViewItem
Skelta.Repository.List.ListDefinition listDefn = null;
Skelta.Repository.List.ListItemCollection _ListItems = null;
listDefn = ListDefinition.GetList(new Skelta.Core.ApplicationObject("repo1"),"Employee");
_ListItems = new Skelta.Repository.List.ListItemCollection(listDefn);
String _ ViewName = "myview";
Skelta.Repository.List.ListView.ListViewItem _ListViewItem = null;
_ListViewItem = Skelta.Repository.List.ListView.ListViewManager.GetListViewItem(listDefn,
"myview");
if (_ListViewItem != null)
{
ListViewManager _ViewManager = new
Skelta.Repository.List.ListView.ListViewManager(_ListViewItem);
_ListItems.IsTopLevelFormView = _ViewManager.IsTopLevelView;
List<string> filterColumns = new List<string>();
string _FilterQueryWithParam = string.Empty;
string _WFStatusParamQry = string.Empty;
bool _WFStatusFilter = false;
Workflow.NET.Config configObj = new Workflow.NET.Config(_ApplicationName);
IDataHandler dbHandler = DataHandlerFactory.GetDataHandler(configObj);
List<IDataParameter> _ViewFilterParamsCol =
_ViewManager.GetFilterParameterCollection(dbHandler, out _FilterQueryWithParam,out
_WFStatusFilter,out _WFStatusParamQry);
   if (_WFStatusFilter)
   {
_ListItems.WFStatusFilter = true;
_ListItems.WFStatusParamQry = _WFStatusParamQry;
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
}
string viewCols;
_ViewManager.GetDisplayColumnsInfo(out viewCols);
string totalRows ="";
DataTable itemsDataTable = null;
if (_ViewManager.DisplayAllItems)
{
itemsDataTable = _ListItems.GetRecords(_UserId, Guid.Empty, "", "", "", out totalRows, _"",
_"", Fields.GridSpecific, Display.AllItems, viewCols, _FilterQueryWithParam,
_ViewFilterParamsCol);
}
else
itemsDataTable = _ListItems.GetRecords(_UserId, Guid.Empty, "", "", "", out totalRows,
_CustomQuery, _CustomQueryJoinClause, Fields.GridSpecific, Display.MainItems, viewCols,
_FilterQueryWithParam, _ViewFilterParamsCol);

}
```

Lists Integration

List integration is used to facilitate the import and export of data to and from the List.

Introduction

Microsoft SQL Server Integration Services (SSIS) is an extensive data integration platform that is used to transfer, transform, and merge information from various sources, and then load onto multiple systems.

Workflow Management List Data Exchange (SLDE) is built on the SSIS data flow component and is flexible to support various industry standard data sources. The SLDE components allow importing and exporting data into List, data store.

To make the List infrastructure flexible to load application data, an important capability is support for data import from and to various data sources. The SLDE components include List Source and List Destination for SSIS data flow, and Workflow Execution Task for SSIS control flow. List is also an Extraction, Transformation and Loading (ETL) tool.

Objective of this Guide

The objective of this guide is to provide information on using List Integration components with SSIS.

Audience for this Guide

This target audience for this guide is:

- An SSIS Package developer who is proficient in SSIS.
- SQL Server Experts.

Guide Conventions

The conventions used in this guide are as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Typographical Convention	Represents
Bold	A menu or a menu option.
Hyperlink	A hyperlink to a related topic.
All CAPS	Keys on the keyboard.
Type	Information you need to enter.
Note:	A note provides additional information about a certain topic.

Abbreviations and Acronyms

Acronym	Full Form
SSIS	SQL Server Integration Component
BIDS	Business Intelligence Development Studio
SSDT	SQL Server Data Tools

Note: Workflow Management List Data Exchange is available only on request and may take time to deliver based on the target environment.

Need for Lists Integration

List integration is used to facilitate the import and export of data to and from the List.

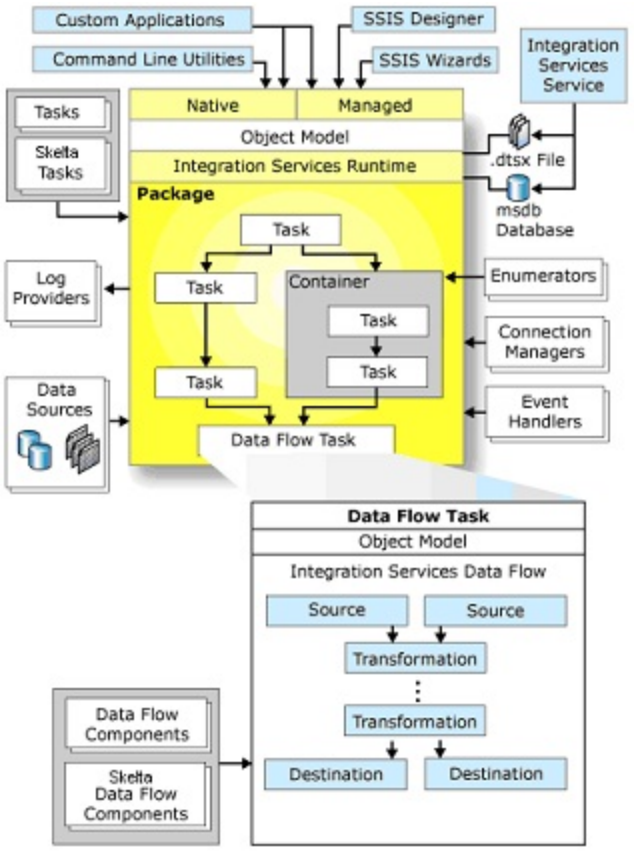
Need for Lists Integration Components

User data is usually scattered across flat files, CSV, and XL, to name a few. The List Integration components are needed to import data from various data sources and to export data to a different format.

About Lists Integration Components

List Source Data Flow and List Destination Data Flow are the two List Integration components. SSIS components are built on custom task and custom data flow component. As the SSIS components are built on SSIS, the architecture is the same as that of SSIS. The following chart shows the SSIS architecture:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Key Features and Benefits

Since List Source Data Flow and List Destination Data Flow are created for SSIS, all the feature and benefits provided by SSIS are available. These include the ability to:

- Handle multiple sources with different formats (flat file, CSV, etc.) for import and export of data.
- Handle Structured, semi-structured, and unstructured data.
- Huge data volumes.
- Change cryptic data into meaningful information when it is imported or exported.
- Load multiple destinations (flat file, CSV, etc.) at the same time.
- Ability to extract and transform data.

The key benefits of the List Integration components include the ability to:

- Import from many types of data
- Export to many types of data
- Merge data
- Support transactions

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Prerequisites

For prerequisites, refer to the Install Integration Services.

Note:
- Ensure SQL Server client is installed on the machine where the AVEVA Work Tasks client component is installed.
- Ensure the SQL Server Integration Services are running on the SQL Server. Refer to the configuration section for more details on configuring the component.
- While importing data, ensure:
-- the format of date/time fields is the same as in the SQL database.
-- null values are blank instead of “NULL”.
-- Boolean values are either “false” or “true” instead of “0” or “1” respectively.

Installation and Configuration

The installation steps include the following:

1. Copy the main assembly to the appropriate component folder under the 64-bit Program Files folder.
2. Install the appropriate assemblies in the global assembly cache (GAC).
3. Stop and restart the SSIS service (because it caches the list of installed components).
4. Restart SSDT, and BIDS or Visual Studio.
5. If necessary, make manual adjustments to the list of custom components in the Toolbox.

Note: If repository settings are not available in SSIS, then install Newtonsoft assemblies from the **<AVEVA Work Tasks Installation Path>\Bin** folder to the global assembly cache (GAC).

64-Bit Installation

SQL Server Client Components

1. Copy Skelta.SSIS.Pipeline.ListDestination.dll and Skelta.SSIS.Pipeline.ListSource.dll to **C:\Program Files\Microsoft SQL Server\<<SQL Server Version Number>>\DTS\PipelineComponents**.

   For <<SQL Server Version Number>> refer to File Locations for Default and Named Instances of SQL Server.
2. Register the DLL files to Assembly GAC. Refer to Install an Assembly into the Global Assembly Cache.

About Configuration

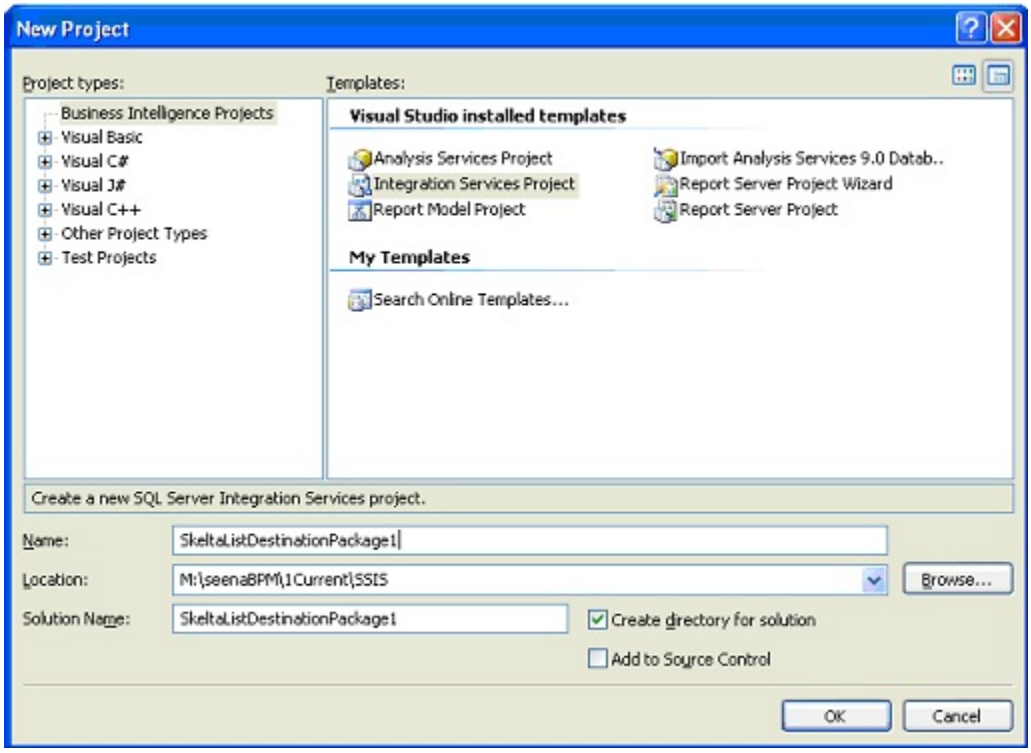
After completing the installation, you need to add the List Integration component to the Toolbox. Follow the steps given below:

1. Create a new SSIS project.
2. Drag and drop a Data Flow task.
3. Add the component to the Toolbox.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

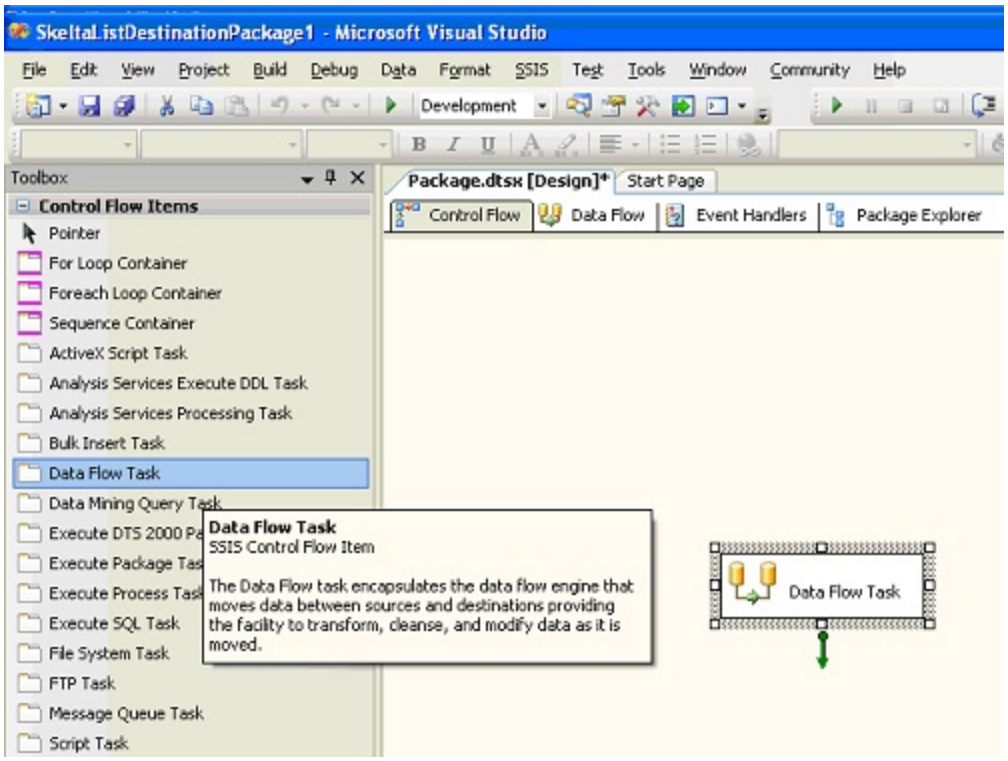
To configure List Destination component, perform the following steps:

1. Create a new SSIS project.
2. Start Visual Studio.
3. Click **File > New > Project > Business Intelligence Projects > Integration Service Project**.
4. Enter **Project Name**.
5. Click **OK** to create the project. A blank Control Flow is created.



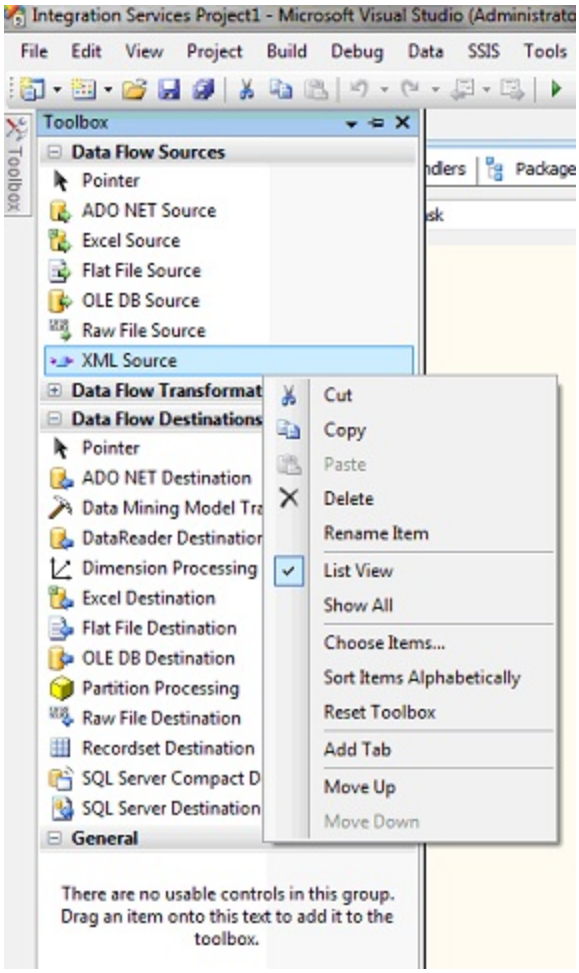
6. Drag and drop a Data Flow Task item into the Control Flow.
7. Double-click on the task to switch to the data flow view.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



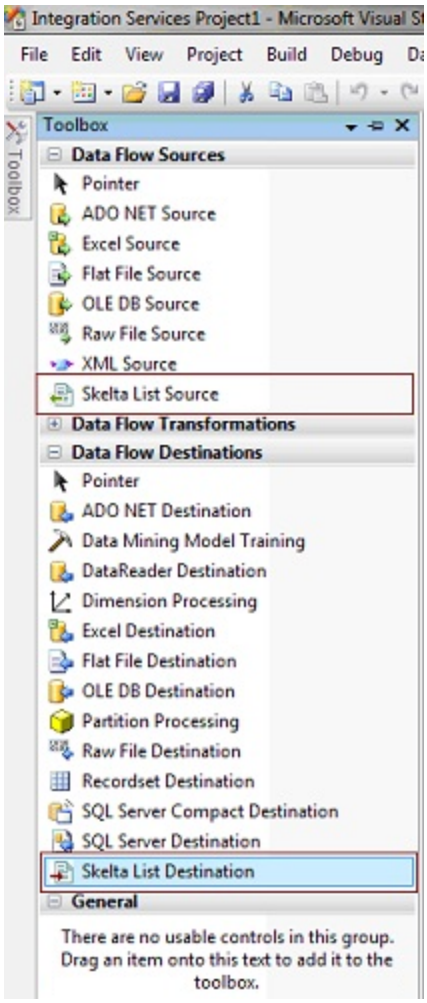
8. Go to the **Toolbox** of the Data Flow Sources and in the select **XML Source** drop-down list, check the **List View** option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



9. You can see the List Source under the Data Flow Sources and List Destination under Data Flow Destinations.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



List Source Data Flow

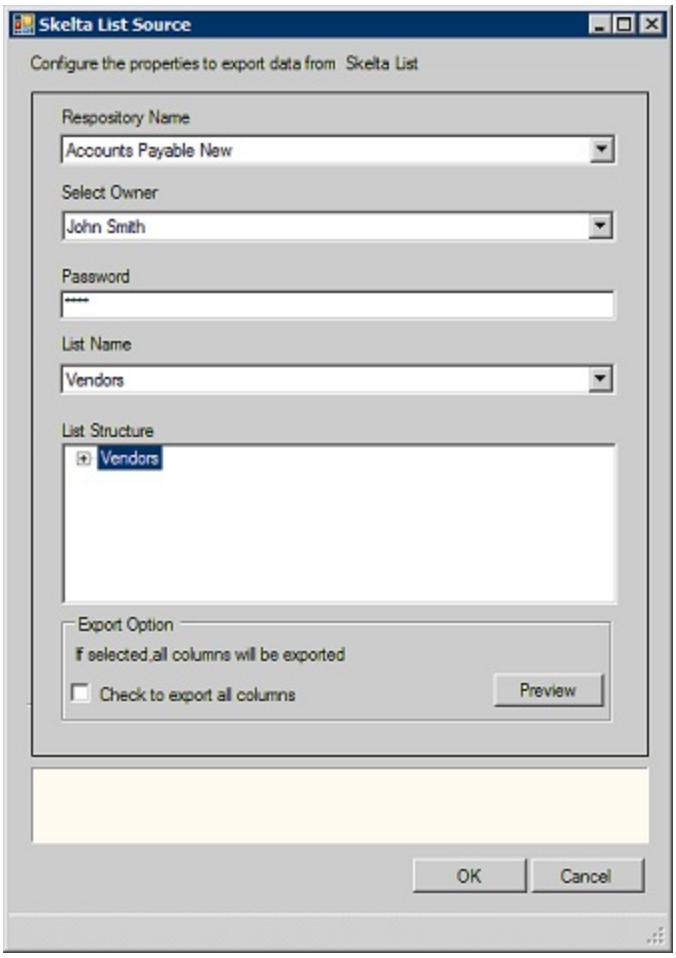
This section includes information about import/export of data from/to any SQL Server Integration Services (SSIS) supported data flow components:

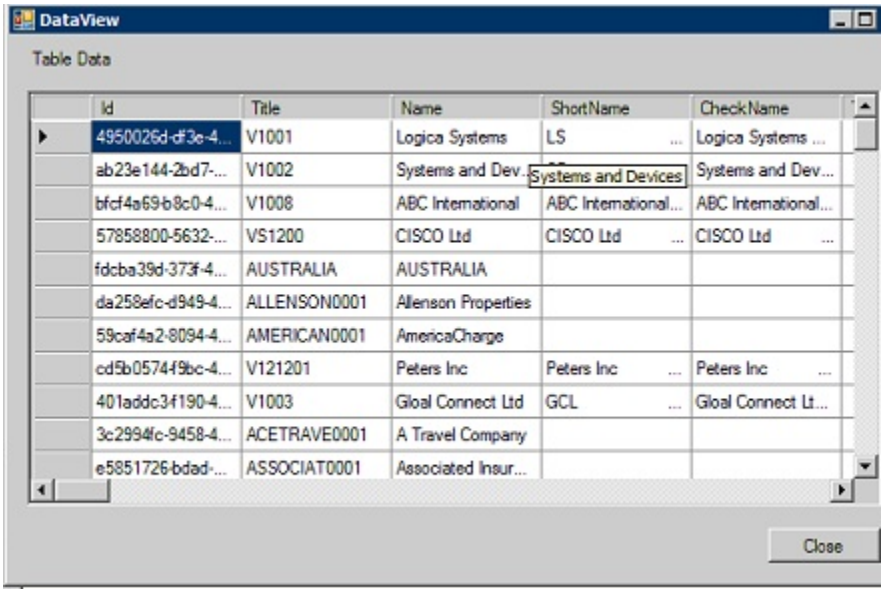
- About List Source Data Flow
- List Destination Data Flow

About List Source Data Flow

The List Source Data Flow feature is used to export data from List to any SSIS-supported destination. The List Source component is available in the list of Source dataflow components. You can select the list from which the data is to be exported. If the list is mapped to multiple tables, you can select the tables from which the data is to be exported.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.





Field	Description
Repository name	Select the repository name.
List Name	Select the **List Name** from which the data is to be

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

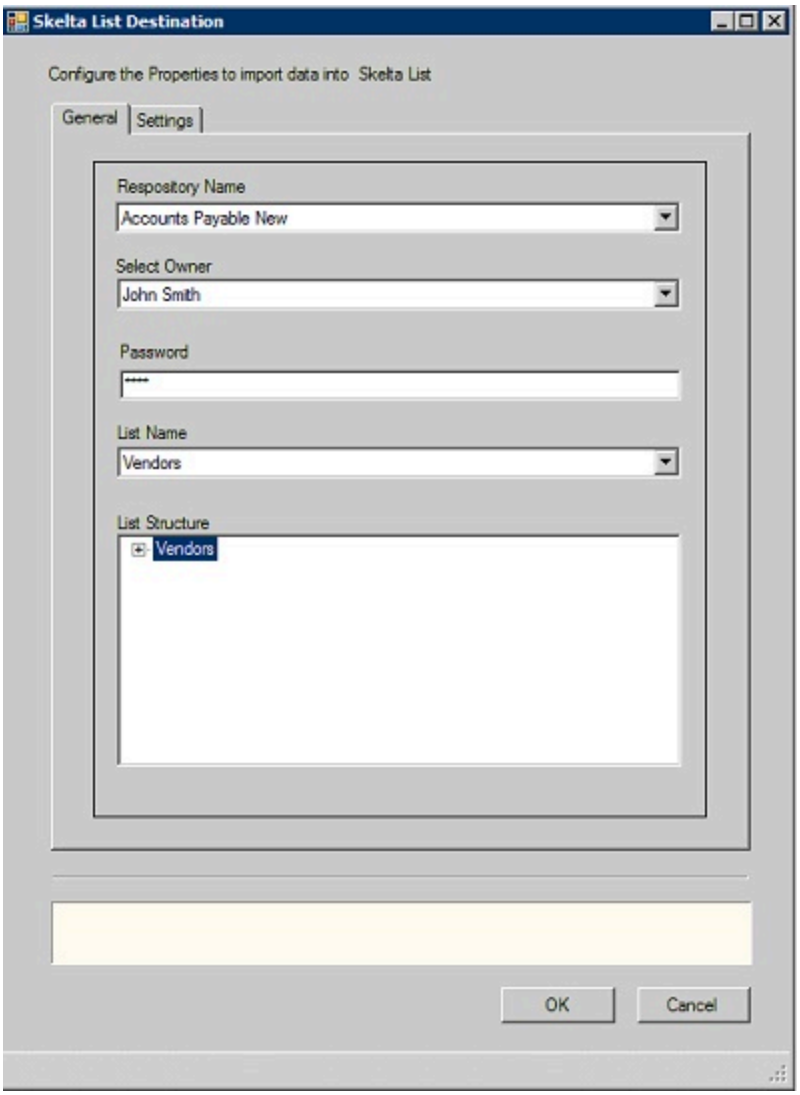
Field	Description
	exported.
List Structure	Select the **Table Name** from which the data is to be imported.
Preview	See the **Preview** of the selected table.
Export Option	Select the **Check** to export all columns if you want to export all the columns. If not, only the user-defined columns will be exported.
Select Owner	Select the **Owner** from the drop-down list.
Password	Key in the password.

List Destination Data Flow

You need to use the **List Destination Data** flow component to import data from any SSIS-supported data flow component. The List Destination component is available in the destination dataflow components. You can select the list to which the data is to be imported. If the list is mapped to multiple tables, you can select the tables to which the data is to be exported.

General Tab

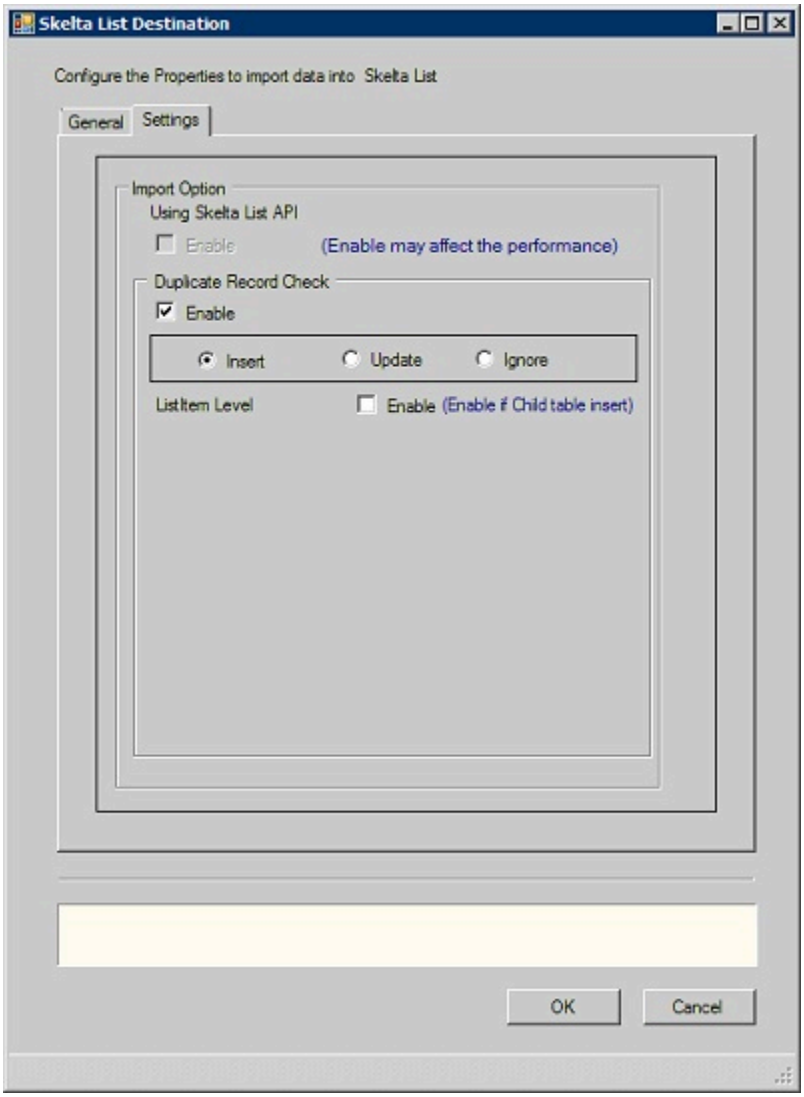
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Field	Description
Repository name	Select the repository name.
List Name	Select the **List Name** from which the data is to be imported.
List Structure	Select the **Table Name** from which the data is to be imported.
Select Owner	Select the owner from the drop-down list.
Password	Key-in the password.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Settings Tab



Field		Description
Using List API	Enabled	Select this box for importing data using the List APIs. Using this option may affect the performance of the system.
Duplicate Check	Enable	Select this box to enable checking of duplicates.
	Insert	Select this radio button if you need to insert the item to the list even if it is a duplicate item.
	Update	Select this radio button if you need

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

		to update the item.
	Ignore	Select this radio button for checking duplicate records. If a duplicate record exists, it will be ignored.

Import and Export Data

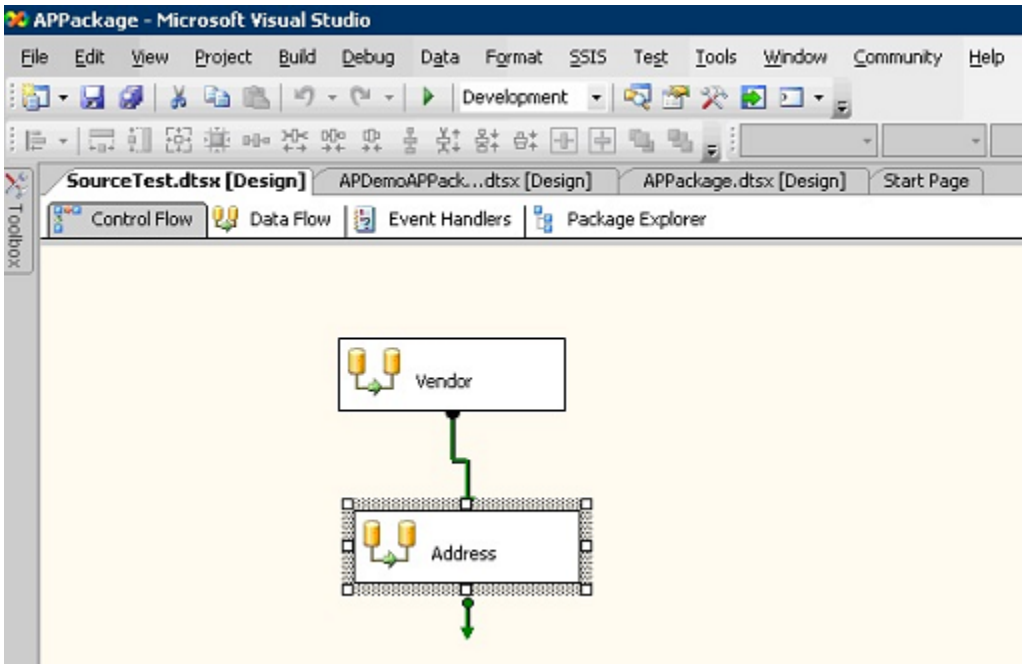
In this section, you will learn to use the List integration components in an SSIS package. The examples refer to Excel as a data source and data destination. You can also use other data sources such as a DBF file, CSV, flat file, to name a few.

How to Export Data to Excel?

Scenario

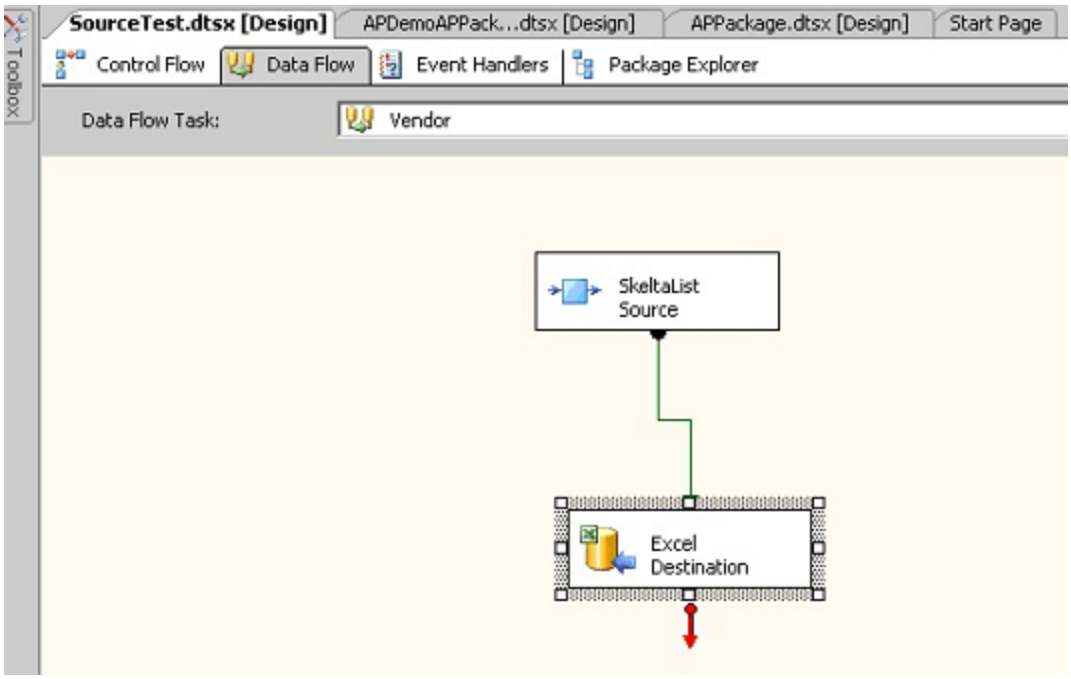
You need to export vendor list data from the List. The SSIS package consists of two dataflow tasks, one for the vendor and one for VendorAddress. The vendor list is mapped to two tables - Vendor and Address. The data from these tables is exported to an Excel file called VendorDetails.xls.

1. Open **SQL Server Business Intelligence Development Studio**.
2. In **Control Flow**, drag and drop the **Dataflow** task from the **Toolbox**.

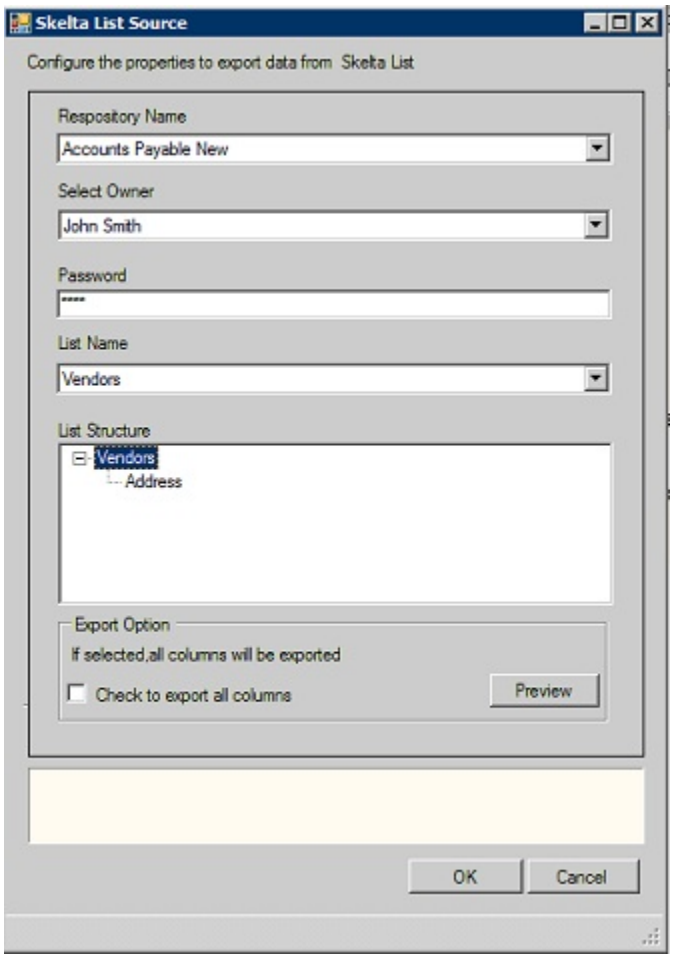


3. Click the **Dataflow** tab of the **Vendor** data flow task.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



4. Drag and drop the **List Source** data flow component.
5. Double-click the **List source** to open the List Source property window.

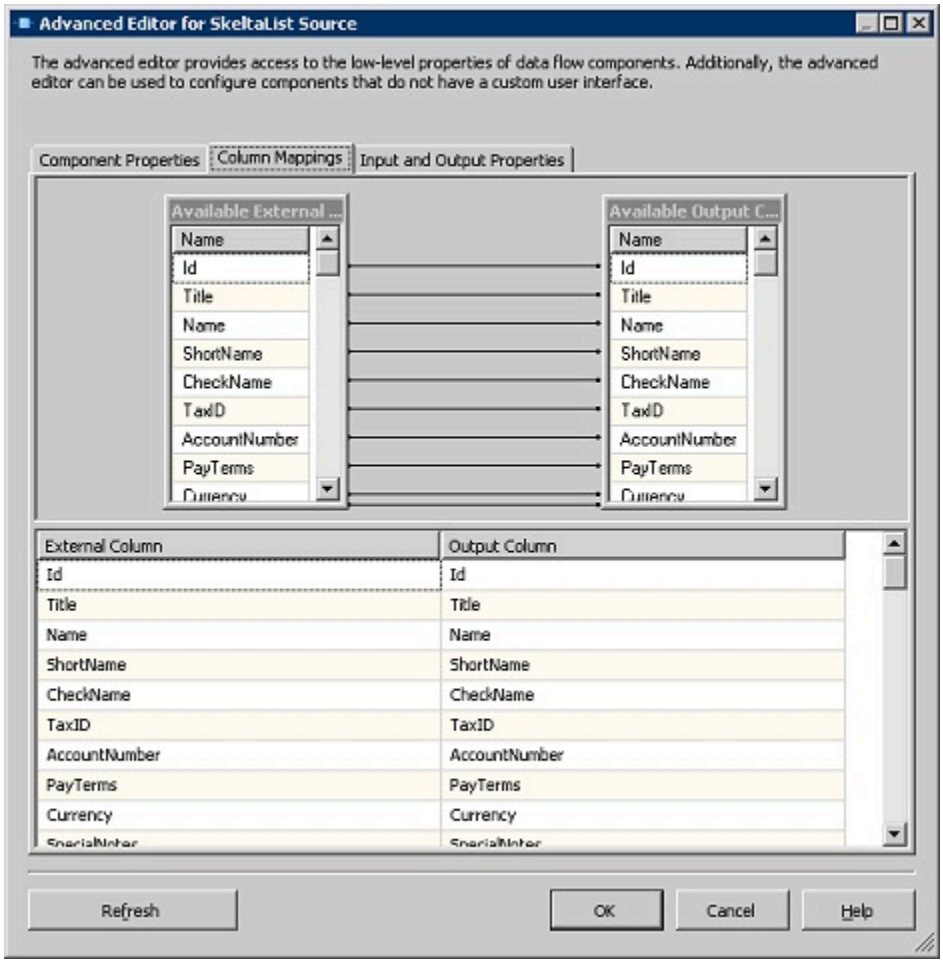


© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Next you need to set the properties. Select the Repository name.
7. Select owner from the drop-down list.
8. Enter the Password.
9. Select Vendor from the list names.
10. Select the Vendor Table from the List Structure tree.

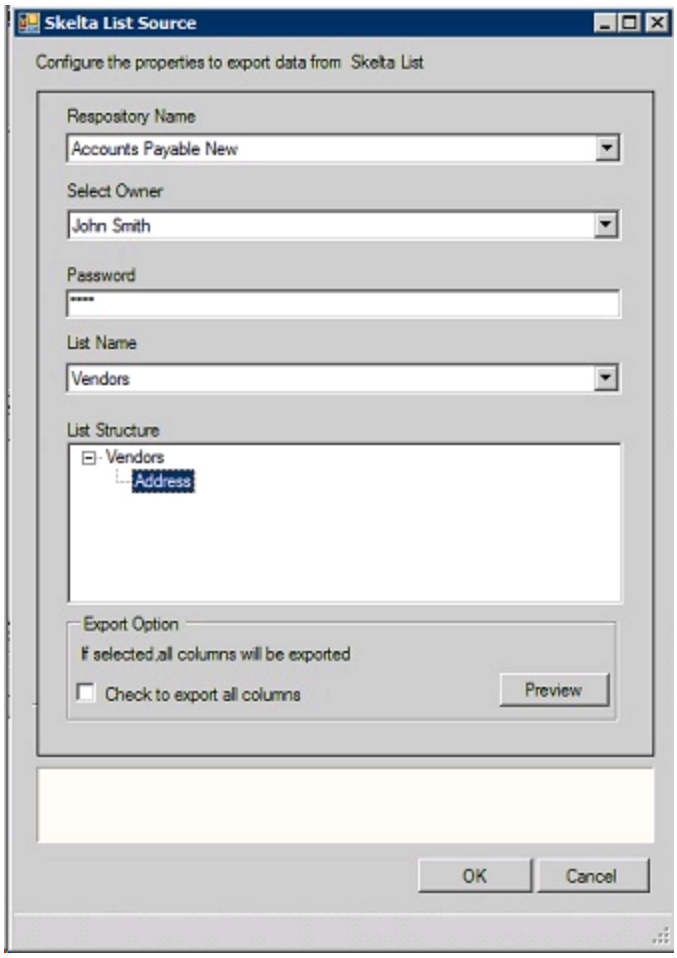
Note: Checking Export all columns option will check all the columns.

11. Click **OK** to save.
12. Drag the **Excel Destination** component. Link the List source with the Excel destination.
13. Next, you need to map the columns.
14. Right-click the List Source component and select the advanced editor.
15. Go to column mapping and map the corresponding columns from List source column to Excel sheet destination columns.

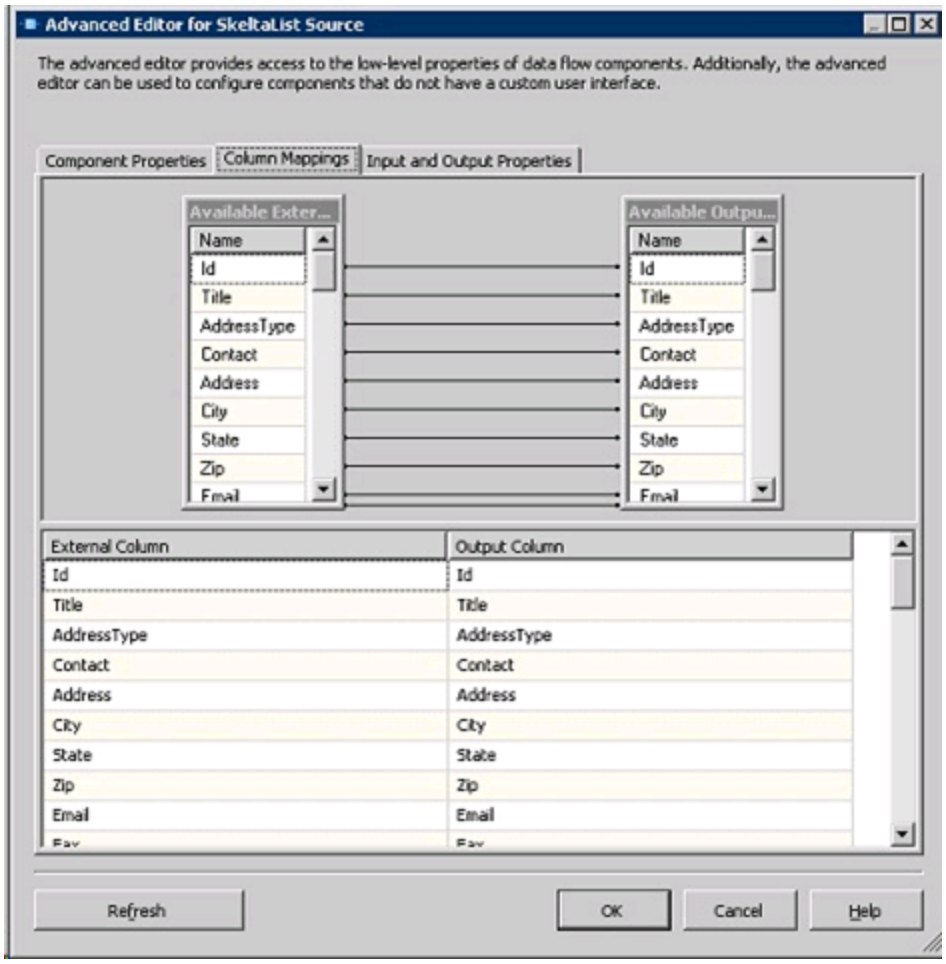


16. Go to control flow design dataflow for **Address**.
17. Drag and drop the Dataflow task and rename it as **Address**.
18. Go to dataflow task. Drag and Drop List Source component and configure the properties as explained in the above steps.

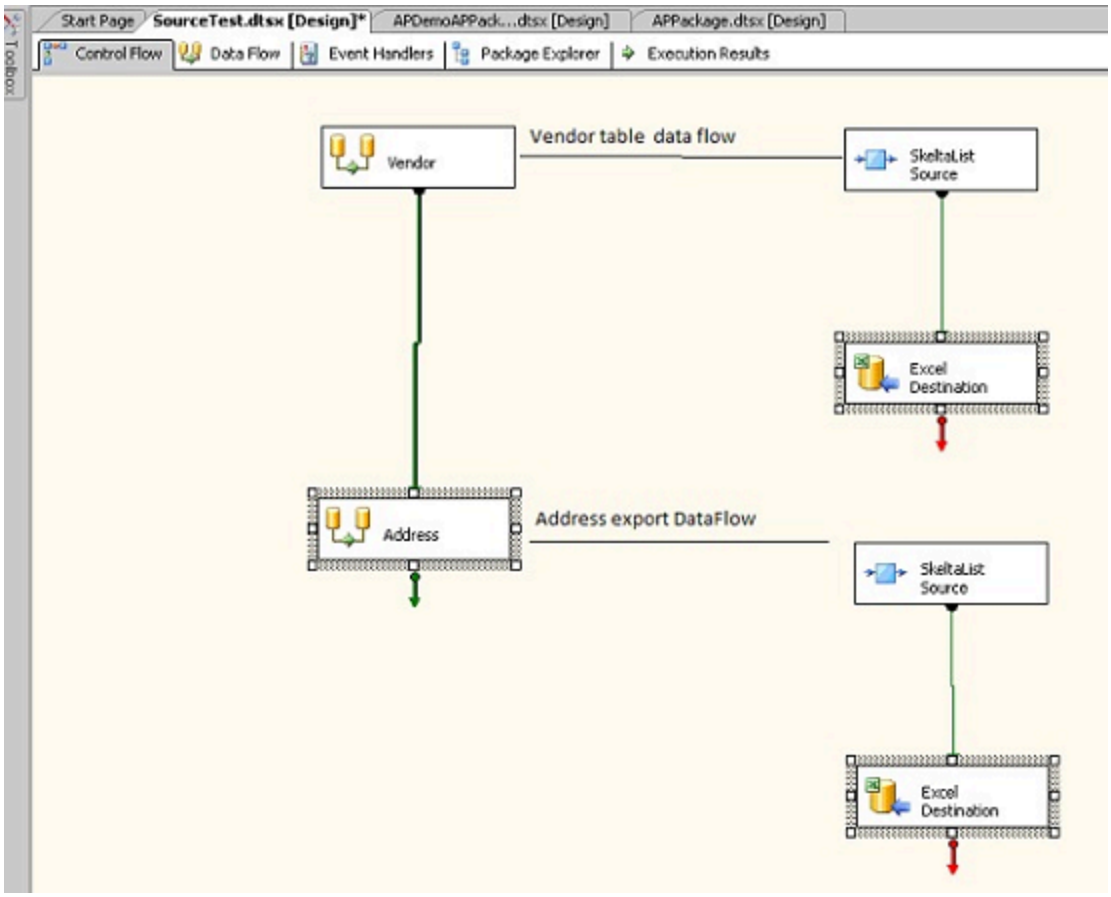
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



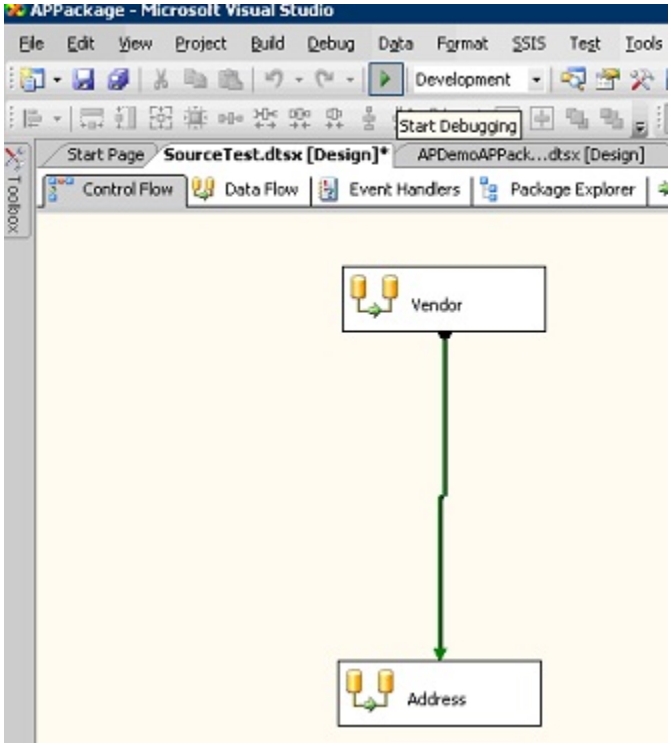
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

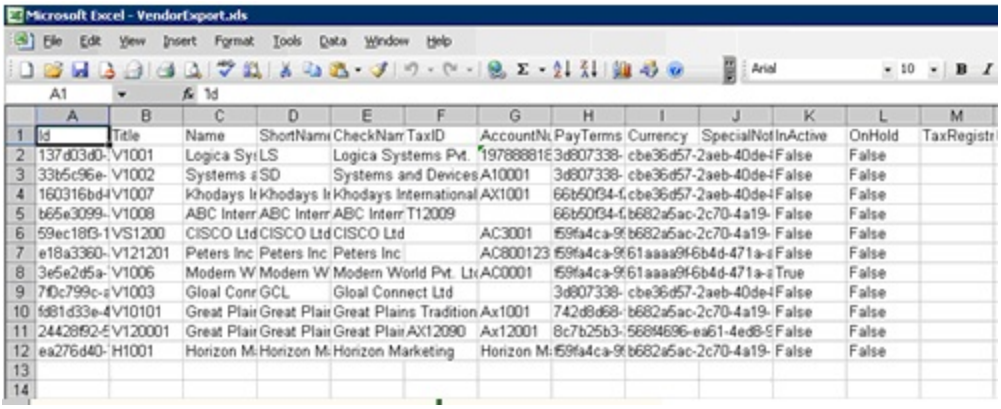


19. In the Package diagram, click on the **Start Debug** option in the BID to execute the package.



20. You have successfully exported data from the vendor list to **VendorExport.xls**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



How to Import Data from Excel?

Scenario

Importing data from a comma-separated flat file to a List called Employee List.

The Employee List is mapped with two tables, EmployeeMaster and EmployeeDetails. The comma-separated flat file contains employee information. The employee first name and last name needs to be inserted in the Employee Main Table. The employee details (Address, Phone, Email, Department, etc) needs to be inserted in the EmployeeDetails Table.

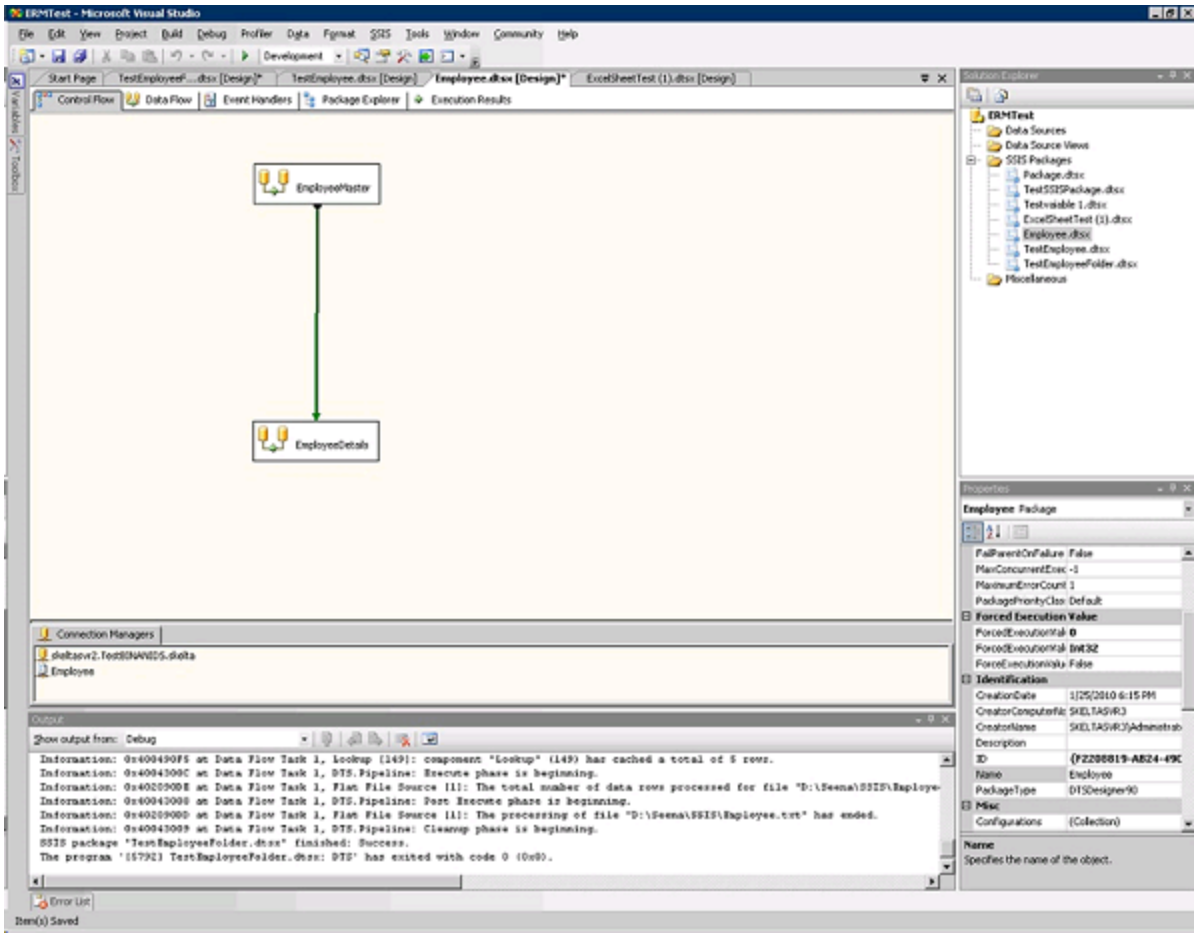
To accomplish the above-mentioned requirement, create an SSIS Package with two Dataflow tasks - one for EmployeeMaster and the second for EmployeeDetails. The steps to create the package:

1. Open SQL Server Business Intelligence Development Studio (BIDS).
2. Add the dataflow task from the Toolbox.
3. Drag and drop the flat file source component to the first dataflow task.
4. Configure the flat file data source.
5. Add the List Destination component and configure it in the second data flow task.
6. Map the column.
7. Open SQL Server Business Intelligence Development Studio.

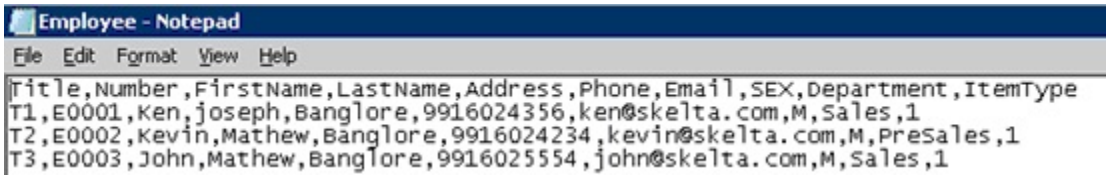
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

8. Drag and drop the flat file source component.
9. Go to Control Flow.Drag and drop the dataflow task from the Toolbox.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

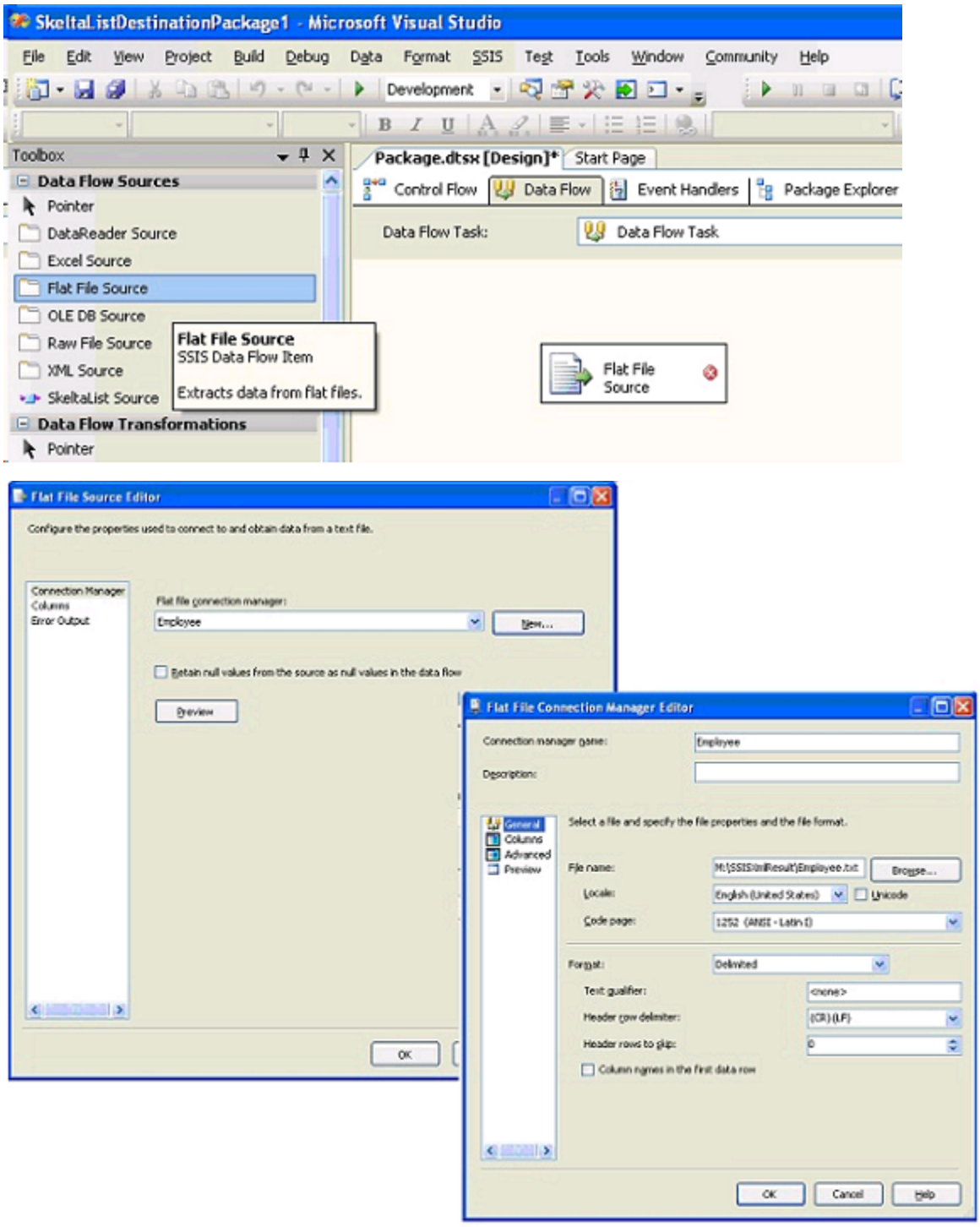


10. Go to Dataflow tab of the Employeemaster data flow task. The Employee.txt has the following data:



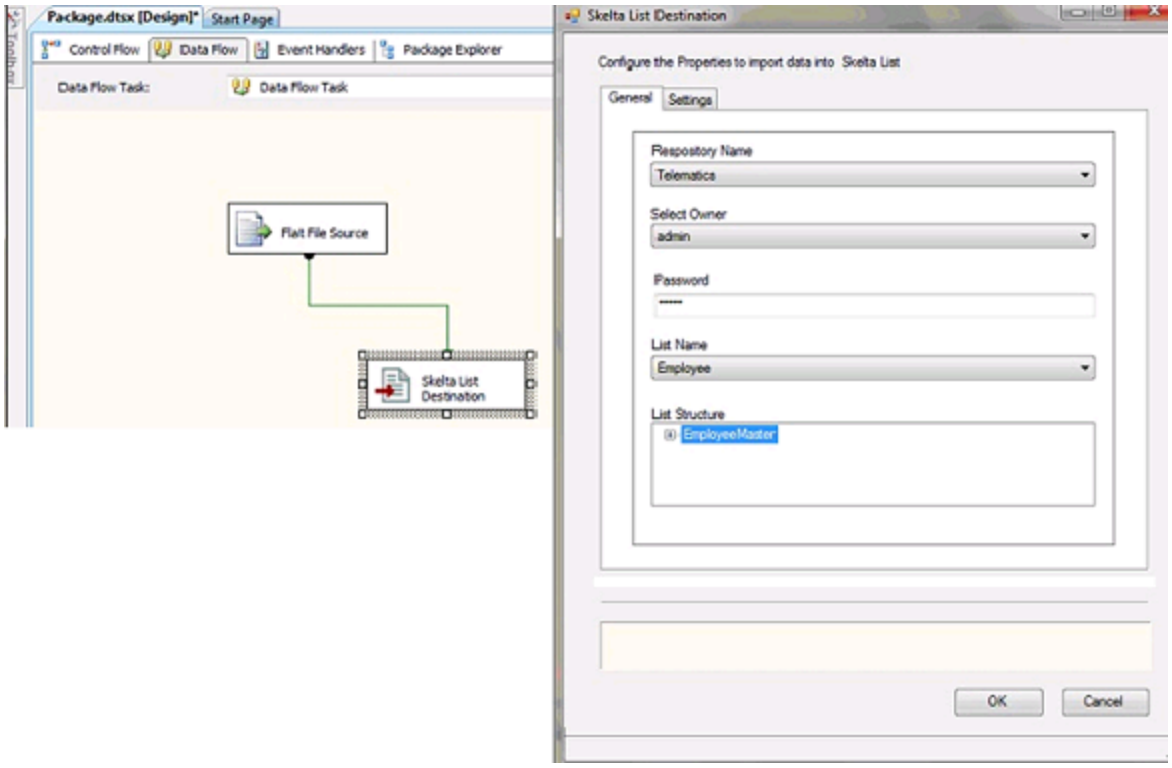
11. Drag and drop the Flat file Source component, and configure the Flat file source as shown below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

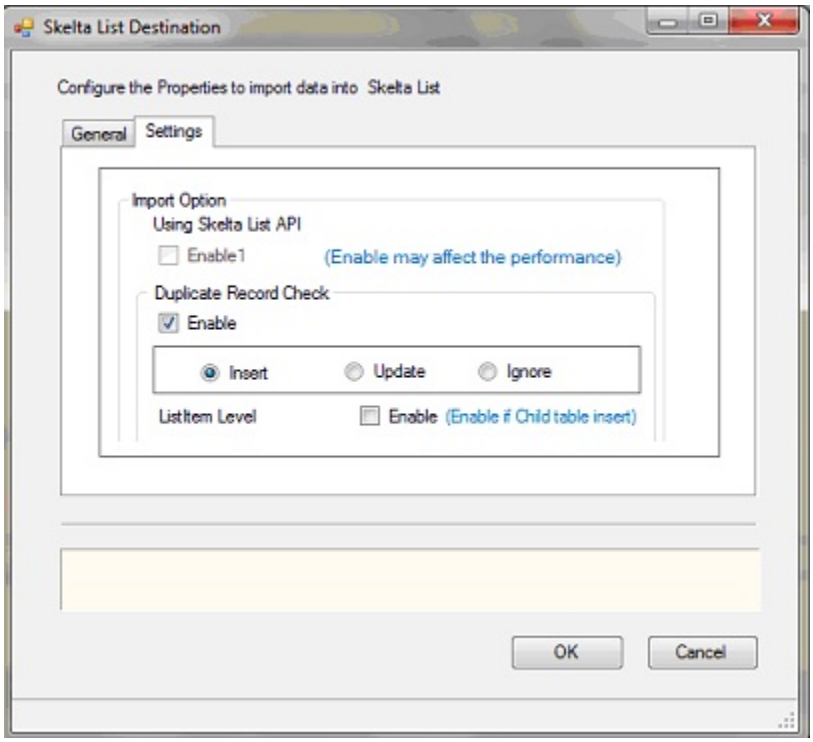


12. Next, you need to add List Destination component. Drag and drop the List Destination data flow component. Double click on the List destination to open the Destination property window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



13. Next, you need to configure the General and settings properties. Select General Tab, Repository name, List name Employee, and Employee Main Table from the List Structure tree.

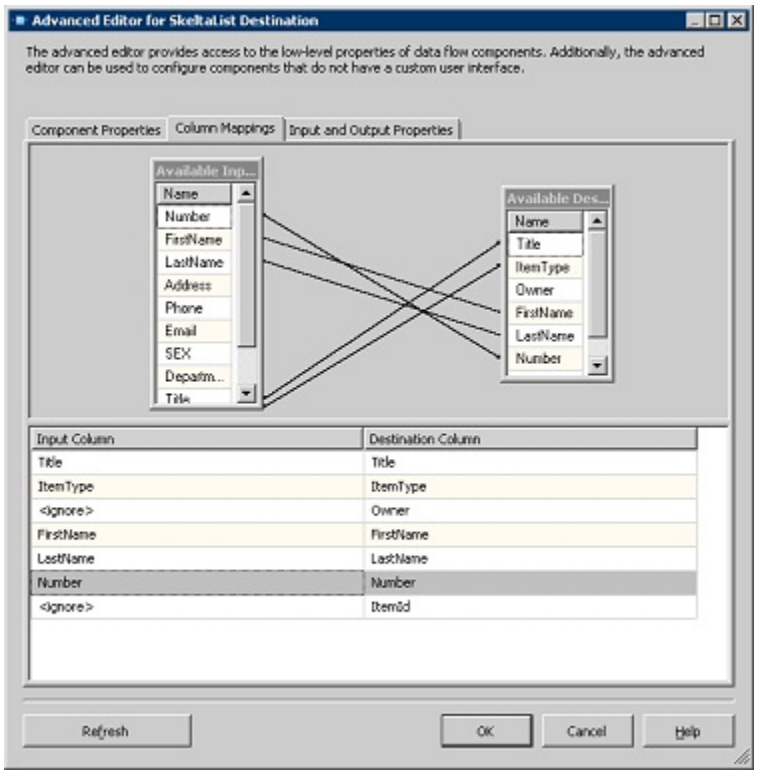


14. Next, you need to set the Export settings. Click the Settings tab.
15. Duplicate Check
    - Enable: Select this box to enable checking of duplicates.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Insert: Select this radio button if you need to insert the item to the list even if it is a duplicate item.
- Update: Select this radio button if you need to update the item.
- Ignore: Select this radio button for checking duplicate records. If a duplicate record exists, it will be ignored.

16. Click **OK** to save.
17. Next, you need to do Column Mapping. Right-click the List Destination component and select the advanced editor. Go to column mapping and map the corresponding columns from Flat file to List Destination columns.



18. Go to control flow design dataflow for EmployeeDetails.
19. Drag and drop Dataflow task and rename it as Employee Details.

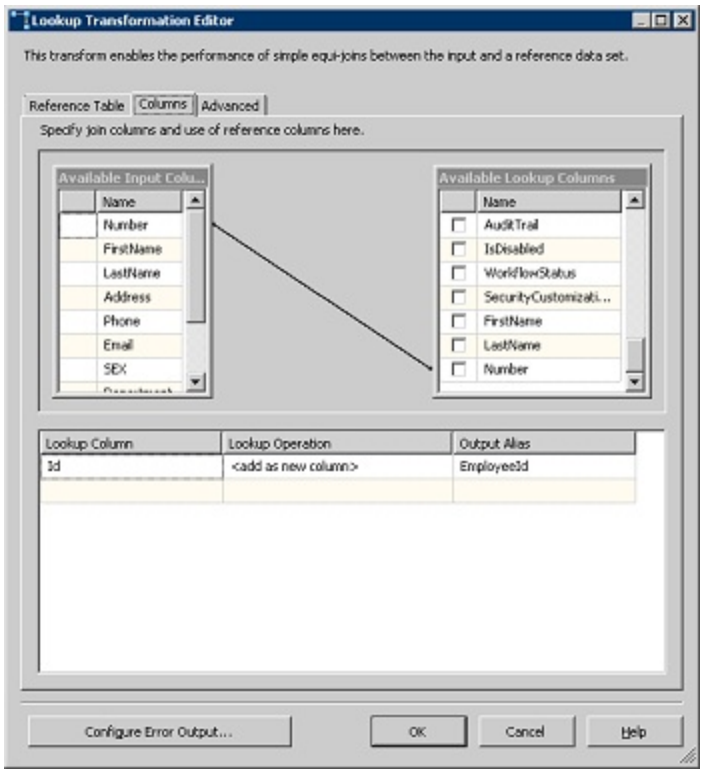
    Note: For the Child Table entry, you need to map the RecordParentItemId field in the Listdestination column.

    EmployeeId is mapped to the Destination column RecordParentItemId. To get the EmployeeId Lookup, the Dataflow Transformation component needs to be used. In the Lookup component, map the number with the Employeemaster table number field and take the Id of the Employee from the Employee Maser Table.

Lookup dataflow Transformation component configuration

1. Drag and drop lookup component in the Dataflow task. Link the Flat file source to Lookup component.
2. In the columns tab map the Number with the EmployeeMaster Column Number, take the Lookup Column as Id.Give Output Alias as EmployeeId. This column can be used to map to RecordParentItemid in the Listdestination Columns mapping.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

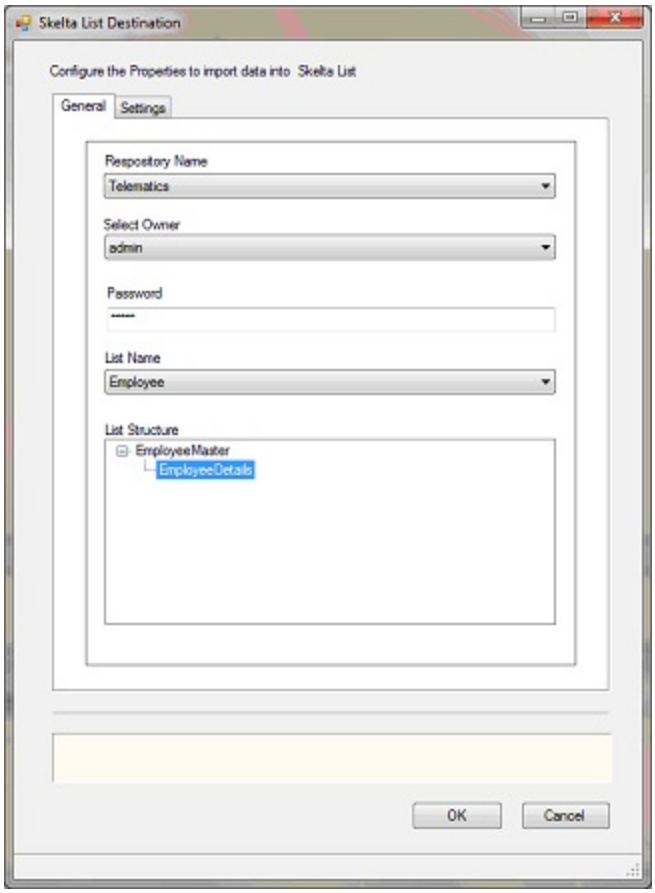


3. Drag and drop the ListDestination.

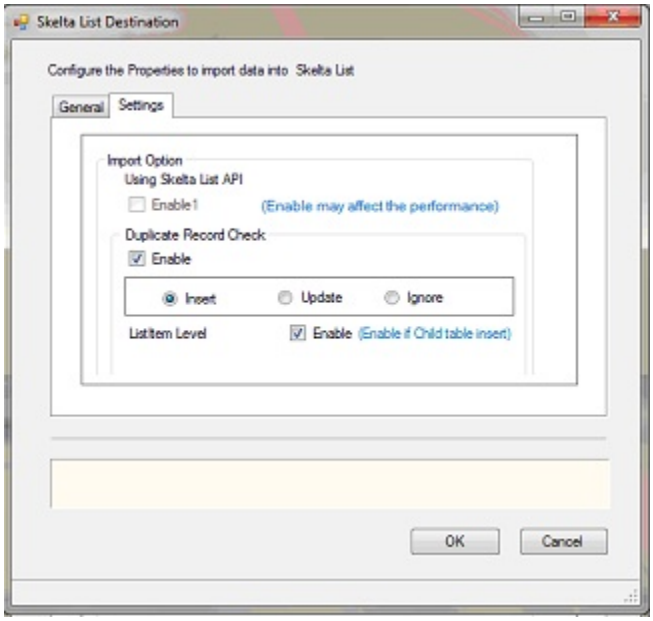
General Configuration

1. Repeat 14th Step. 1. Next, you need to set the Export settings. Click the Settings tab.
2. Select EmployeeDetails table.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

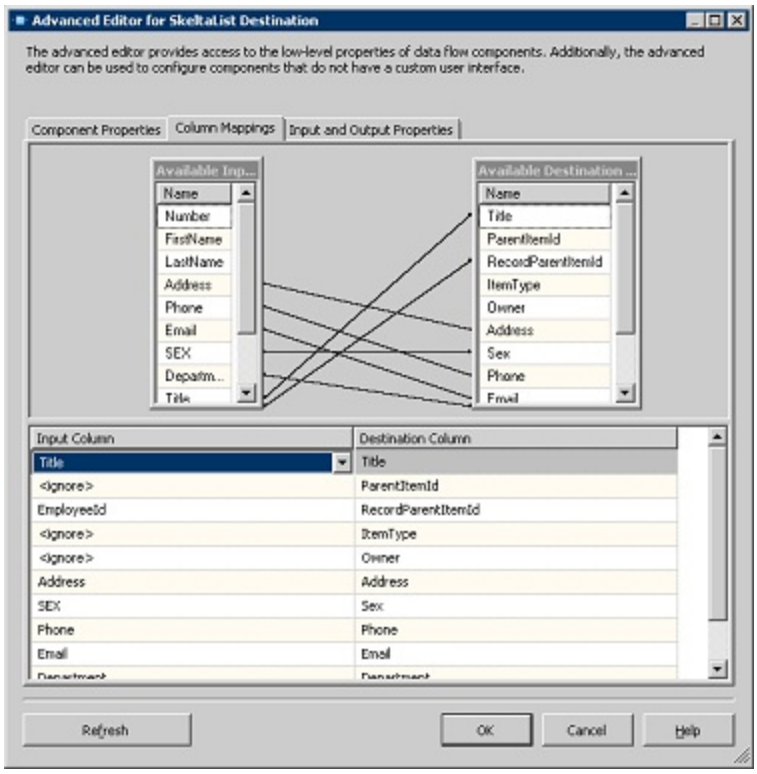


3. Next, you need to set the Settings configuration. If the insertion is to Child table Enable ListItemLevel.

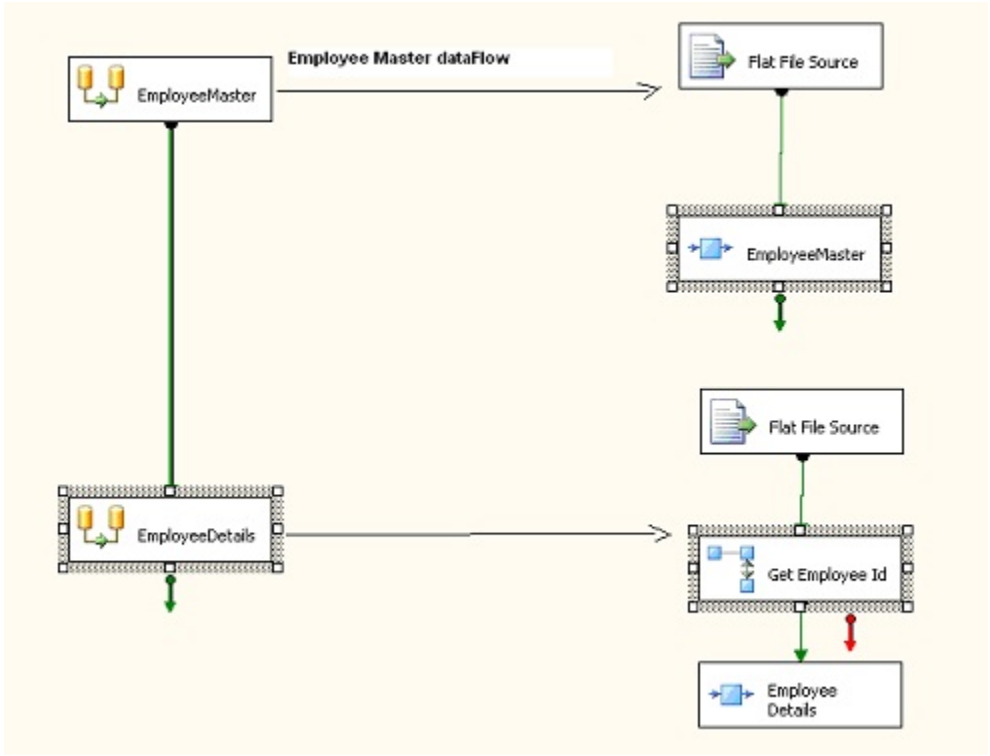


4. Next, you need to set the Column Mapping.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

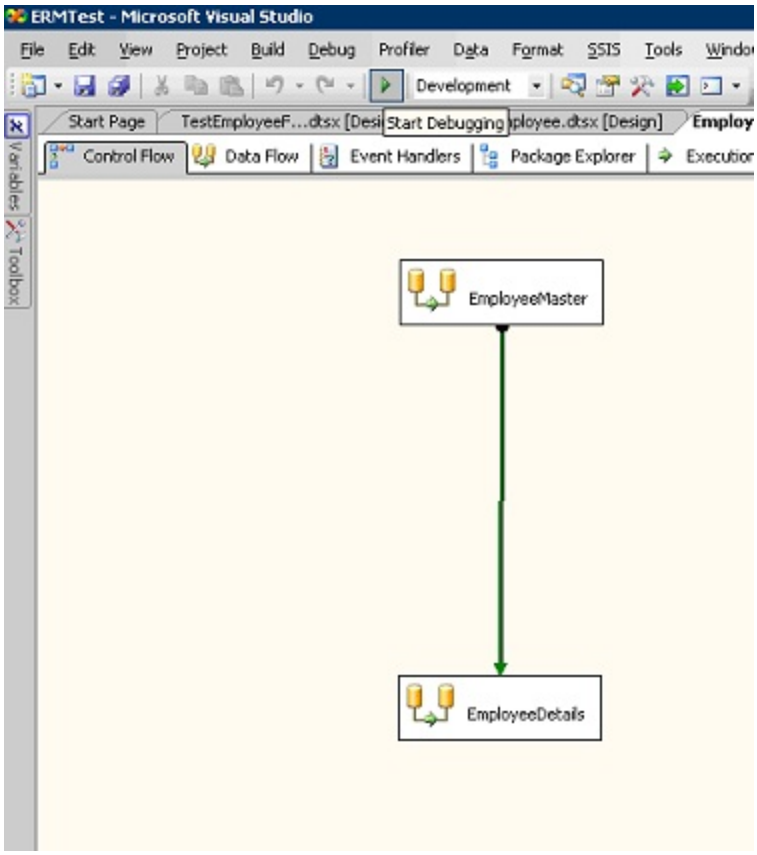


5. The Package diagram is shown below.



6. To Execute the package, Click the Start Debug option in BID.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



7. After execution of the Package, the Employee data from Employee.txt has been imported in the Employee list.

Troubleshooting Lists

This section provides information on the known issues, the issues encountered while working with Lists and the resolution.

Known Issues

List Item

While trying to create a new List Item, if the **Edit** and **View** security rights are set for the Form, you will not be allowed to create the new List Item.

Troubleshoot Lists

Issue

When the **OnCreate** event is triggered, the system is not able to find the newly added list item while trying to update the list XML variable.

Possible Cause

OnCreate event fires just before saving the list item, which triggers the **Update Variable** activity before creation

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

of the list item. This results in the system not being able to find the newly added list item.

Resolution

Add a **Human Activity** before the **Update Variable** activity.

OR

Add a **Scheduler Activity** before the **Update Variable** activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Quickflow User Guide

This *Quickflow User Guide* is for users to create a system that executes the business logic faster using "micro workflow execution engines" running parallel to the workflow engine.

Getting Started

Quickflow helps you to create a system that executes the business logic faster using "micro workflow execution engines" running parallel to the workflow engine.

Quickflow application supports following features:

- Design and execute transient workflows
- Associate Quickflow with various event providers like Forms, LogicApp, FileWatcher, Scheduler, and Message Broker
- Enable the activity status persistence
- Integrate with ArchestrA Identity Manager (AIM) / System Management Server

Prerequisites

Before deploying Quickflow, ensure that the following prerequisites are met:

- ERLANG is installed
- RabbitMQ is installed
- RabbitMQ Management Plugin and AMQP 1.0 Plugin is enabled
- RabbitMQ users are configured
- Valid certificate is available
- AIM is configured and registered

Install ERLANG/OTP

Download **ERLANG/OTP** (version 27.1.2) from *http://www.erlang.org/downloads* and install it.

Note: Run the Erlang installer using an Administrative account so that the registry key for the RabbitMQ installer is present in the system.

Install RabbitMQ

Download **RabbitMQ** (version 4.0.4) from *https://www.rabbitmq.com/install-windows.html* and install it.

For more information on RabbitMQ, refer *https://www.rabbitmq.com.*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Enable RabbitMQ Management Plugin and AMQP 1.0 Plugin

After installing RabbitMQ, enable the RabbitMQ Management plugin and AMQP 1.0 plugin.

Open the RabbitMQ Command Prompt and run the following commands:

- `rabbitmq-plugins enable rabbitmq_management` (to enable RabbitMQ Management plugin)
- `rabbitmq-plugins enable rabbitmq_amqp1_0` (to enable AMQP plugin)

Also, ensure the following:

- You have access to http://localhost:15672 to ensure access to the RabbitMQ Management site.
- You use localhost. The guest credentials works only if the hostname is localhost.

Configure RabbitMQ Users

Configure additional users and provide appropriate access rights.

1. Log in to the **RabbitMQ Management Site**.
2. Select the **Admin** tab.
3. In **Add a user** section, enter the name and the password.
4. Provide access rights to all channels and topics to the user.
5. Click **Add user**.

Certificate to be used with the Quickflow Engine

Any communication from and to a Quickflow Engine is secured. For the secured communication, the system requires a certificate. The System Management Server can be used to import/generate the certificate. You can also use a commercial certificate if you have or create a new certificate.

1. Open the Certificate Management Console.
2. Find the appropriate certificate you want to use.
3. Select **All Tasks** and **Export** the certificate.
4. Follow the steps provided by the wizard. Save the certificate (for example, MyCertificate.pfx) on the desktop.

Configure and Register Token Service

Configure Token service using System Management Server (SMS). For more information on configuring Token service, refer Configuring and Registering Token Service.

After configuring Token service, you can register it with AVEVA Work Tasks 2023 SP1 using **Farm Configuration Wizard** or **Central Configuration** > **Token Service Setting**.

Note: Only one AIM server can be registered for a Farm.
- While configuring Token service, ensure the SMS server is online.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Deploy Quickflow

You can deploy the Quickflow Engine using the **Central Configuration** page.

Prerequisites

- Install **AVEVA Work Tasks 2023 SP1**. For more information on what you need to do prior to deploying a Quickflow Engine, see Prerequisites.

To deploy Quickflow Engine

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, and then **Central Configuration**.
2. Click **Quickflow Settings**.

   The **Quickflow** page appears.
3. Enter the following details under **Message Broker Details** section:
   - **URI**: Enter the URI of the server where Message Broker is installed. URI is the machine name or the IP address of the machine (e.g. 192.168.0.100).
   - **User Name**: Enter the user name of the Message Broker server. The user name should not include space.
   - **Password**: Enter the password of the Message Broker server. The password should not include space.
   - **Port**: Enter the port for Message Broker server. You cannot enter the port number that is already used by other application.
   - **AMQPS**: Select the check box for the secured connection with Message Broker server.

     Note:
     - If you use Azure service bus, select the AMQPS check box and provide the Azure bus Uri and the credentials.
     - If you use other Message Brokers, uncheck the AMQPS check box and enter the Uri of the server where Message Broker is installed. Uri is the machine name or the IP address of the machine.
   - **Token Service**: Select the registered AIM token service or the AVEVA Connect token service (to connect through AVEVA cloud).

     Note:
     - If you access Central Configuration using AVEVA Connect authentication, then deploy Quickflow using AVEVA Connect token service.
     - If you access Central Configuration using Azure Active Directory authentication, then deploy Quickflow using AAD token.
     -If you reconfigure the Token service, then Quickflow must also be re-deployed.
4. Enter the following details under **Server Details** section:
   - **Server Name**: Displays the list of AVEVA Work Tasks servers available.
   - **Https Port**: Enter the port number for the server.
   - **Certificate**: Select the valid certificate file.
   - **Password**: Enter the password of the attached certificate.

     Note:
     - When you redeploy the Quickflow, ensure to select the Certificate again while deploying.
     - You cannot update the **Server Details** section, if you have selected the **Secure Web Sites** option while configuring **Security Settings** in Farm Configuration Wizard. The configured Security Settings details will

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

be considered while deploying Quickflow.

5. Click **Deploy**.

   The Quickflow application gets integrated in the AVEVA Work Tasks system.

   Note:
   - When the AVEVA Work Tasks - Client Service is restarted, the Quickflow application pool should be started again.
   - If the Quickflow deployment fails, see the Troubleshoot section.
   - If the execution time exceeds 10 seconds, the Quickflow fails showing an error.

After the Quickflow is successfully deployed, ensure that:

- Quickflow Status Monitor service is installed and running.
- Quickflow web application is created under the "AVEVAWorkTasksSites" site.

Create Quickflow

In this section, you will learn how to create a Quickflow. As an example, the "Invoke Web API" activity is configured to invoke a web API from the Quickflow.

Prerequisites

- Download and install AVEVA Work Tasks 2023 SP1.
- Create a repository.
- Log on to the repository using Administrator credentials.

This section is categorized into:

- Designing a Quickflow
- Publishing and executing a Quickflow
- Viewing the result in the Execution View

To design a Quickflow

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**.

   The **Workflows** page appears.
3. On the action bar, click **New,** and then click **Quickflow.**

   The **New Quickflow** wizard appears.
4. In the **Title** box, type a name for the Quickflow, and in the **Description** box, type a description for the Quickflow.
5. Select the **Persistence** checkbox to enable the activity status persistence.

   Note:
   - For a Quickflow with parent and child Quickflows, the activity status persistence depends on the the

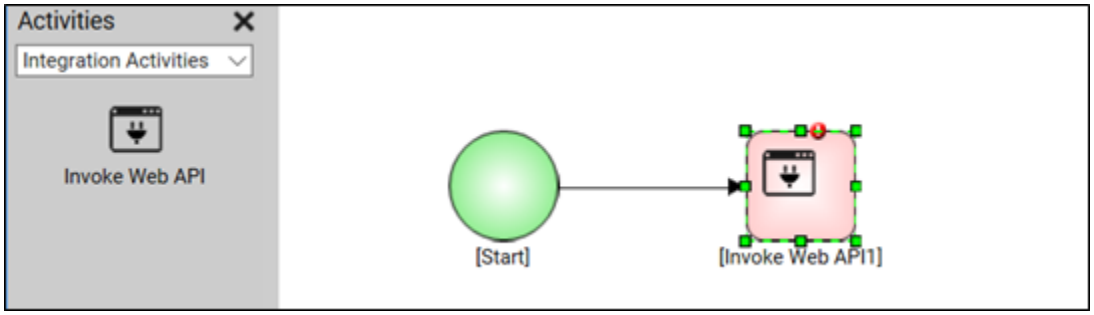
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

persistence enabled or disabled for the parent Quickflow and not on the child Quickflow persistence property.
For example, if the persistence is enabled for parent Quickflow, the activity status will persist even if the child Quickflow has disabled persistence property.
- If the persistence is enabled, the activity status will not be reflected when the execution time exceeds.

6. Click **Save & Continue**.

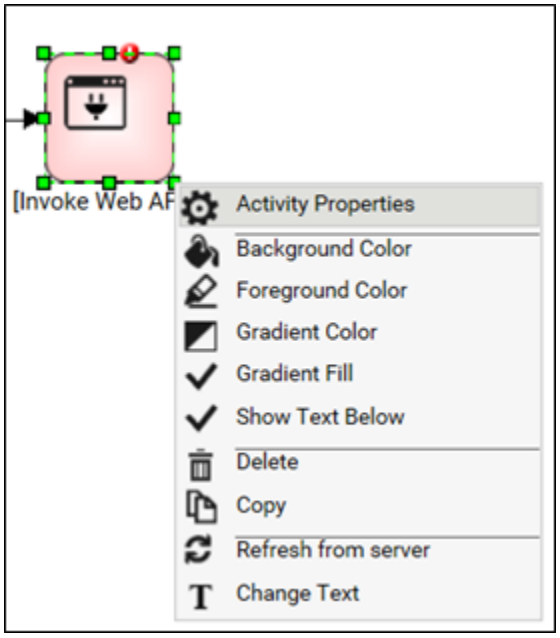
   The Quickflow is created, and gets opened in the **Process Designer** page.

7. Drag an **Invoke Web API** activity from the **Integration Activities** to the design area.



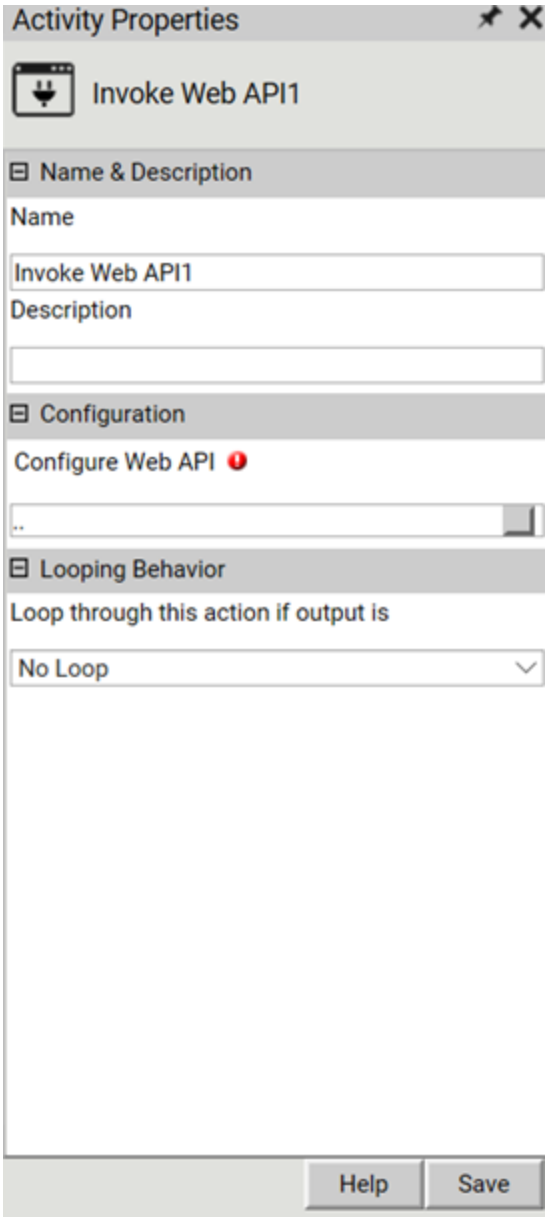
8. Right-click **Invoke Web API** activity, and then click **Activity Properties**.

   The **Activity Properties** window appears.



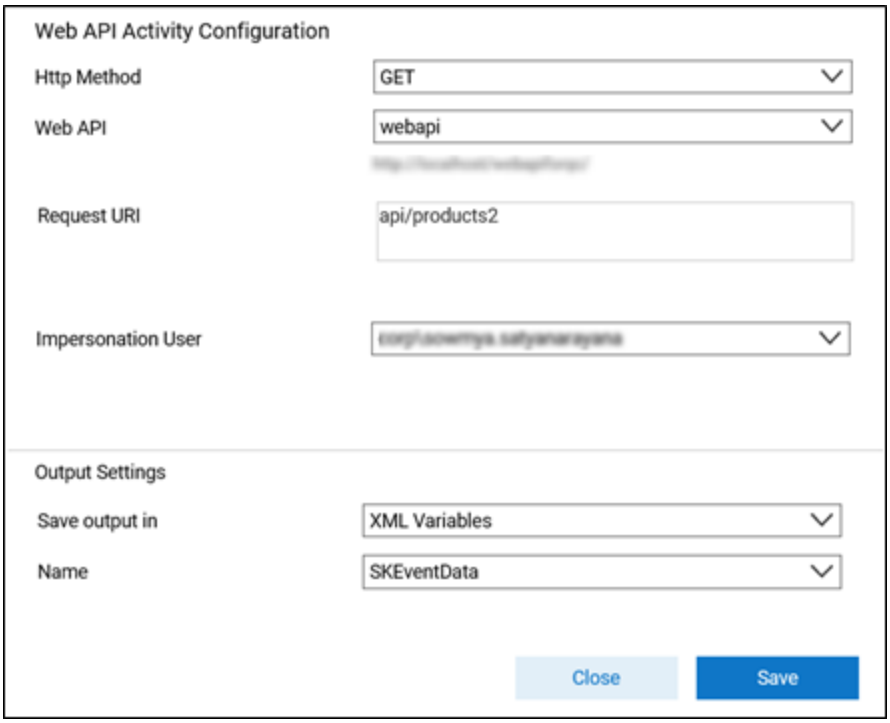
9. Configure the **Invoke Web API** activity properties.
   a. Enter a **Name** and **Description** for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



b. Click the edit icon under the **Configure Web API** property.
   The **Web API Activity Configuration** window appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



c. You have to configure the Web API to be called. Web APIs are frameworks to build HTTP services for browsers and mobile devices.

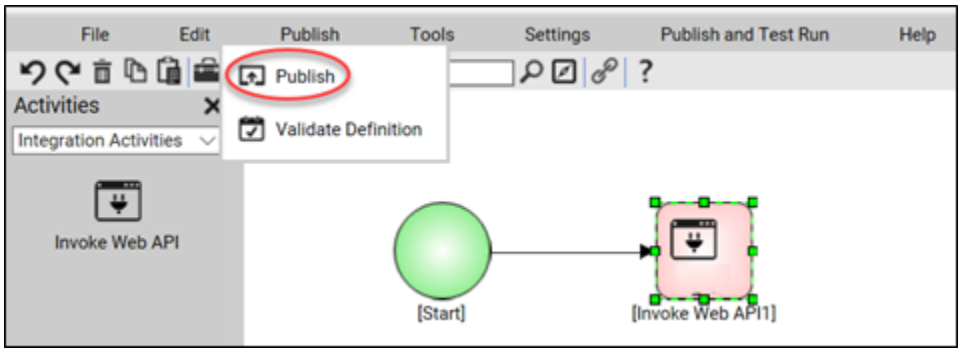
   For more information about this activity property, see Configure Web API.

10. Click **Save** in the **Web API Activity Configuration** window to save all the data entered.

    Your first Quickflow is created.

To Publish and Execute the Quickflow

1. Click **Publish** from the menu.



   A message appears to confirm that the Quickflow is published successfully.

2. Click **Close**.

   The **Process Designer** closes and the **Workflows list** appears. The list now displays one new Quickflow.

3. Right-click on the **Quickflow**, and then click **Publish and Test Run**.

   This option publishes the Quickflow and then test runs it.

   The **Process execution view** appears where you can see the execution status.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Start Activity** and the **Invoke Web API Activity** are executed successfully.

Note:
- If the Status is "Execution Pending" in the **Process execution view** window, then click the **Refresh** button to get the latest status.
- If the execution time exceeds 10 seconds, the Quickflow fails showing an error.

The Variable values are not stored in the database and hence will not be available in the Execution view or the Repository.

Note:
- In some pages, you may find that Quickflows are named as Workflows in the Enterprise Console.
- Quickflow with **Unsuccessful** output is not supported.

Save Workflow as Quickflow

You can save the published workflow as Quickflow for faster execution. The **Save as Quickflow** option is available only for the published workflows.

Note: The workflows which have the activities and properties that are supported by Quickflow can only be saved as Quickflow.

To Save the Workflow as a Quickflow

- Select and right-click the workflow you want to save as Quickflow, and then select **Save as Quickflow**.

  The selected workflow gets saved as Quickflow.

Activities

The different activities found in an AVEVA Work Tasks Quickflow are categorized as the following:

- Start Activity
- Communication Activities
- Engine Activities
- Integration Activities
- Message Queue Activities
- Logic App Activities
- AttributeReadWrite Activity
- Custom Activity

Refer to the Supported Activities table for a more comprehensive list.

Supported Activities

The table lists the supported activities and the respective supported properties:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity	Name and Description	Configuration	Quickflow Supported Configuration?	Exception Behavior	Looping Behavior
Start Activity	Yes	Variables	Yes Type - Dropdown will not have **actor** option to select	No	No
		Contents	No		
		XmlVariables	Yes Type - Xml Document and Xml Iterator options are available Xsl option is hidden		
		Analyst's Process Model Filename for Execution View	No		
		SOA Object Instances	No		
		Initialize Variable(s)	Yes		
Notification Activity	Yes	Notification Type	Yes Only Email notification available	No	No
		Actors	Yes		
		Notification Contents	Yes		
AVEVA Work Tasks Pro Push Notification activity	Yes				
Blocking Output Activity	Yes	Set Blocking Output	Yes	No	No

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity	Name and Description	Configuration	Quickflow Supported Configuration?	Exception Behavior	Looping Behavior
Database Activity	Yes	Database Connection	Yes	No	No
		Query Builder	Yes		
Decision Activity	Yes	Decisions	Yes	No	No
ForEach-Loop Activity	Yes	Xml Iterator	Yes	No	No
For-Loop Activity	Yes	Loop Variable	Yes	No	No
		Loop Count	Yes		
Invoke Web API Activity	Yes	Configure Web API	Yes	No	No
Logger Activity	Yes	LogType	Yes	No	No
		LogMessage	Yes		
Next Loop Activity	Yes	-	-	No	No
Rule Activity	Yes	Condition	Yes	No	No
Set Workflow Output Variable Activity	Yes	Set Workflow Output	Yes	No	No
Synchronize Activity	Yes	-	Yes	No	No
Update Parent Variables	Yes	Parent Workflow Application Name	Yes	No	No
		Parent Workflow Name	Yes		
		List Parent Variables	Yes		

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity	Name and Description	Configuration	Quickflow Supported Configuration?	Exception Behavior	Looping Behavior
Update Variable Activity	Yes	Set Variable	Yes Operator - <##> option is hidden Actions - Contents option is hidden	No	No
Xml Navigator Activity	Yes	Xml Iterator	Yes	No	No
		Location	Yes		
Workflow Activity		Repository	Yes - Read Only		
		Workflow	Yes		
		Version	Yes		
		Contents for ChildWorkflow	Yes		
		Variables for ChildWorkflow	Yes		
		XmlVariables for ChildWorkflow	Yes		
		Wait for the child workflow?	Yes - Disabled		
		Execution mode	No		
		Control child workflow?	No		
		Is Blocking Output Supported?	No		
Invoke Web API Activity	Yes	Configure Web API	Yes	No	No
Script Activity	Yes	Programming	Yes Language supported - C#.NET and VBScript	No	No

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity	Name and Description	Configuration	Quickflow Supported Configuration?	Exception Behavior	Looping Behavior
XML Action Activity	Yes	Get Variable	Yes	No	No
		Get Data			
		Set Variables			
		Assign Empty Node Values			
SendMQData Activity	Yes	Message Queue Connection	Yes	No	No
Call Logic App Activity	Yes	Logic App Connection and Data	Yes	No	No
AttributeReadWrite Activity	Yes	Unique Environment Name	Yes	No	No
		Relative Reference Object			
		Retry Count			

Note:
- Human activities are not supported in Quickflow application.
- As the Exception Behavior property is not supported in Quickflow, the behavior of the activity output is different from that of Workflow. Even if the activity execution results in unlinked output, the Quickflow shows it as complete.

Start Activity

Activity Description

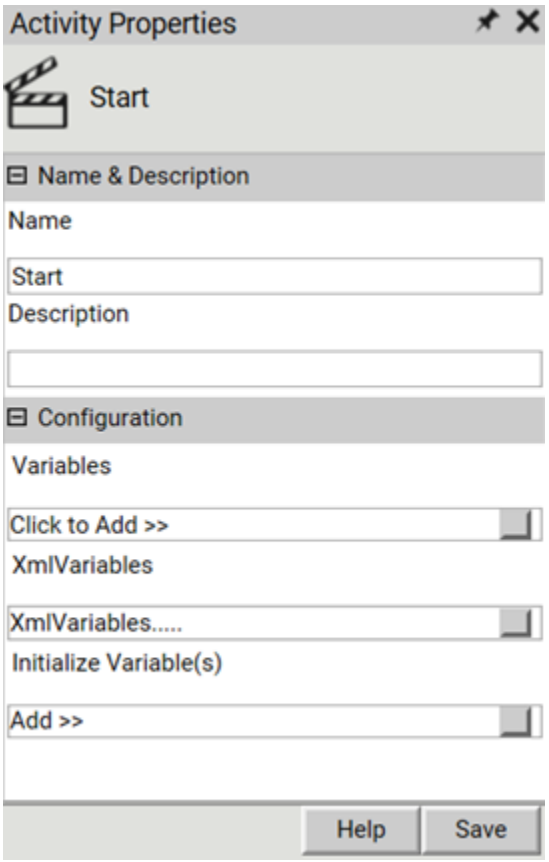
The Start activity is the first activity in a Quickflow. The icon corresponding to this activity is displayed by default in the Design Area. The Variables used in the execution of the Quickworkflow are declared here. The starting point for redirection can also be set in this activity. This will enable redirection at the beginning of the Quickflow until it is turned off later in the Quickflow.

Activity Properties

You can configure the Start activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **Start** activity and select

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the **Activity Properties** option.



Note: Every time you input a value for a property, ensure to save the data. Click the **Save** button on the **Activity Properties** window. You need to save the property so that the subsequent properties in the activity can inherit the required information.

The Activity Properties are organized in groups of related properties. The properties under each group are described in this topic.

Name & Description

You can use these properties to specify the name and description of the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

You can use these properties to specify the web service details.

Variables - The variables used in the workflow, including Local Variables, can be specified using this property.

Property Type: Optional (*Set this property only if needed*.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To set the Variable property

1. To specify the variables, click the button for the 'Variables' property to open the property window.
2. Enter the variable name in the **Workflow Variables** field. Prefix the variable name with '@' if it is a local variable.
3. Select the corresponding variable type from the **Type** drop down field.
4. After entering the variable details, click **Add Row** to add this variable and to display a new row for entering another variable.
5. Remove rows if necessary by selecting them and clicking **Remove Selected**.
6. After entering all the variables, click **Update**.

See Start - Variables for a detailed description of the Variables property window.

XML Variables - The XML Variables used in the Quickflow can be specified using this property.

Property Type: Optional (Set this property only if necessary.)

To set the XML Variables property

1. To specify the XML Variables, click the button for the 'XML Variables' property to open the XML Variables window.
2. Click **New Variable** to add a new variable.
3. This will enable the **Variable Name** field. Enter the name of the new XML Variable in this field and a description in the **Description** field.
4. Now select the XML Variable type from **Type** drop down. You can select from either **XMLDocument** or **XMLIterator**.
5. Click the **Schema** tab caption 'Schema' to specify the schema for the XML variable.
   - If XMLDocument was selected in Step 4, clicking this tab will display the corresponding set of controls.
     - You can select Variable or Content as the option for internal storage from Use [ ] for Internal Storage drop down field.
     - Specify a file name or browse for the file containing XML Data. Click the Load button to load the contents of the file. The XMLschema corresponding to the loaded file is displayed in the configuration area.
       You can also build the schema from scratch within this area.
   - If XMLIterator was selected in Step 4, clicking this tab will display the corresponding set of controls.
     - You can select the XML Variable from which you can extract the node to be set as the XML Iterator.
     - Select the node to be set from the Select Node drop down field. The XML schema corresponding to the selected node is displayed in the configuration area.
       You can also build the schema from scratch within this area.

     a. Click the Save Variable button, to save the new XML Variable.
     b. Click Close to close the Property window.

See XML Variables for a detailed description of the XML Variables window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Start - Variables

Variables Property Window

You can set the workflow variables in the Variable property window. To open this window, click the button in the 'Variables' property. The Variables dialog box is displayed in a grid, which has a blank row where you can enter the parameters for a new variable.

The dialog has two tabs at the top to manage the rows of variables in the grid.

- **Add Row** - You can click this tab to add the newly entered variable and display a new row to the grid where you can add another variable.
- **Remove Selected** - You can click this tab to remove the selected variable in the grid.

The grid has the following parameters that you need to specify for the new variable.

- **Workflow Variables** - You can use this field to enter the variable name. Prefix the variable name with '@' if it is a local variable.
- **Type** - You can use the drop-down list in this field to select the data type for the variable. The drop-down list displays the following data types - string, number, date, array, actor, object, boolean, float, and decimal.

There are two buttons and a check box at the bottom of the dialog box.

- **Update** - You can click this button to save the newly added variable(s).
- **Close** - You can click this to close the property window without saving the newly added variables.
- **Last Row needs to be added** - You need to select this check box to save the last row added.

Note: To initialize the **Date** variable, enter the date in mm/dd/yyyy format.

XML Variables

AVEVA Work Tasks uses XML variables to provide data inputs to workflows. XML variables can be configured in the Start activity through the XML Variables property. The following are the types of XML Variable in AVEVA Work Tasks:

- XML Document
- XML Iterator

This section provides further information on XML Variables and on the XML Variable User Interface.

About XML Variables

AVEVA Work Tasks uses XML variables to provide data inputs to workflows. XML variables can be configured in the Start activity through the **XMLVariables** property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Different Types of XML Variables:

The following table lists different XML variables available for a Quickflow:

XML variable	Description
XML Document	Using 'XML Document' variable, the required items can be fetched.
XML Iterator	'XML Iterator' variable allows you to define variables for repeating XML elements and attributes.

The XML Variable Interface provides the following functionality:

- Defining XML schema for the XML Variable either by specifying the complete schema or loading an XML document which is the source of the schema.
- Defining variables of type XML Document to retrieve data from XML files.
- Defining variables of type XML Iterator to retrieve data within repeating nodes within an XML schema. The XML Iterator represents a collection of XML Nodes at some level in the XML variable schema. It allows you to perform actions on XML nodes such as looping through XMLnodes and getting the value of a specific node at any location.

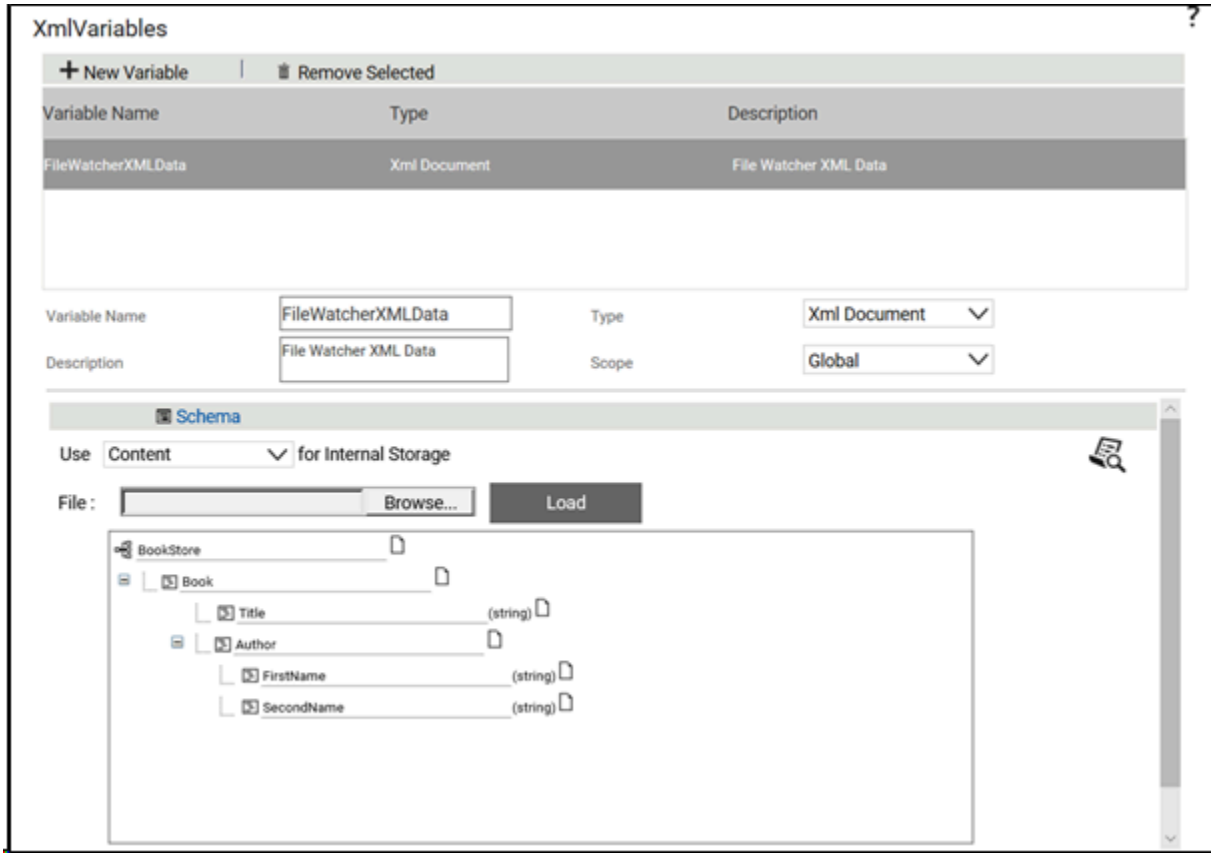
Note: Support for Special Characters in XML node Names of XML Variables: Special characters such as, < > { } * ? [ ] ( ) - _ , = | ; ` +, and % are allowed while naming XML node of XML Variables. XMLVariable will not support **XML node names** with special characters, such as \, ~, !, ', :, ", #, `, &, ., @, ^, and $.

XML Variable User Interface

The XML Variable User Interface is for configuring XML Variables. This window is opened when you click the **XMLVariables** property of the **Start** activity. All XML Variables added for the workflow are displayed in rows in the **grid**. The variable properties that are specific to a XMLVariable are displayed in the Variable Properties section for an XML Variable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

XML Variable User Interface



Common Properties Available for All XML Variables

The following properties are common for all XML Variables:

XML Variable Property		Description
Variable Name		This property is used to specify the name of the variable.
Type		This property specifies the type of the XML Variable. The available XML Variable types are XML Document and XML Iterator.
Description		This property specifies the description of the XML Variable
Scope	Global	If the XML Variable is of Global scope, then you have to set the value of this field as Global for the variable. Where that variable can be used anywhere in that workflow

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

XML Variable Property		Description
		instance.
	Local	If the XML Variable is of Local scope then the value of the variable is local to a workflow instance.

Introduction to XML Document XML Variable

An XML Variable of type **XML Document** can be used to fetch the required items.

XML Variable of type 'XML Document'

When the variable type is 'XML Document', the following properties are displayed as seen in the image below.

Property Description:

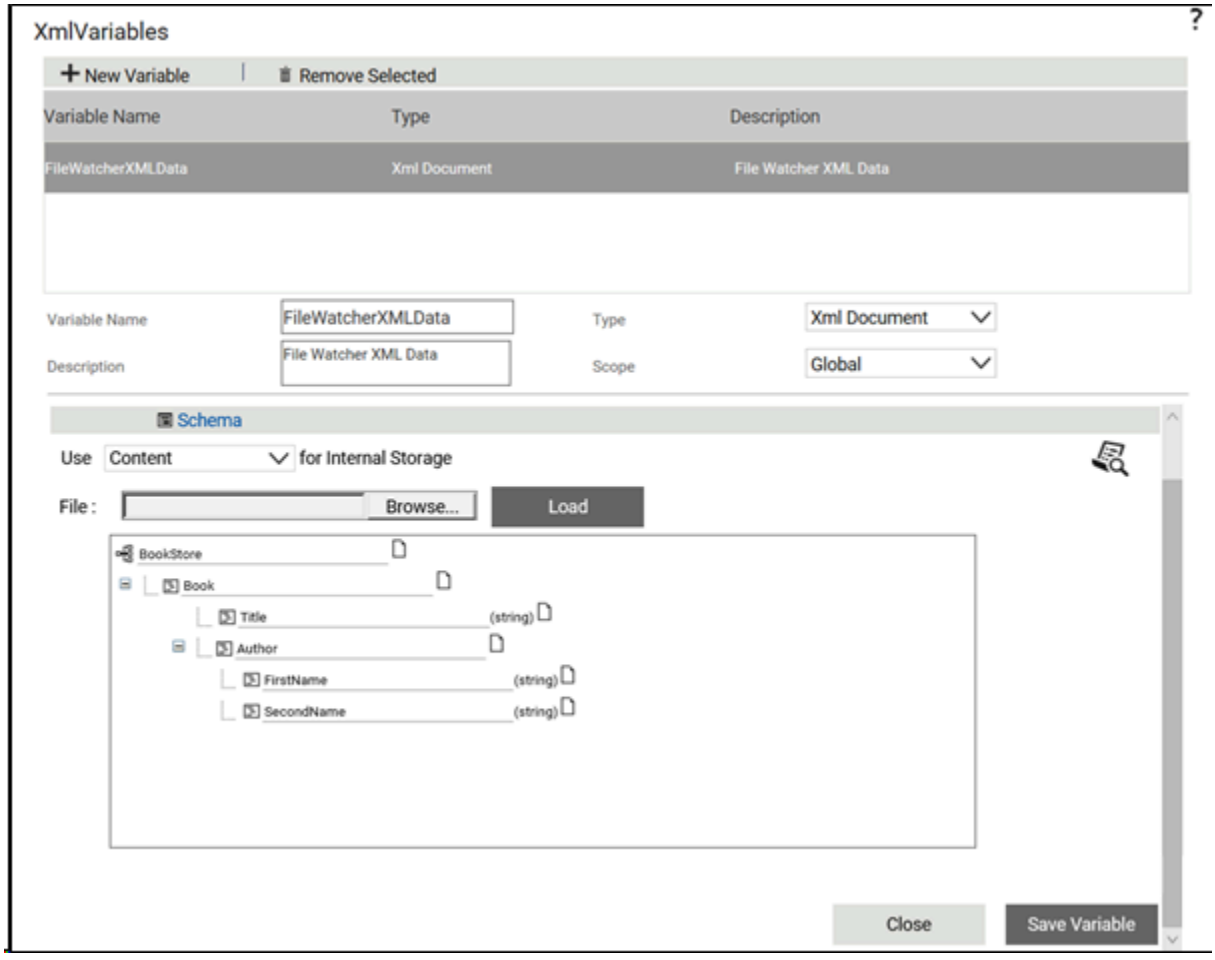
XML Variable Property	Description
Internal Storage	This property is used to specify the internal storage type of the XML Variable. • If the selection is Content, then XML Variable value will be stored as content in workflow context. You cannot update values for such XML variables, since the content data cannot be changed during workflow execution. • If the selection is Variable, then the same will be stored as a variable. You can update the values for such XML variables using "Update Variable" activity under engine activities. **Note:** The default storage option of XML variable is Variable.
File	Location of the schema file, if any XML schema is available. Otherwise user can create the schema using Schema designer.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create XML Variable of Type 'XML Document'

To add a XML Document Variable

1. Click **New Variable** button in the XML Variable User Interface.
2. Enter the following information:
   - **Variable Name**: Enter the name of the variable.
   - **Type**: Select XML Document from the drop-down list. This field defaults to XML Document. The XML Iterator type option is available only if another variable of XML Document type exists.
   - **Description**:- Enter a description for the variable.
   - **Scope**: Select the scope of the variable. The scope of the variable can be either Global or Local. Select Local, if the value of the variable is local to a workflow instance.
3. **Schema**: Build the XML Schema. You can either build the schema or use the existing XML file to create the schema.
4. Specify how the variable is to be used in Use for Internal Storage field. The field defaults to Content. To store as variable, select Variable in this field. If the scope of the variable is **Global**, you must set this field to **Variable**.
5. Click **Save Variable** to save the new variable.
6. Click **Close** to save the node properties.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

See Also: Build XML Schema

Build XML Schema

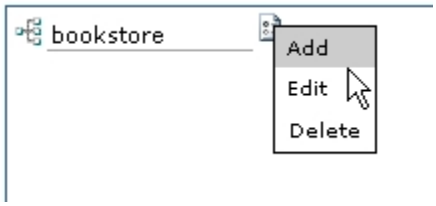
You can build a XML Schema in two ways:

1. Define schema in the XML Variable interface
   a. Adding a Node
   b. Editing a Node
   c. Deleting a Node
2. Loading schema from an existing file

Define Schema in the XML Variable Interface

When a new variable is created, the schema displays the root node by default. Click the string "root"and type the name over it.

The following image shows the node options:



To add a node

1. To add a node, click the Properties icon to the right of the name. Select **Add**. The **Properties** window for the new node appears.
2. Specify the following properties of the node:
   - **Name**: Enter a name of the node.
   - **Type**: Select the node type. The options available are **element** and **attribute**.
   - **Data Type**: Select the data type. This property specifies the data type of the XML Node. The available data types are string, integer, long, double, decimal, date, time, dateTime, boolean, and base64binary.
   - **Repeating**: Select this check box, if the node is a repeating one.
3. Click **OK**.
4. Click **Save Variable** to save node properties.
5. Click **Close**.
6. To add more nodes, repeat step 1 to step 6.

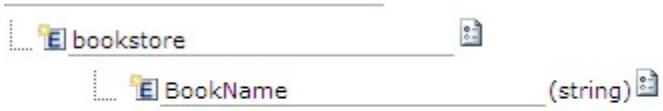
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: Select the respective node where you want add a node. If a node is of Attribute type, then you cannot add a node under it. The Add option will be disabled for Attribute type of node.

To edit an existing node

1. To edit an existing node, click the **Properties** icon to the right of the node to be edited.
2. Select **Edit**. The Properties window for the existing node appears.

3. Modify properties in the Properties window.
4. Click **OK**.

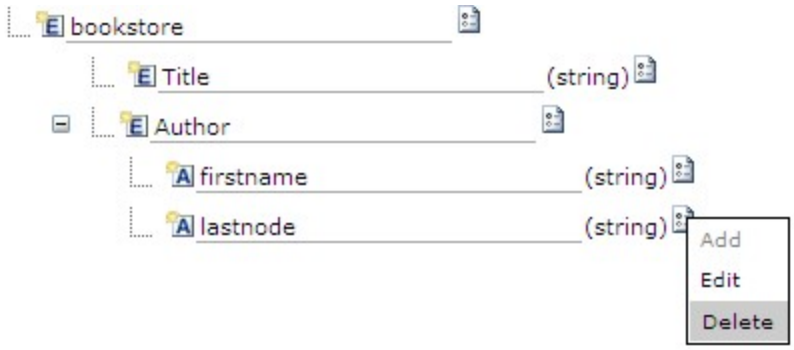


5. Click **Save Variable** to save node properties.
6. Click **Close**.

To delete a node

1. To delete a node, click the **Properties** icon to the right of the node to be deleted.
2. Select **Delete**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



To load Schema from an Existing File

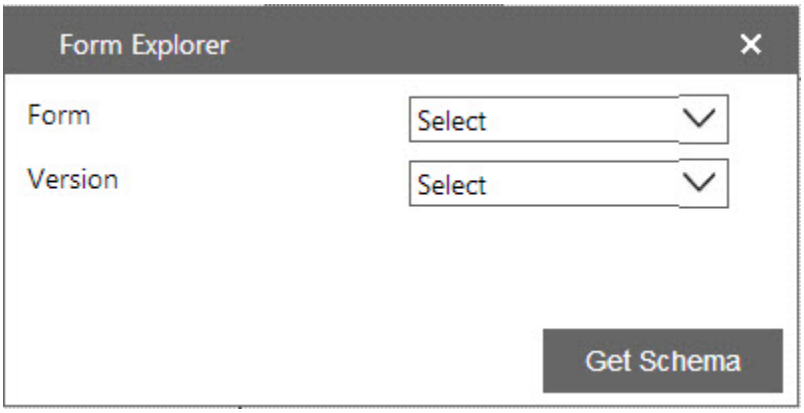
1. Click **Browse** to select file with schema information. You can select XML files to load schema.

2. The XML schema is loaded from the file.
3. To edit a node, right-click the node, and then select **Edit**. Modify properties in the Properties window.
4. To delete a node, right-click the node, and then select **Delete**

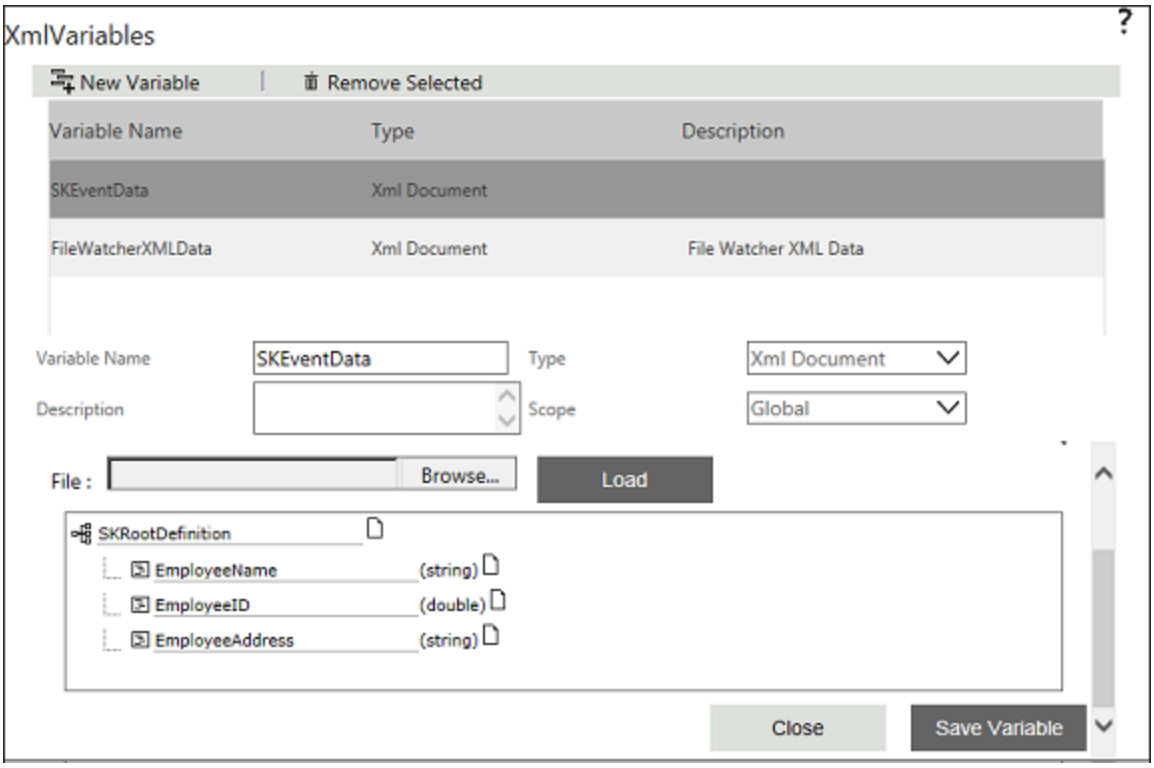
Determine XML Variable Schema from Forms

After a form is associated to a particular workflow, the form creates the XML Variable named **SFFormData** of type **XML Document** in the **Start** activity of that workflow. This will have the schema of the form.

AVEVA Work Tasks provides the **Form Explorer** icon to choose the form and its version. This icon will get displayed only for XML Variable of type XMLDocument. Any XML Variable of type XML Document can now get its schema from any form and its version available in the **Manage Forms** List.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Before introducing, when some changes were done in the form, the XML Variable had to be deleted and the form had to be re-associated in order to get the updated list of controls.

Refer to the *Lists and Forms Guide* for more information on Lists and Forms.

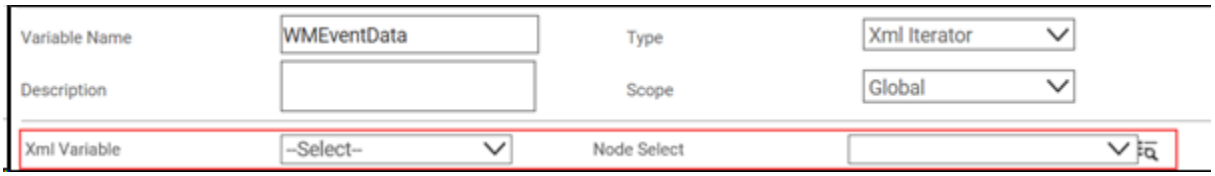
Introduction to 'XML Iterator' XML Variable

XML Iterator Variables allows you to define variables for repeating XML elements and attributes. This variable type represents a collection of XML nodes at any level in the XML Variable schema. It allows you to perform actions on XML nodes such as looping through XML nodes and deriving the value of a specific node at any location.

Note: The XML Iterator type option is available only if another variable of XML Document type exists

XML Variable of type 'XML Iterator'

When the variable type is changed to **XML Iterator**, following extra properties are displayed as seen in the image below.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Description

'XML Iterator' XML Variable Property	Description
XML Variable	All variables of type XML Document and XML Iterator are listed for selection. Select the parent XML Variable. The parent variable of an XML Iterator variable can be another XML Iterator Variable.
Node Select	Only nodes in the parent variable are displayed in this drop-down. Select the node that the variable represents.

See Create XML Variable of Type XML Iterator for more information.

Create XML Variable of Type 'XML Iterator'

To create XML Variable of type XML Iterator Variable

- Click **New Variable** button in the XML Variable Interface.
- Enter the following information:

  Variable Name - Enter the name of the variable.

  Description - Enter a description for the variable.

  Type - This field defaults to XML Document. Select XML Iterator from the drop-down. The XML Iterator type option is available only if another variable of XML Document type exists.

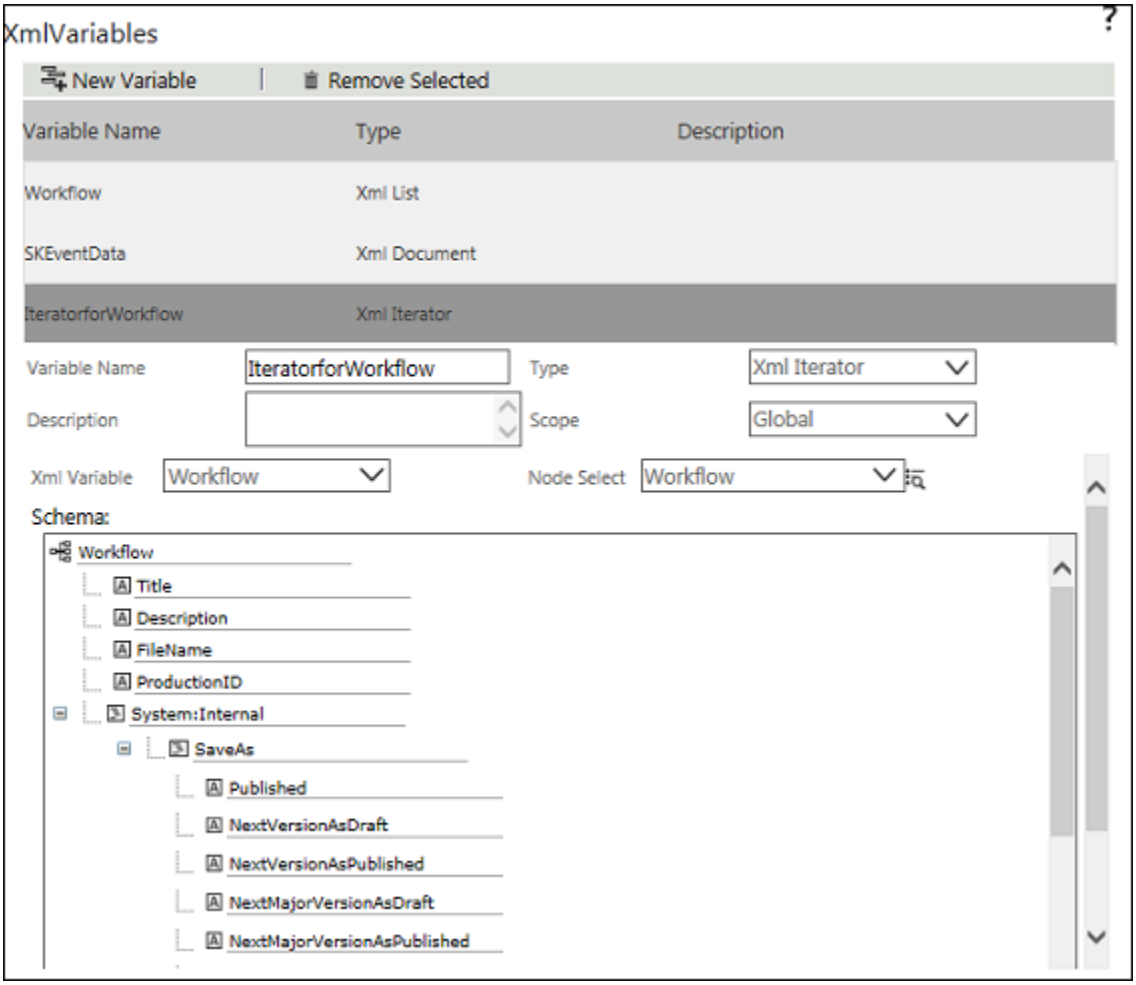
  Scope - Select the scope of the variable. The scope of the variable can be either Global or Local.
- When the variable type is changed to **XML Iterator**, two additional properties are displayed as seen in the image below.

  XML Variable - Select the parent XML Variable. All variables of type XML Document and XML Iterator are listed for selection. The parent variable of an XML Iterator variable can be another XML Iterator Variable.

  Select Node - Select the node that the variable represents. Only the nodes in the parent variable are displayed.
- Click **Save Variable**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

XML Iterator Variable Properties



Use 'XML Iterator' Type Variables

XML Iterator Variables can be used in any of the following ways:

- **Building conditions for setting Activity Properties** - XML Iterator Variables can be used like an XML Variable to build conditions for setting Activity Properties. However, when you select the XML Variable of XML Iterator type, only the nodes under the selected node of the parent variable are shown.
- **In ForEach-Loop activity** - XML Iterator variables can be used in the ForEach-Loop activity to loop through an activity till it executes for the entire collection in the XMLIterator variable. Example of XML Iterator Variable in ForEach-Loop.
- **Navigate to a specific node in the collection** - XML Iterator variables can be used in the XML Navigator Activity to go to a specific node in the collection and then perform an action. Example of XML Navigator Activity.

Example of XML Iterator Variable in ForEach-Loop

In this example, the XML Iterator Variable is used to get approval from all managers in the Managers.XML file

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

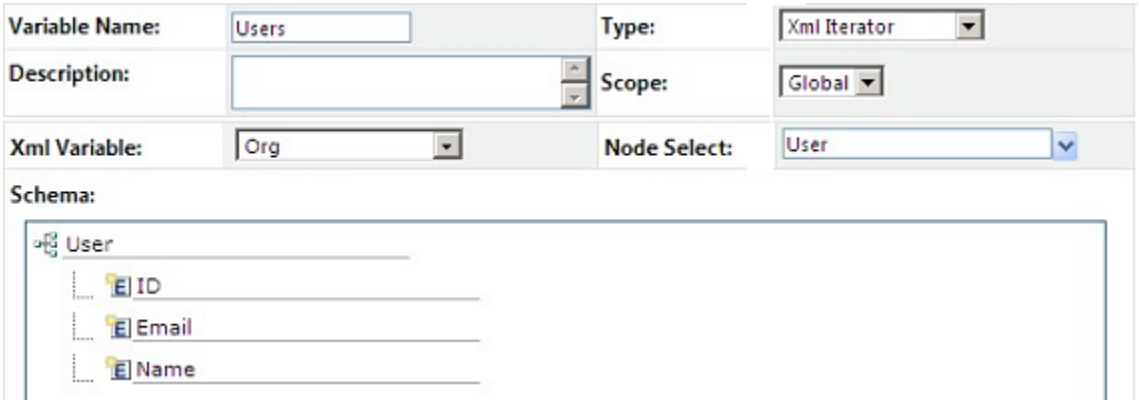
containing the list of all the users who are managers.

Managers.XML file

```
<Managers>
<Users>
<User id=1>
<Name>John</name>
</User>
<User id=2>
<Name>Jim</name>
</User>
<User id=3>
<Name>Victor</name>
</User>
<User id=4>
<Name>Andy</name>
</User>
</Users>
</ Managers >
```

Step 1: XML Iterator Variable is defined for users. The node Users holds the collection of managers.

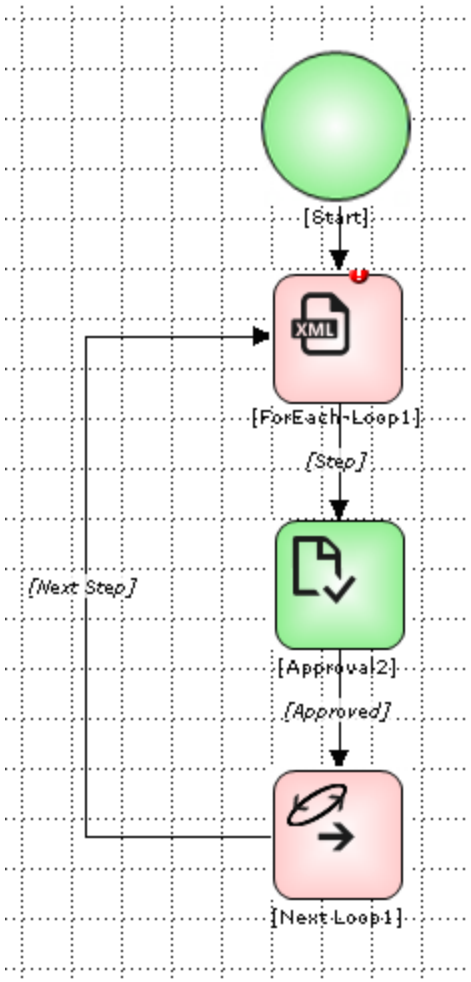
XML Iterator Variable for Managerial Users



Step 2: The approval activity is set within a ForEach-Loop activity.

ForEach-Loop Activity for Looping Approvals

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

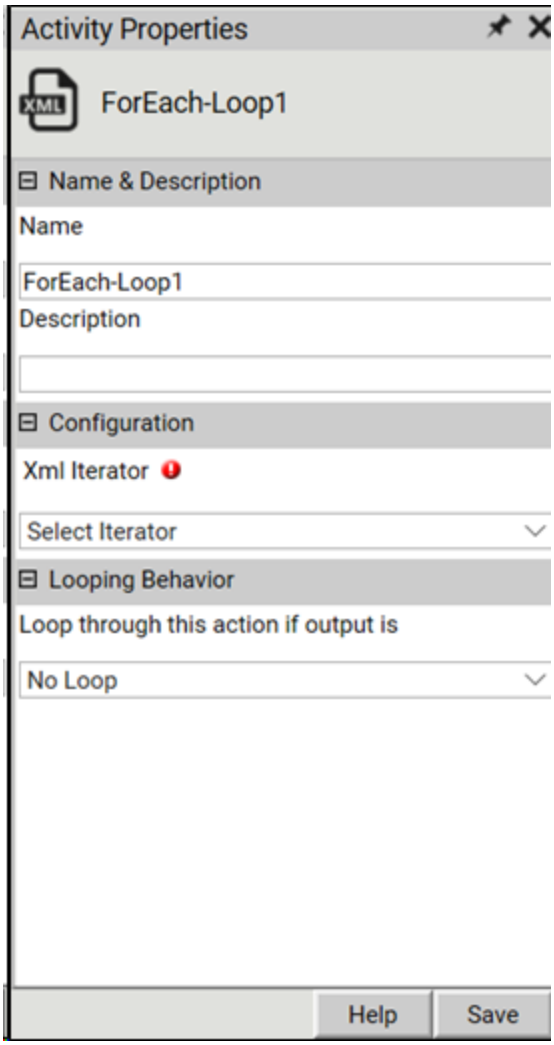


Step 3: The XML Iterator Variable 'Users' is defined in XML Iterator field of ForEach-Loop properties. The Execution type is set to Serial as the approval is to be processed serially.

Note: The Execution type is set to Parallel as the approval is to be processed simultaneously by all approvers. When the Execution type is Parallel, the Next Loop activity is not required.

ForEach-Loop Property for XML Iterator

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



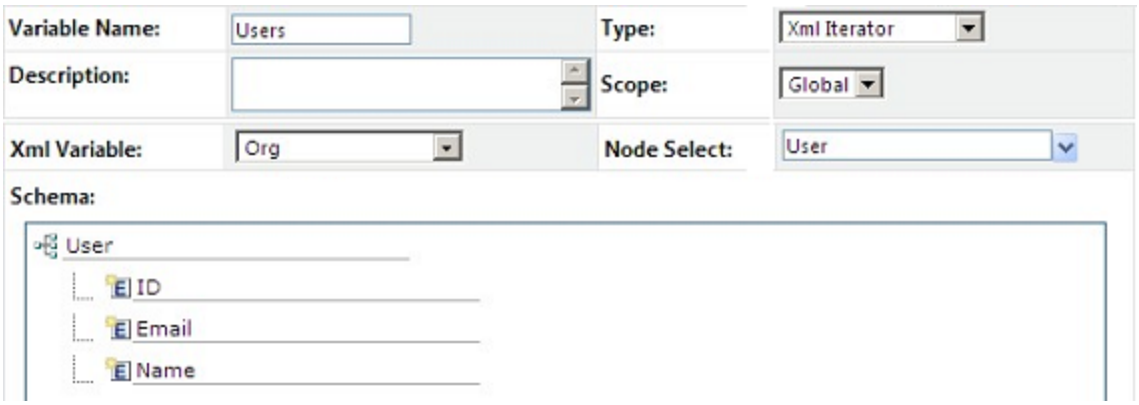
Example of XML Navigator Activity

A company allocates new laptops to employees as and when new inventories are received based on date of requisition. Allotments are to be approved by the manager of the requisitions user. Users with requisitions are added to a waiting list by the Facilities Management System. Allotments are made to the user at the top of the list. When the allotment is complete, the user is removed from the list. If a manager rejects the allotment, the allotment is made to the next person in the waiting list. This example demonstrates how the XML Navigator Activity is used to automate a process.

Step 1: A global XML Iterator variable is defined for the node users.

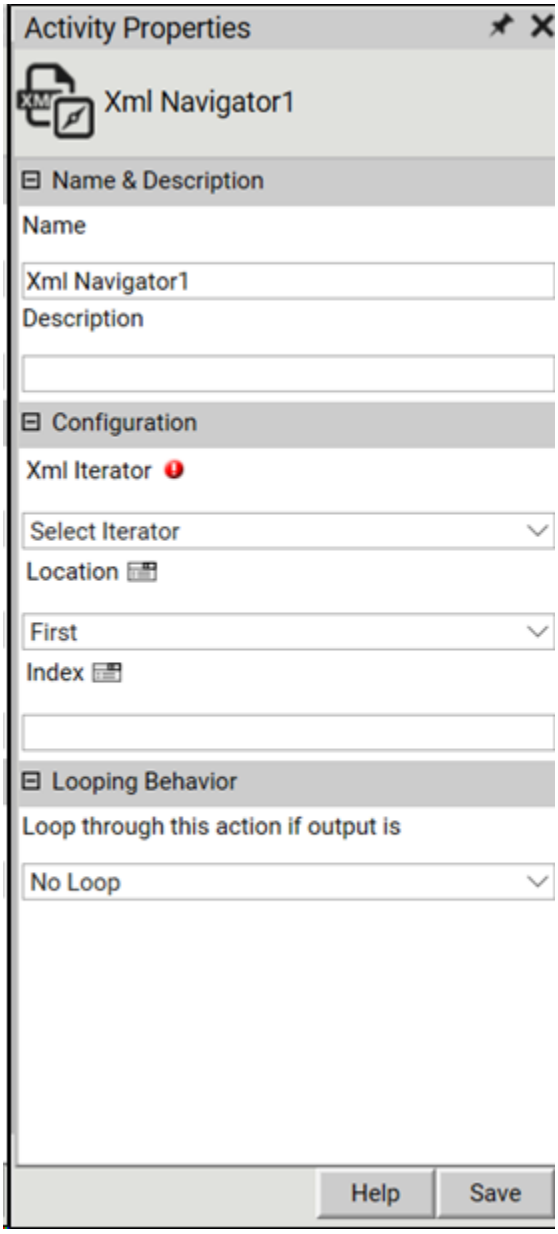
XML Iterator Variable for Users

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Step 2: The XML Navigator activity is added to the workflow with **Location** set to First.

XML Navigator Properties

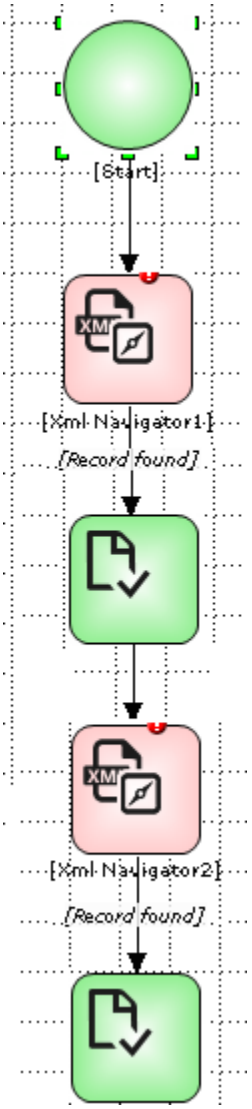


© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Step 3: An Approval activity is added after the XML Navigator Activity that sends an approval task to the manager of the user at the top of the list. The XMLNavigator links to Approval only if a record is found in the waiting list.

Step 4: The Approval task is linked to another XML Navigator Activity with the property set to **Next**.

Step 5: The following figure depicts the workflow after adding these two properties.

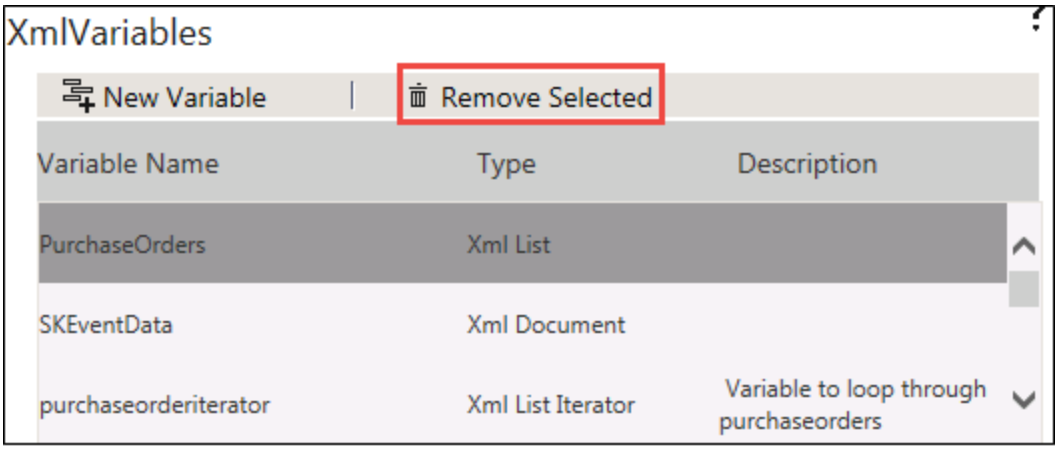


Delete XML Variables

To delete an XML Variable

1. Select the variables to be removed.
2. Click **Remove Selected** in the **XMLVariables** window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Note: You cannot delete a variable that is the parent of another **XML Iterator** type variable

Communication Activities

There are two main types of Communication activities:

- Notification Activity: The Notification Activity is incorporated in the workflow process design to send notifications to actors in the form of email.
- AVEVA Work Tasks Pro Push Notification Activity: The AVEVA Work Tasks Pro Push Notification activity is used to send notifications to the AVEVA Work Tasks Pro application.

Notification Activity

Activity description

The Notification Activity is incorporated in the workflow process design to send notifications to actors in the form of email. The messages can be customized for each actor by appending the resource properties to the message.

Activity Properties

The Notification activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Notification activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Notification Type - The supported Notification type is "Email".

Property Type: Mandatory (This property must be set if the action is to be assigned to any resource(s).)

Actors

The property in the Actors group has been described below. You can use this property to specify the content of the notification.

ToResource- This property specifies the resource(s) to which the notification is to be sent.

Property Type: Mandatory (This property must be set if the action is to be assigned to any resource(s).)

Steps to set the To property

1. To specify the actors to receive the Notification message, click the button for the 'To' property to open the property window.
2. In this window, build an expression to select the actors using the actors properties. Select an attribute from the first drop down list, then select the operator, type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the corresponding drop downs. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression to the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Notification - ToResource for a detailed description of the To property window.

See the Using XML Variables section in the About XML Variables topic for details on how to use the XMLVariables to build the expression.

Notification Contents

The properties in the Notification Contents group have been described below. You can use these properties to specify the content of the notification.

From Email Address - This property specifies the email address from which the email notifications will be sent. If the Notification type is of the form 'Email' then this property must be set. If it is not set, then the email notifications will not be sent.

Property Type: Optional (This property needs to be set only if the notification type selected in the corresponding property is 'Email'.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Message - This is the actual content of the notification sent to the resource(s). The notification can be customized for each resource.

Property Type: Optional (This property needs to be set only if required.)

Steps to set the Message property

1. To specify the Notification message, click the button for the 'Message' property to open the property window.
2. In this window, enter the message text directly in the text editor.
3. If you want to personalize the message you can do so using the **Field** controls to add resource property values. For example, to add a message that includes the name of the actor submitting the approval request, you can do as follows: First type in the text "Hello". Next select "Variable" and "Variable.SubmittedBy.name" from the two drop downs for **Field** and click **Append**. This will add the placeholder for the variable after the text entered in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the name, resulting in a personalized salutation. Type in a message in the next line to complete the message body.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, repeat the last command, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the message, click **Update** to save it.
7. Click the **Close** button to close the window without saving the message.

See Notification: ToResource for a detailed description of the Message property window.

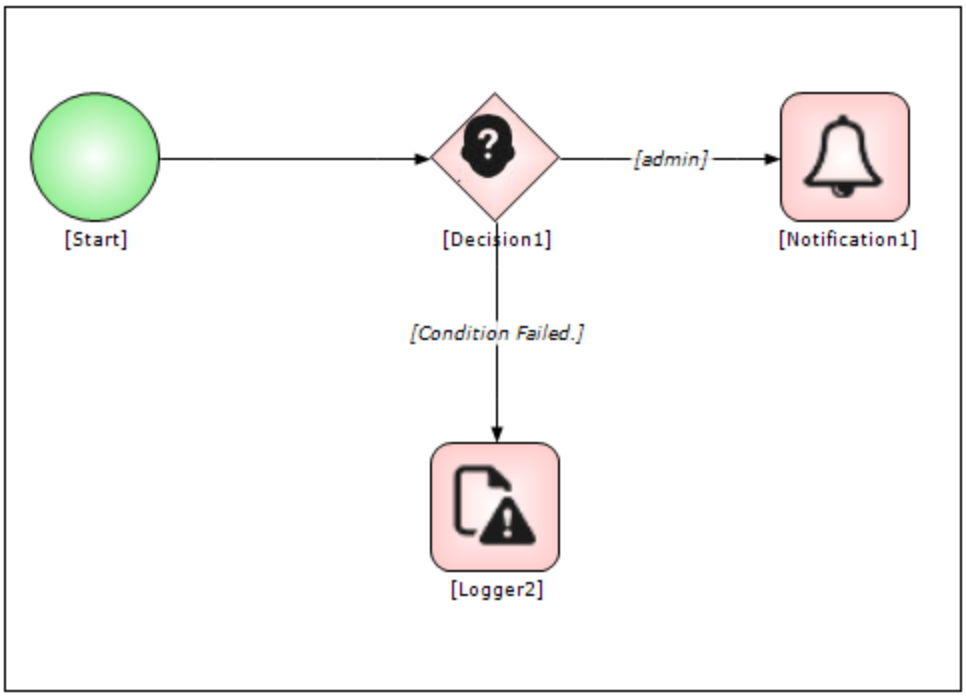
Activity outputs

The Notification Activity does not return any value.

Example for Notification Activity

In an organization, whenever Decision1 condition is successful, a Notification is sent to the Admin. If the Decision1 condition fails, it will be logged into the Logger2, as shown in below workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



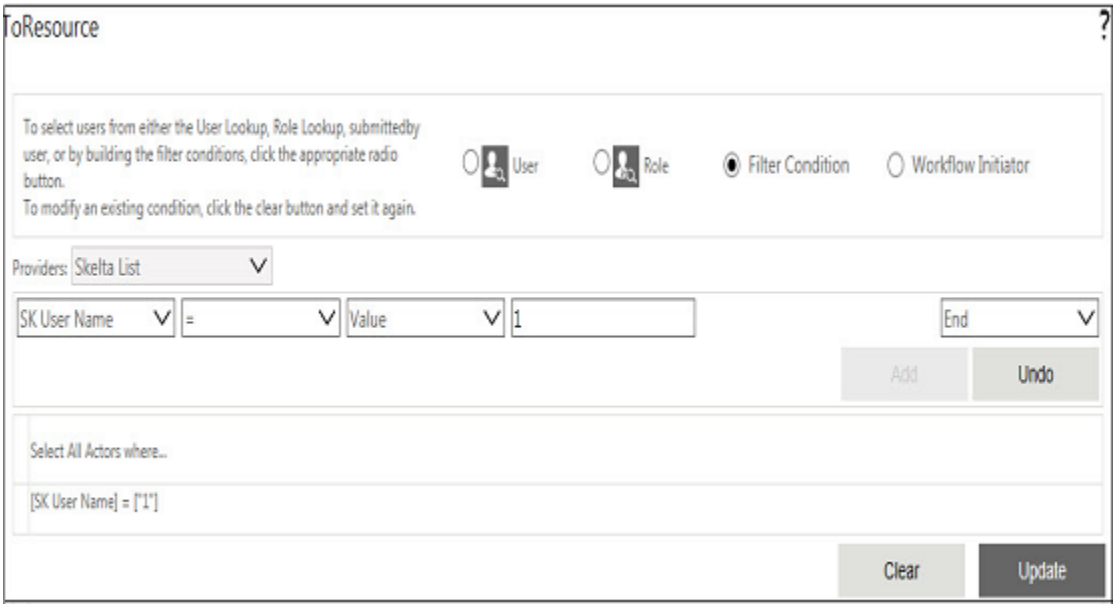
Whenever a workflow triggers a Decision, it will be logged in Logger2 if it fails. If Decision1 is successful, a Notification will be send to the Assign actor.

Settings For Notification Activity:

Notifications can be sent in the form of **Email** to cell phones.

Step 1: Select **Notification Type** as **Email** type to send notification.

Step 2: Click **ToResource** property to specify the resource(s) to which the notification has to be sent.



Step 3: Select specified **Actor** and click **Update** to save the changes.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Notification: ToResource

You can set the actors to whom the Notification is to be sent in this property window. To open this window, click the button for the 'To' property.

The Assign Actor window provides the flexibility of assigning actors either by building an expression or through User lookup, Role lookup or Workflow Initiator. This window has a set of radio buttons and drop downs to select the actor and different parameters required to build an expression to identify the actor(s). You can also find related buttons next to the drop downs.

Different Methods of Assigning an Actor

- **Assigning an Actor by building an expression**:

  Filter Condition is the default option that is selected when the Assign Actor window is accessed. This option allows you to build an expression to identify the actor. The section below explains how the set of drop down buttons helps you choose different parameters required to build and expression.
- The Providers drop-down- You can select the required provider from the Providers drop-down.
- The Attribute drop down - You can select the attribute to be used to identify the actor, e.g., "AdministratorId", in this drop down.
- The Operator drop down - You can select the operator for the expression here. E.g., "="
- The Value type drop down - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.

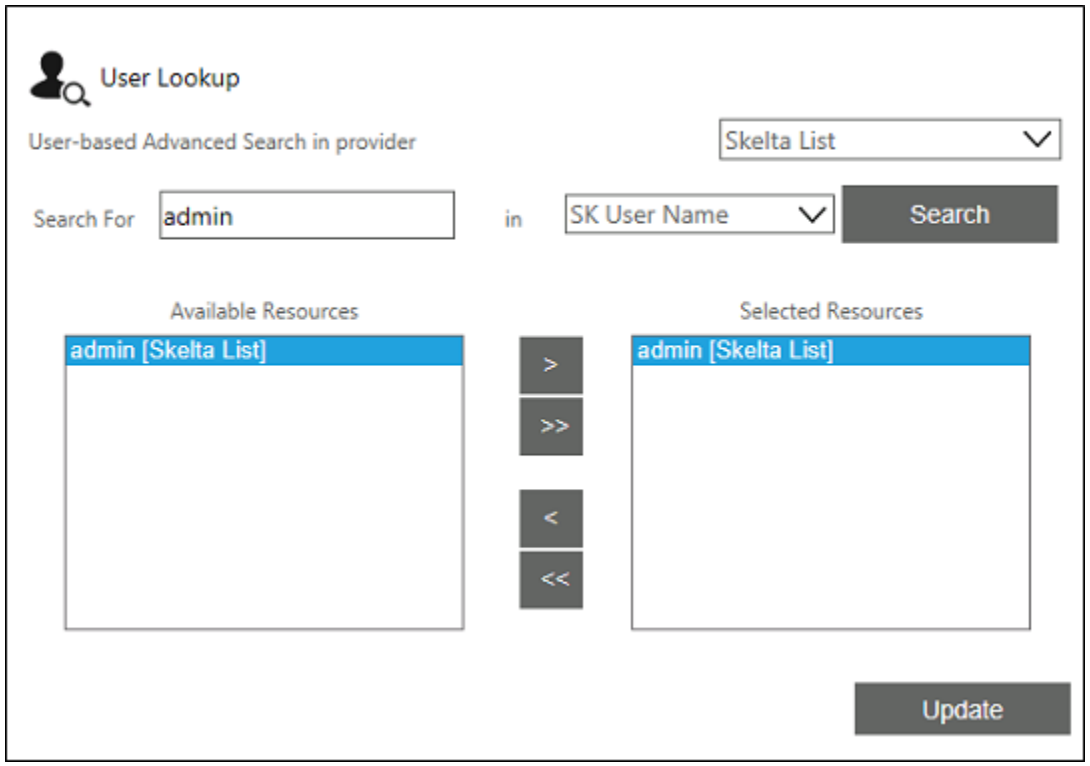
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

- **Assigning an Actor through User Lookup**
  Through the **User Lookup** you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through User Lookup
  - Select the **User Lookup** option.
  - The **User Lookup** dialog is displayed, search and select the required actor.

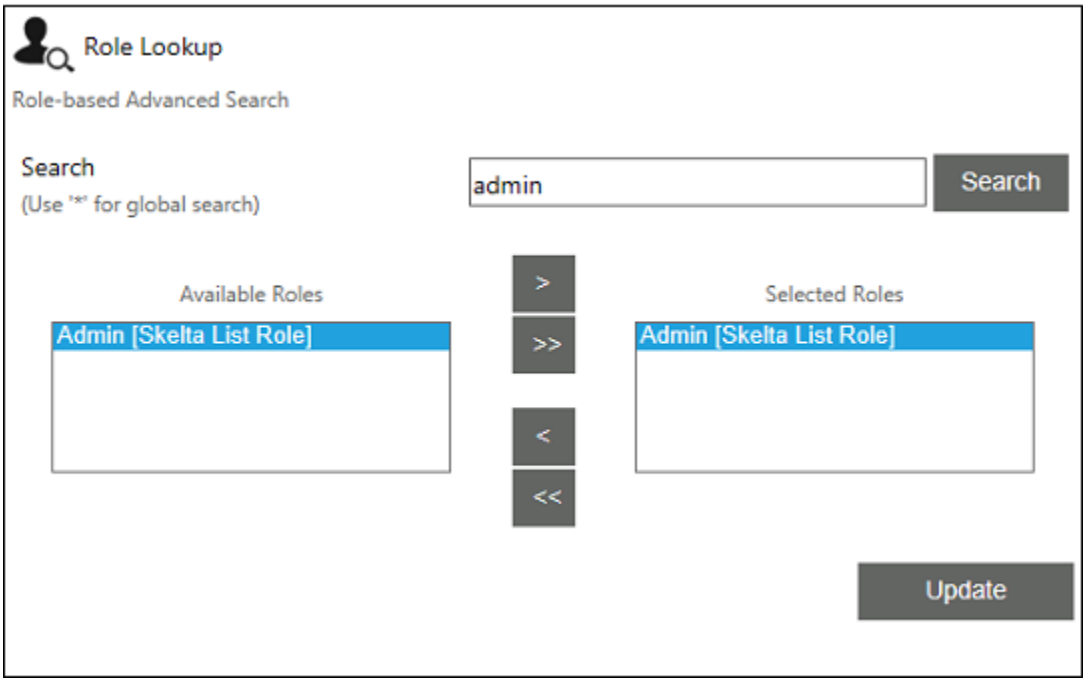


  - Click **Update** in the **User Lookup** dialog and verify the Assign Actor window.
    In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Click **Update** in the Assign Actor(s) window to update the selected actor(s) in **Assign Actor** property. Actor(s) selected through User Lookup will be updated in the Assign Actor property.

- **Assigning an Actor through Role Lookup**
  Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an actor through Role Lookup
  - Select the **Role Lookup** option.
  - The **Role Lookup** dialog is displayed, search and select for the required role.

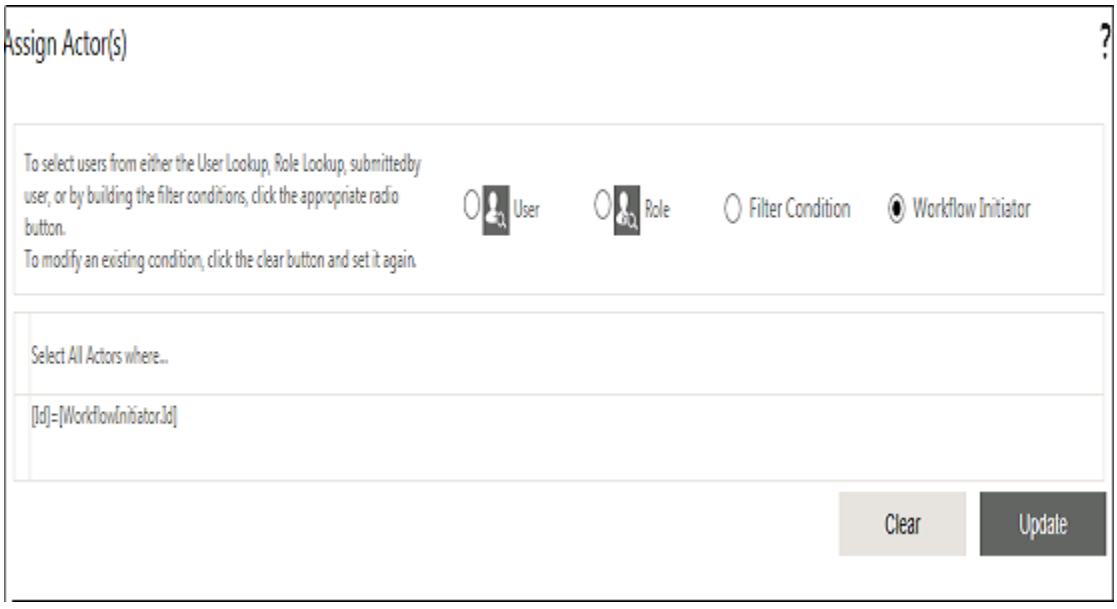


  - Click **Update** in the **Role Lookup** dialog and verify the Assign Actor window.
    In the **Assign Actor** window, you will see that an expression is displayed, refer the image below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Click **Update** in the Assign Actor(s) window to update the Assign Actor property.

- **Assigning an Actor through Workflow Initiator**
  Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Assign Actor window, refer the image below:

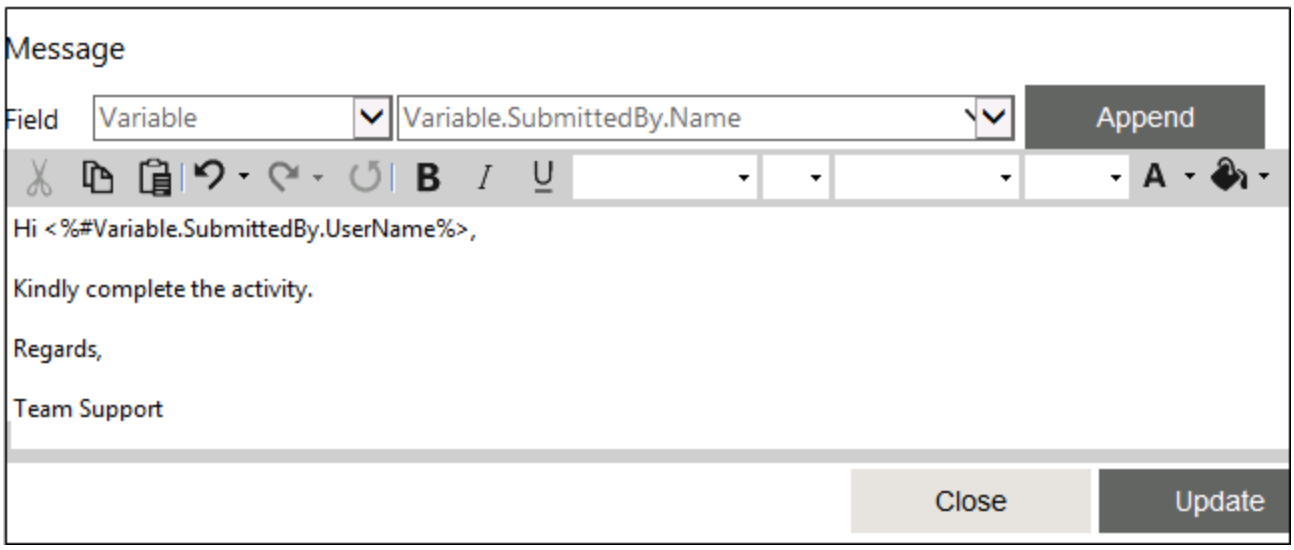


Notification: Message

Message Property window

You can specify a message for the Notification in this property window. To open this window, click the button for the 'Message' property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The window has a text editor to enter the Notification message text. You can do this either by entering a discrete value directly in the text editor or by using the **Fi**eld controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/drop down - You can specify the actual value of the message. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the message text in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Notification message.
- **Append** - You can click this button to add the specified value, variable or content to the text editor.

Tip: You can specify the Notification message using a combination of discrete value and variable or content. This helps in personalizing the message for each actor. For example, you can specify the salutation line in the message as "Hello <%#Variable.SubmittedBy.name%>,". Here, the string "Hello" must be typed directly in the text editor. The second half of the salutation is a placeholder, which will be replaced by the variable value during run-time. This must be added by selecting the appropriate variable option from the **Field** drop down and clicking **Append**. The rest of the message can be typed in the editor. This will result in the same message going to different actors with their respective names in the salutation line.

The text tool bar above the text editor allows you to manage and format the Notification message text. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the message
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window, you will find the following buttons:

- **Update** - You can click this button to save the message.
- **Close** - You can click this button to close this window.

AVEVA Work Tasks Pro Push Notification Activity

Activity Description

The AVEVA Work Tasks Pro Push Notification activity is used to send notifications to the AVEVA Work Tasks Pro application. When this activity is added and configured in a workflow along with other activities which need human interaction, a notification is sent to the actor intimating the actor that action is required on the assigned task.

Activity Properties

Name and Description

The properties in the Name & Description are used to specify the name and description for the activity.

Name: This property is used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description: This property is used to specify a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group allows you to configure the functionality of the activity.

Recipient Resource: This property is used to configure the resources who have to receive the notification. A maximum of 16 resources can be configured as recipients to receive notification. If more than 16 resources are configured, then the 16 active resources will receive the notifications.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

You can also build an expression to configure the recipient, this expression will be evaluated at runtime and the recipient will be configured.

Property Type: Mandatory (This property needs be set to add the recipients.)

For more information on Recipient Resource, see Configure Recipient Resource.

Notification Type: This property allows you to configure custom notification type based on the device and operating system used.

Property Type: Mandatory (This property needs to be set to configure the type of notification specific to a device.)

For more information on Notification Type Setting, see Setting Notification Type.

Message: This property allows you to configure the content of the notification message.

You can also build an expression to configure the notification message. The message will be updated dynamically at runtime.

Property Type: Mandatory (This property must be set to configure the content of the notification.)

Retry on Failure?: When this property is set to **Yes** and the notification is not sent, attempts to re-send the notification will be made based on the value defined in the Retry Count property.

By default it is set to **Yes**.

Property Type: Optional (This property needs to be set only if necessary.)

Retry Count: Number of re-tries that needs to be attempted to send the notification. The value specified for this property should be from 1 to 10 only.

Property Type: Optional (This property needs to be set only if necessary.)

Activity output is successful: This property lets you decide when the activity is successful. The available options are **If notification is sent to at least one device** and **If notification is sent to all devices**.

- When **If notification is sent to at least one device** is selected, the activity will be successful only when the notification is sent to at least one device.
- When **If notification is sent to all devices** is selected, the activity will be successful only when the notification is sent to all the devices

Reference Topics

- Set Notification Type
- Mapped Error Outputs
- AVEVA Work Tasks Pro Push Notification Activity Scenarios

Configure Recipient Resource

The **Recipient Resource** property allows you to configure the users who have to receive the notification. A maximum of 16 resources can be configured as recipients to receive notification. If more than 16 resources are configured, then the 16 active resources will receive the notifications.

Different Methods of Assigning an Recipient Resource

Assigning an Recipient by building an expression:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Filter Condition is the default option that is selected when the Recipient Resource is accessed. This option allows you to build an expression to identify the Recipient. The section below explains how you can choose different parameters required to build and expression.

- The Providers list - You can select the required provider from the Providers list.
- The Attribute list - You can select the attribute to be used to identify the Recipient. For example, "AdministratorId", in this list.
- The Operator list - You can select the operator for the expression here. E.g., "="
- The Value type list - You can select the type of value to assign to the attribute in the expression. You can specify Variable, Content, Value, or XMLVariables here.
- The Value text box/list - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding list.
  - If you had selected "Value", it will display a text box here. You can enter a discrete value in this box to complete the expression.
  - If you had selected "Variable", "Content", or "XMLVariables", there will be a list instead of the field. This list will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator list - You can select logical operators from this list to build compound expressions. The options available are "End", "And", "Or", "')' And", and "')' Or". You can select "End" to complete the expression.
- **Add** - You can click this button to add the new expression to the text area.
- **Undo** - You can click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

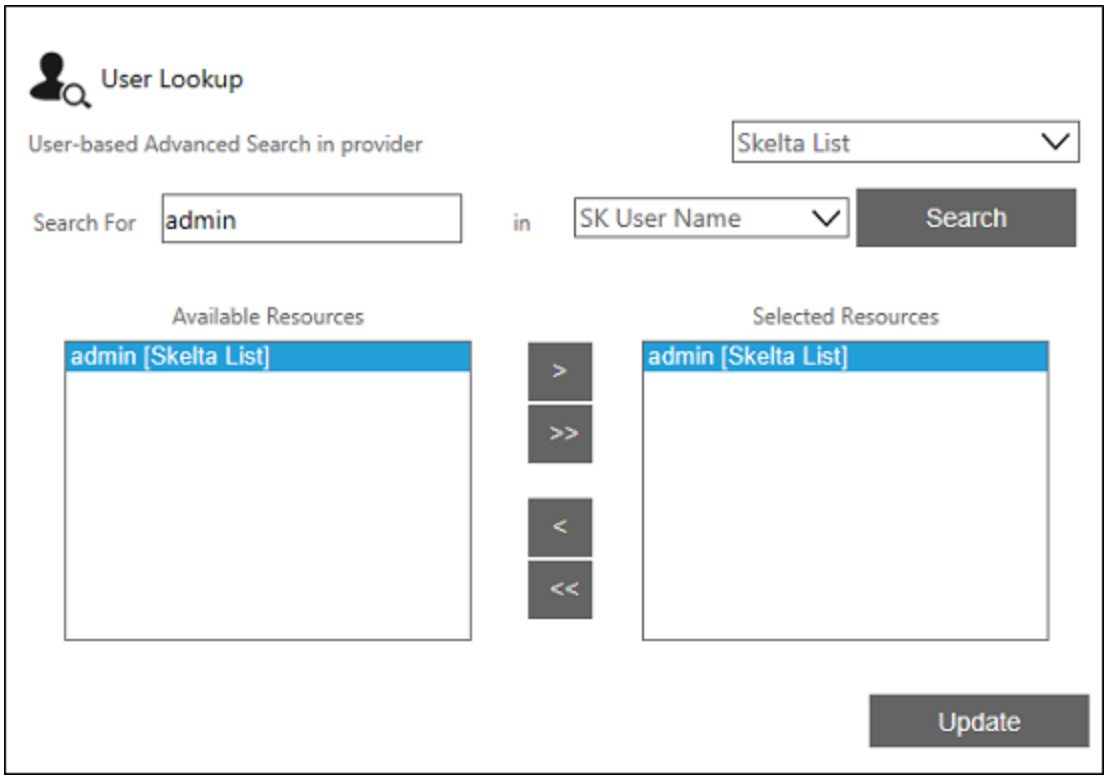
- **Clear** - You can click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - You can click this button to save the expression that has been added to the text area.

To assign an Recipient Resource through User Lookup

Through the **User Lookup** you can search and select user(s). This automatically builds the expression and the same is updated in the text area. Follow the steps given below to assign an Recipient Resource through User Lookup

1. Select the **User Lookup** option. The **User Lookup** dialog is displayed.
2. Search and select the required user.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



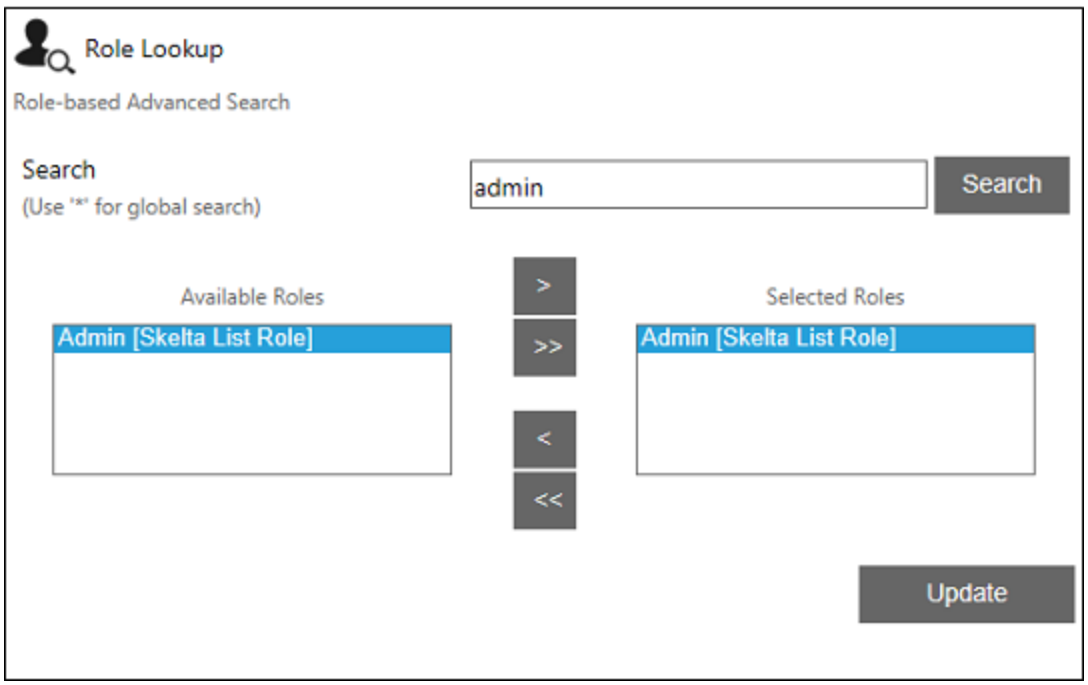
3. Click **Update** in the **User Lookup** dialog. In **Recipient Resource**, you will see that an expression is displayed.
4. Click **Update** in **Recipient Resource** to update the selected user in the **Recipient Resource** property.

To assign an Recipient Resource through Role Lookup

Through the **Role Lookup** you can search and select a specific role. This automatically builds the expression and the same is updated in the text area.

1. Select the **Role Lookup** option. The **Role Lookup** dialog is displayed.
2. Search and select for the required role.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click **Update** in the **Role Lookup** dialog and verify the Recipient Resource. In **Recipient Resource**, you will see that an expression is displayed.
4. Click **Update** in **Recipient Resource** to update the **Recipient Resource** property.

Assigning an Recipient Resource through Workflow Initiator

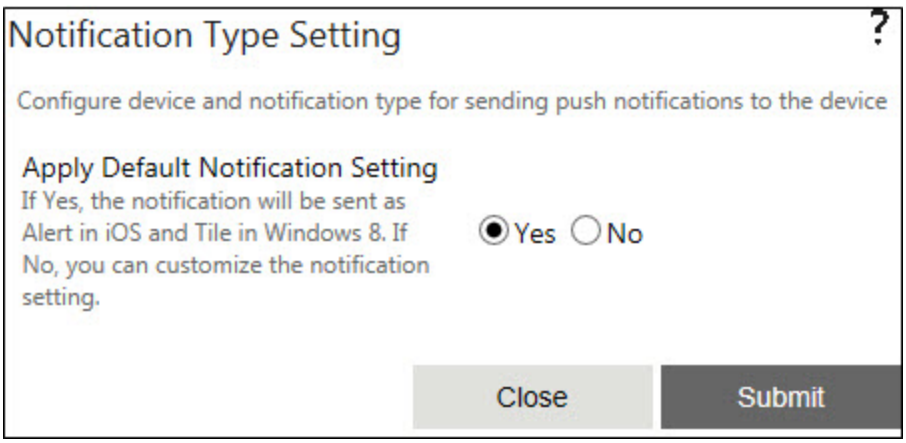
Selecting the **Workflow Initiator** allows you to assign the task to the Workflow Initiator, that is the user who closes the workflow. An expression is built automatically and updated in the text area of Recipient Resource.

Set Notification Type

In **AVEVA Work Tasks Pro Push Notification Activity**, you can configure different types of notification based on the device being used. You can choose to continue with the default notification type or customize the notification type.

To use the default notification type, set the option **'Apply Default Notification Setting'** as **Yes**. Default notification for iOS is **Alert** and for Windows 8.1 it is **Tile**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Configuring Custom Notification Type

1. Click the edit icon in **Notification Type** property. The **Notification Type Setting** is displayed.
2. Select **No** for the option **'Apply Default Notification Setting'**. The option of **Device and Notification Type** is now available for selection.

   The default selection for the option **'Apply Default Setting'** is **Yes**.
3. Select the operating system used on your device and then select the **Notification Type**.
   - iOS operating system, supports only the default notification specific to iOS.
   - For Windows 8.1 operating system, the available notification types are **Tile** and **Toast**
     - When **Tile** is selected the notifications are displayed on tile of the application
     - When **Toast** is selected the notifications are displayed as pop-up notifications

---

Note: Push Notification activity will continue to execute successfully even after unregistering for notification from the **Farm Configuration Wizard**. In such cases, unpublish the workflow, unregister for notification from the **Farm Configuration Wizard**, and then republish the workflow.

AVEVA Work Tasks Pro Push Notification Activity Scenarios

Activity in waiting state

The activity will be in waiting state, when the first attempt to send notification fails and the property **Retry on Failure** is set to **Yes**.

When the first attempt by the activity to send the notification fails, an entry is made into the database. This entry is read by the **ArchestrA Mobile Notification Service** which in turn attempts to re-send the notification based on the value entered in the **Retry Count** property.

The activity will be completed only after all the attempts to retry the notifications are made.

Activity output is successful when Recipient Resource is inactive/blocked/not logged in

When multiple recipient resources are configured in the AVEVA Work Tasks Pro Push Notification Activity and one or more recipient is inactive/blocked/not logged in, the activity will be considered as successful if the notification is sent to all other active recipients.

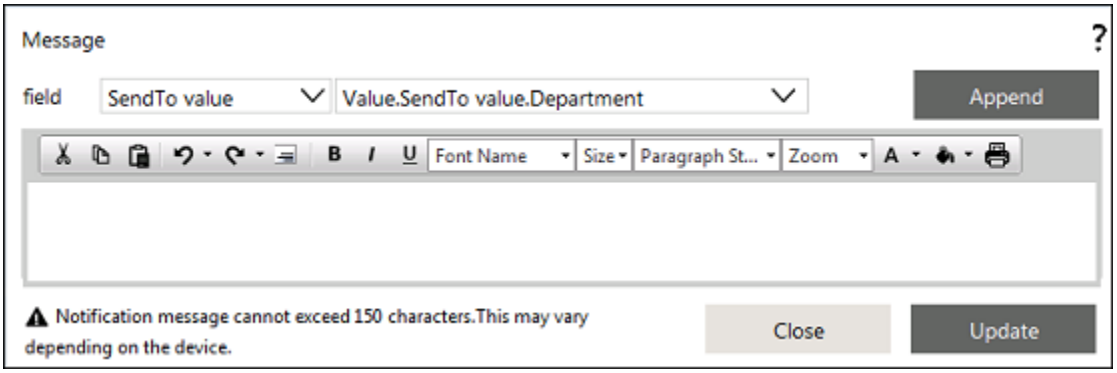
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Unsuccessful activity output scenarios:

- First attempt to send notification fails and **Retry on Failure** is set to **No**.
- When **Retry on Failure** is set to **Yes** and **Activity is successful** is set to **If notification is sent to at least one device:** If notification is not sent to any device even after consecutive attempts to resend notification based on the value specified in **Retry Count**.
- When **Retry on Failure** is set to **Yes** and **Activity is successful** is set to **If notification is sent to all devices:** If notification is not sent to one or more devices even after consecutive attempts to resend the notification based on the value specified in **Retry Count**.

Configure Notification Message

Use the **Message** property to configure the content of the notification message.



You can also build an expression to configure the notification message. The message will be updated dynamically at runtime.

For iOS devices, the notification message can have a maximum of 150 characters. In Windows 8.1 devices, if the notification message exceeds 208 characters, then the notification message is truncated. In both iOS and Windows 8.1 devices, the notification message is displayed in plain text, hence all html tags are removed.

Engine Activities

The following are the types of Engine activities:

- BlockingOutput Activity: The BlockingOutput activity facilitates synchronous execution of workflow activities with an active form.
- Database Activity: The Database activity enables execution of queries, stored procedures, and functions from the workflow.
- Decision Activity: The Decision activity can be used to insert a decision in the workflow.
- ForEach-Loop Activity: The ForEach-Loop activity is used to loop an activity through a collection within an XMLIterator Variable.
- For-Loop Activity: The For-Loop activity can be used to start a loop within a workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Logger Activity: The Logger activity logs messages like information, warnings, and errors.
- Next Loop Activity: The Next Loop activity can be used as the terminal activity in a loop initiated using a For-Loop Activity.
- Rule Activity: The Rule activity allows complex rules or expressions to be built into the workflow.
- Set Workflow Output Variable: The Set Workflow output variable activity can be used to set the output of the parent workflow from the child workflow.
- Synchronize Activity: The Synchronize Activity synchronizes the execution of different activities.
- Update Parent Variables Activity: The Update Parent variables activity is used to update a parent workflow variable from a child workflow, provided both child and parent workflow belong to the same repository.
- Update Variable Activity: The Update Variable activity can be used to update the value of Variables and XmlVariable that can be used in the subsequent activities.
- Workflow Activity: The Workflow activity enables the execution of a child workflow from a parent workflow.
- XML Navigator: The XML Navigator activity is used to either locate or navigate to a specific record in a collection within an XMLIterator Variable.

BlockingOutput Activity

Activity Description

The BlockingOutput activity facilitates synchronous execution of workflow activities with an active form.

This is typically used in forms, whose fields need to be updated based on the output of multiple workflow activities of the same workflow, executed one after the other, without closing the form.

Activity Properties

The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

Specify the Output to be returned to the application that calls this workflow when this activity is encountered.

Note:
- This activity works only when the event is in process like Item Adding, Item Updating and not for the events like Item Added, Item Updated because once the process is completed the item cannot be blocked any more. This activity is useful when you want to prevent the users from performing specific tasks. For example: If you want to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

prevent users from updating an existing item.
- When you use the BlockingOutput activity for returning the blocking output value, ensure that all the activities being executed before this activity, do not have sleeping behavior. For example: There should not be any activities, such as human or scheduler activities, which go to the sleep mode. However, Workflow activity with some specific settings can be included before this activity. For more information about the Workflow activity, see Workflow Activity.

Database Activity

Activity Description

The Database activity enables execution of queries, stored procedures, and functions from the workflow. The database activity establishes connection with the database server and retrieves or updates data in the database. The activity gets the schema of the database table to an XML variable according to the query, stored procedures, or functions passed.

Application Scenario

Consider a scenario where a manager wants to send appraisal forms to all the employees. The manager can add the database activity with a query to select the email address column from the Employee Database. The Output XML will have the schema based on the query and this XML variable can used to assign employees to whom the appraisal forms are to be sent. When the workflow is initiated, the query in the database activity will be executed and all the employees will receive the appraisal form in their Inbox.

Activity Properties

You can configure the Database activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **Database** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

Note: Every time you input a value for a property, ensure to save the data. Click the **Save** button on the **Activity Properties** window. You need to save the property so that the subsequent properties in the activity can inherit the required information.

Name & Description

You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

You can use these properties to configure the functionality of the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Database Connection

Database connection property is used to get the connection to the Database Server. You can use this property to select the Database connection (which has been configured in **Enterprise Console > Settings > Database Connections**).

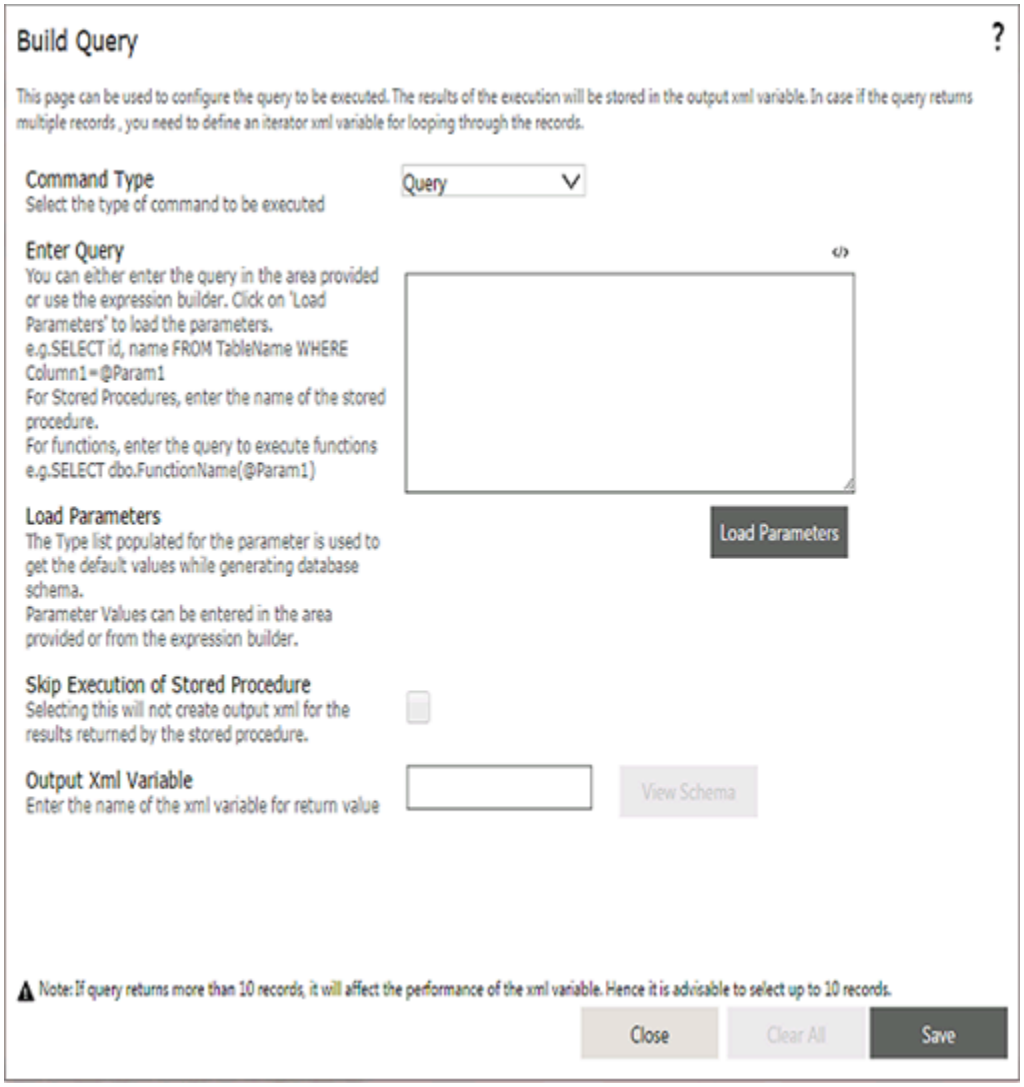
Property Type: Mandatory (This property must be set if the activity has to establish connection with the database server.)

Query Builder

To enter the query, click the **Query Builder** property to open the property window.

In this window, set the following properties:

1. Command Type
2. Query
3. Output XML variable



Command Type: The command types that can be selected

- Query

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Stored Procedure
- Functions

Enter Query: As the Command Type is selected, you can enter a query, Stored Procedure, or Function.

Note: The **Expression Editor <>** icon is displayed only for the **Command Type "Query ",** as the expressions set cannot be evaluated in design time for stored procedures and functions.

Query: Query can be entered in the text box or from the Expression Builder.

For example, the query can be **Select * from EmployeeTable** and if the query is to be passed with parameters then the query will be **Select * from EmployeeTable where EmployeeID=@Id**. Here @Id is the parameter. When you click **Load Parameters**, it will load the parameters within the query. In the Value field you can pass the value for the parameter.

Query with 'Order By', 'Group By' keywords are supported.

The following Parameter values can be saved with empty values:

- **Name**: Displays the name of the parameter.
- **Type**: Select the Parameter type from the drop-down list. This type helps you to execute the query with internally defined default values only while generating the database schema. The appropriate Type needs to be selected for the Query and Functions. For stored procedures the Types will be automatically selected.
- **Value**: Type the value of the parameter.

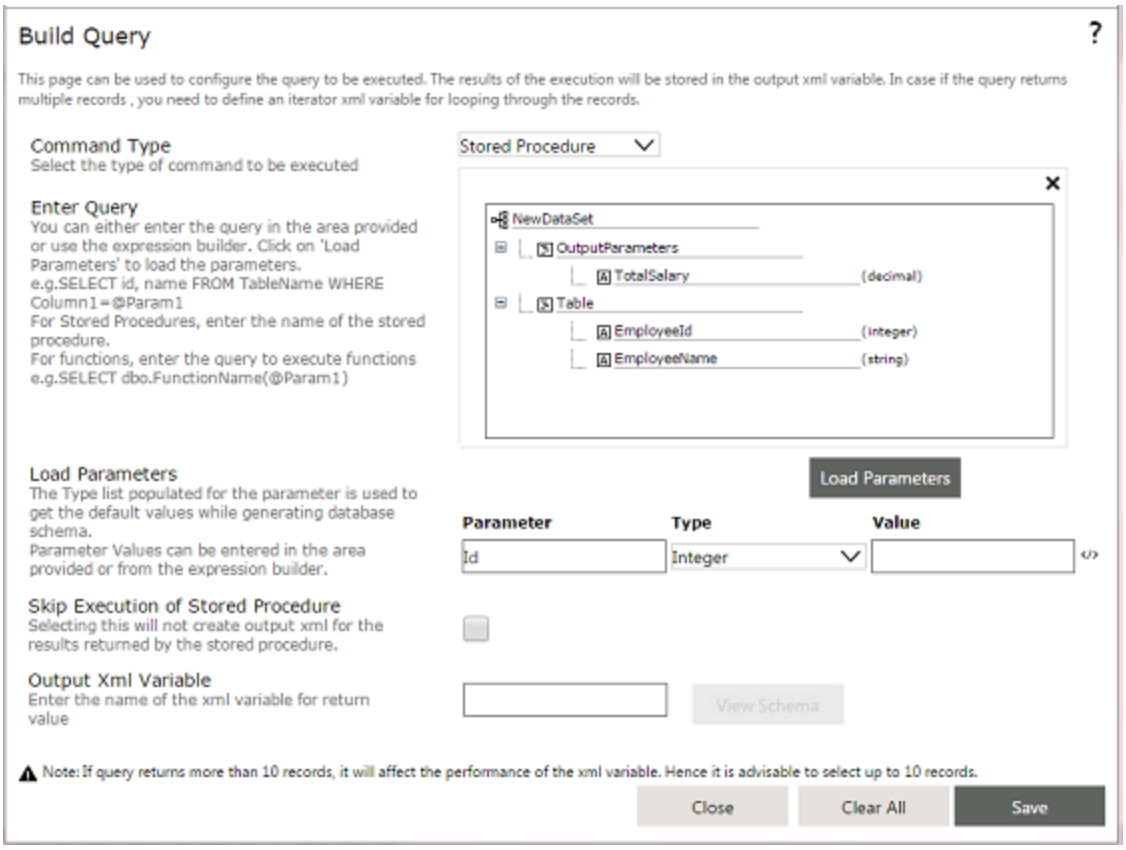
Parameter helps you retrieving values from the database where the value is null or empty.

If the query is to retrieve values from the database we have to specify the Output XML variable name. For more information, see the table *Database Activity*.

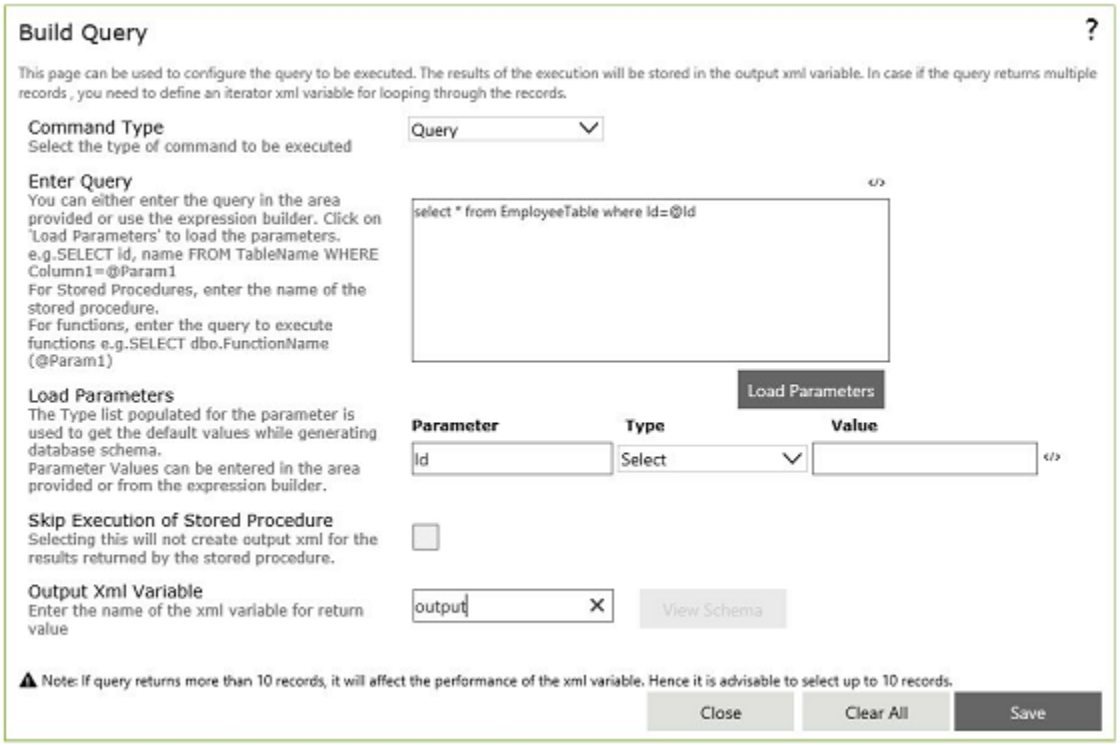
Multiple queries are not supported. If multiple queries are used, XML variable will carry the result of the last query written.

Click **View Schema** to view the schema of the output XML variable "outXML"

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



XML variable used in one activity cannot be reused within other database activities.

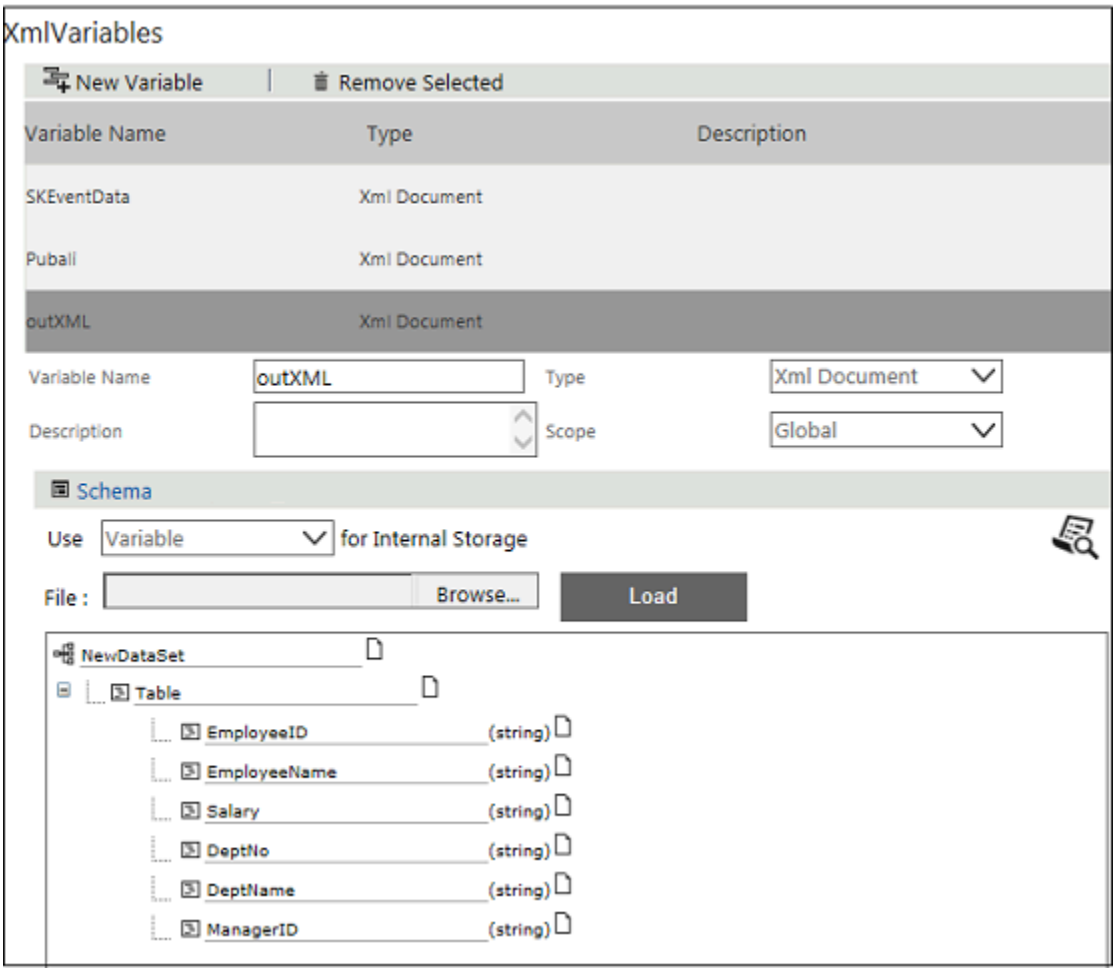


If the output schema containing return values have been modified after the property is saved, open the Query Builder property and click Save. The Schema will get updated with the latest values. Click **View Schema** to see

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the updated changes.

This XML Variable "outXML" will be added in the **Start > XML Variables** as shown below.



Parameter values cannot update data types like 'image', 'blob' and 'unique identifier' from XML variables or variables.

Stored Procedure

Stored procedures can be executed by simply specifying the name of the Stored Procedure (provided there is a stored procedure created in your database server). On the click of "Load Parameters" it will load the parameters present in the Stored Procedure. If you want to retrieve values on executing the Stored Procedure you can specify the name of the XML variable. If return values from the stored procedure are stored in output parameters, the activity will load the XML schema with the output parameters.

To discover the output XML schema, system internally executes stored procedure with parameter values as empty string or null. If a stored procedure has validations on parameter or data integrity constraints, you cannot configure a stored procedure.

If a stored procedure returns the values for return Result Set and output parameters, then both the values will be available in the XML variable.

Note: If there are multiple Return Result Sets, then only the first return Result Set will be considered.

- **Skip Execution of Stored Procedure**: To avoid the execution of stored procedure to get the output xml schema, select this check box so that you can configure stored procedure with parameter validation or data integrity constraints.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:
- It is recommended to use proper parameter values/variables that has appropriate values at runtime to execute the stored procedure.
- To execute a stored procedure, which expects default values for execution, you can create a wrapper stored procedure with the same parameter. You can then call the actual stored procedure with default values passed to the stored procedure if the parameter values are empty or null.

Functions

Functions are executed in the same way as executed in the database server, for e.g. "select dbo.FunctionName". If the function contains parameters you should specify the exact parameters. For e.g. "select dbo.FunctionName(@param1, @param2)" (provided there is the function created in your database server). If you want to retrieve values on executing the function you can specify the name of the XML variable. If return values from the function are stored in output parameters, the activity will load the XML schema with the output parameters.

Note:
- Table valued functions are not supported.

Output XML Variable for return value:

Output XML variable will have the schema of the table as per the query entered. If the query is "Select * from EmployeeData" then the XML variable will have all the columns as nodes in the schema which can be looped through and used in the subsequent activities. Output XML Variable name should be unique, it cannot be repeated.

Note: To create an XML Iterator of the output XML Variable the node selected should always be the "Table" node.

Capture runtime exceptions in a variable

You can create a variable with the name **^<ActivityName>Error** in the **Start** activity. The exception message will be available in the variable if an exception occurs.

For example: For an activity named as "DBAct1", you have to create a variable with name as **^DBAct1Error** in the **Start** activity.

For more information on creating variables, see Start - Variables.

Property Type: Mandatory (This property must be set if the activity is to retrieving any value from the database (i.e. if query is of "select" type, else this field will be optional because it will not load any schema for 'insert', 'update' or 'delete' type of query)

Retry Connections: This property allows setting the maximum retries if a Database goes Offline at run time. By default the value set will be 1.

Property Type: Optional (This property needs to be set only if necessary.)

The following table shows the SQL database types and their corresponding parameter types for a database activity:

SQL Database Type	Database Activity Parameter Type
bigint	Integer
binary	Byte
bit	Boolean

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SQL Database Type	Database Activity Parameter Type
char	String
date	Date
datetime	DateTime
datetime2	DateTime
datetimeoffset	DateTime
decimal	Decimal
float	Double
int	Integer
money	Currency
nchar	String
numeric	Decimal
nvarchar	String
real	Single
smalldatetime	DateTime
smallint	Integer
smallmoney	Currency
sql_variant	Object
time	Time
timestamp	Binary
tinyint	Byte
uniqueidentifier	Guid
varbinary	Binary
xml	Xml

Note: If the Database connection is Windows authenticated, the Identity of the Quickflow Application Pool should be the user who has access to the same database.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Decision Activity

Activity Description

The Decision activity can be used to insert a decision in the workflow. The output of the activity can be used to determine the path of workflow execution.

Activity Properties

You can configure the Decision activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **Decision** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The property in the Configuration group has been described below. You can use this property to specify the decisions.

Decisions - The Decisions property allows the user to create complex rules to control the execution path of the workflow. In contrast to the Rule action, which only allowed a True/False kind of decision, the Decision property can be used to rules with multiple outputs based on the condition specified.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

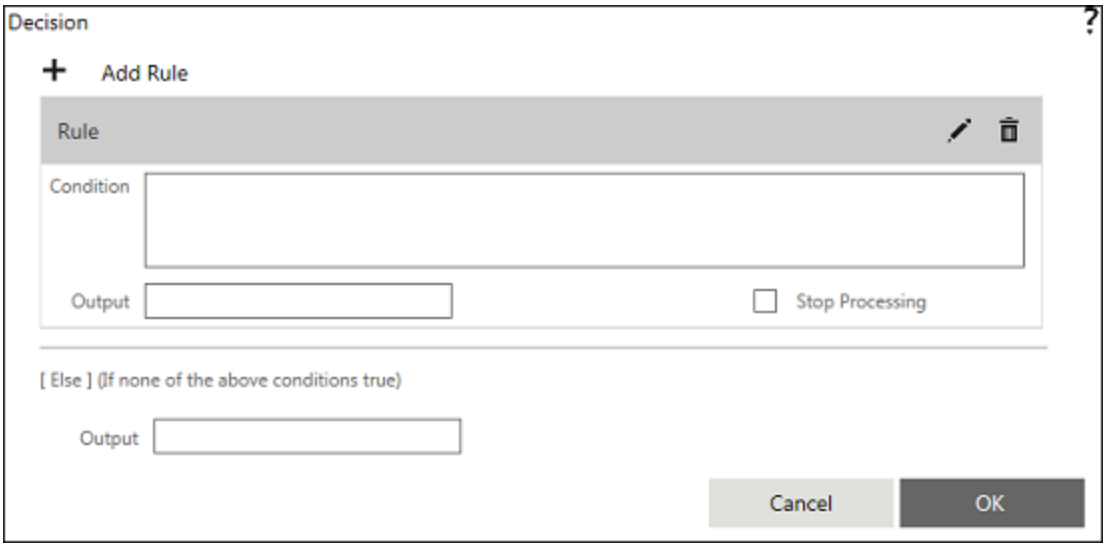
Steps to set the Decisions property

See Decision: Decisions for a detailed description of the Decision property window.

Decision: Decisions

You can create conditions for the decision action in this property window. To open this window, click the edit button for the 'Decisions' property. The following window shows the **Decision** property window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



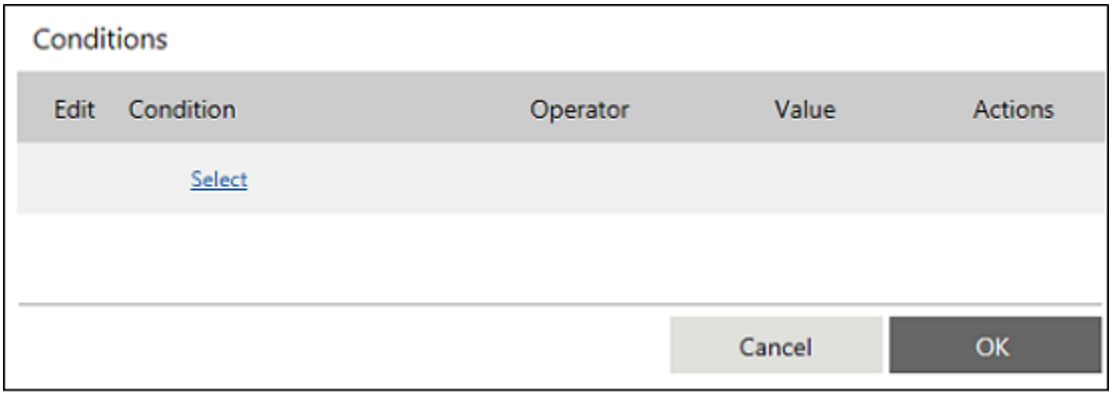
This **Decision** window has the following controls to specify the condition for a rule:

- **Add Rule**: Click **Add Rule** to add a new set of the above controls for adding another rule.
- **Condition**: Specify the expression for the condition in this text area.
- **Condition icon** : Click the condition icon to open the Condition window to build the condition expression.
- **Output**: Specify the output for the rule in this text box.
- **Stop Processing**: Select this check-box to enable the process to be stopped when this rule is "true".

The window also has the **Output** text box to specify the output for the exception, that is when none of the rule conditions are true, in the [**Else**] (**If none of the above conditions true**) field.

Condition window

You can build expressions for the rule condition in this window, which is opened when you click the Condition icon in the Decisions property window. This window has the Select link, which you can use to specify the left hand side of the expression. On mouse-over this link opens a combo, with a collection of objects that you can specify for the left hand side of the expression. The window has the **OK** button at the bottom to save the new expression. The **Cancel** option allows you to cancel the expression and return to the property window.



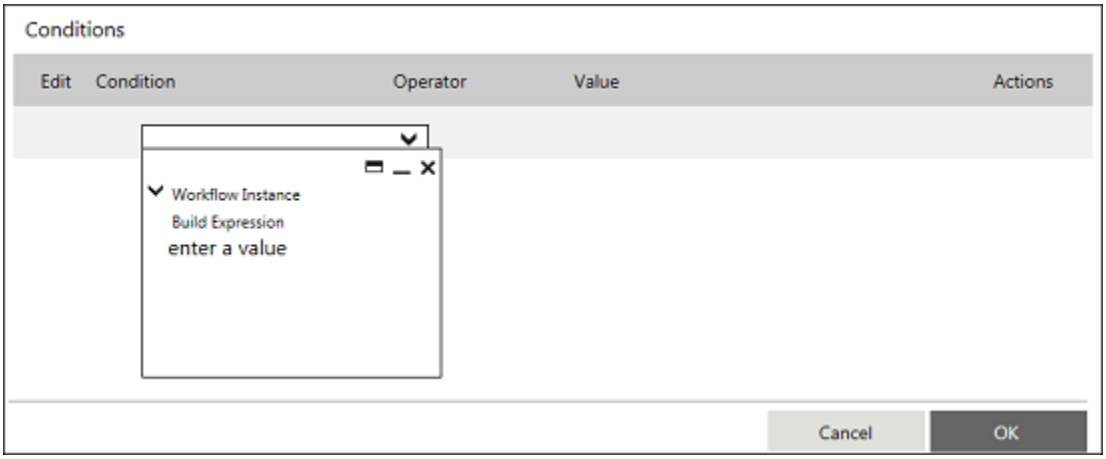
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Building an Expression for the Decision Condition

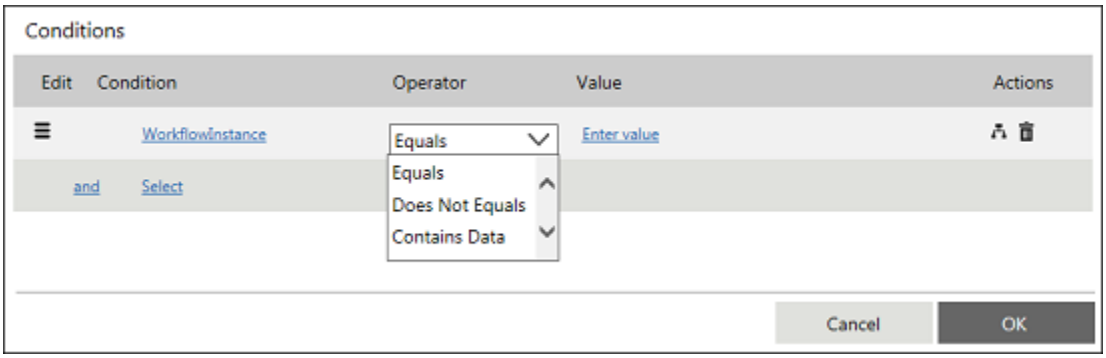
To illustrate the process of selecting an object for the right hand side of the expression, let us consider an XML Variable with "Employee" as a node and "FirstName" as an element under it. Specify this as the object for the left hand side of the expression. From the mouse over drop down, scroll down till you see the XML Variables node. Expand this node and the "Employee" node under it. Select the "FirstName" option.

Note: The C# code is not directly supported in the Expression editor for the Decision activity. If you want to return any value from the C# code, refer to the Returning Value from the C# Code topic.

The following window shows how to set an object for the left hand side of the expression:



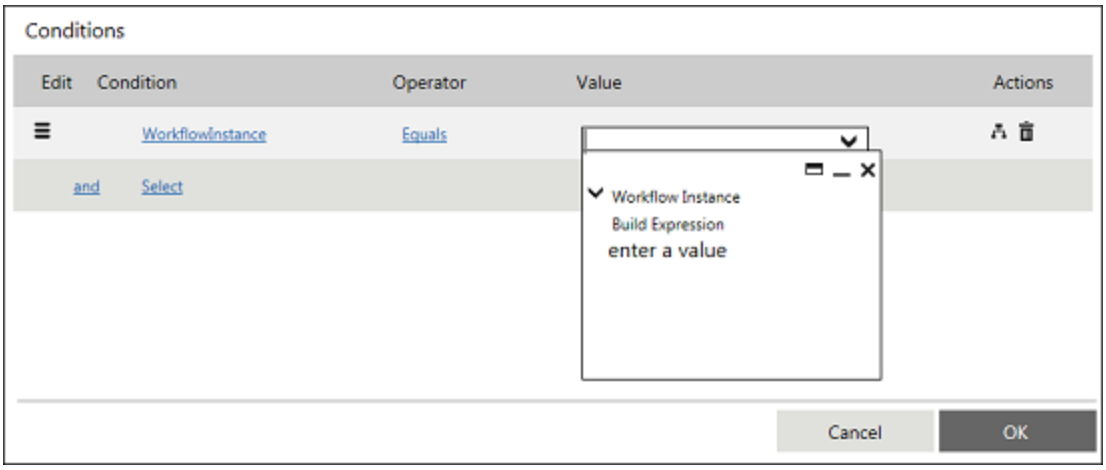
This will display the operator combo with operators corresponding to the data type of the selected object. In this case, the data type being string, the operators are string operators, shown in the image below. Select "Equals" as the operator.



Operators combo with string operators

The default control to specify the object for the right hand side of the expression, which is a text box, is displayed at the right hand side of the row, as shown in the image below. You can specify a matching value in this text box. As the object selected is First Name, specify "John" here.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



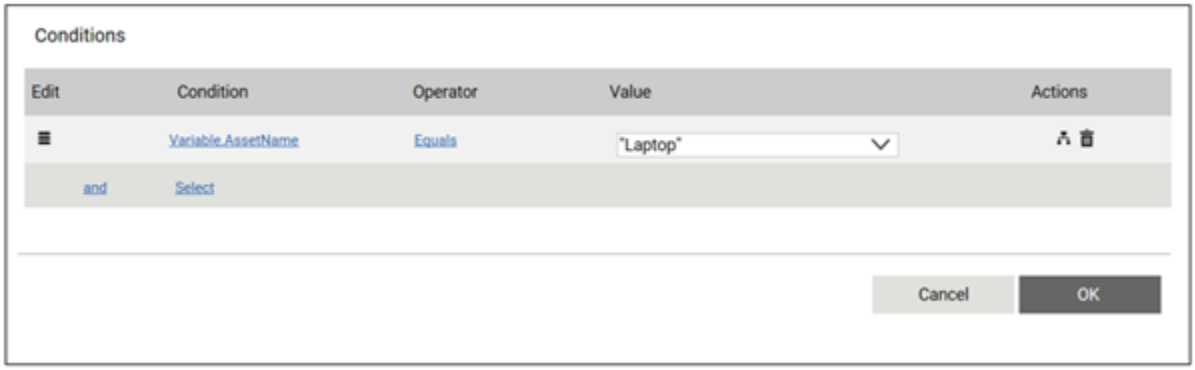
Default control for specifying the right hand side of the expression

This will complete the first expression and the settings will be displayed as shown below. You can mouse over any link to edit the value.

Using the Expression mode icon

Click the Expression mode icon, , to change the default text box to the Expression mode. In the Expression mode, you can select an object from the combo for the right hand side of the expression.

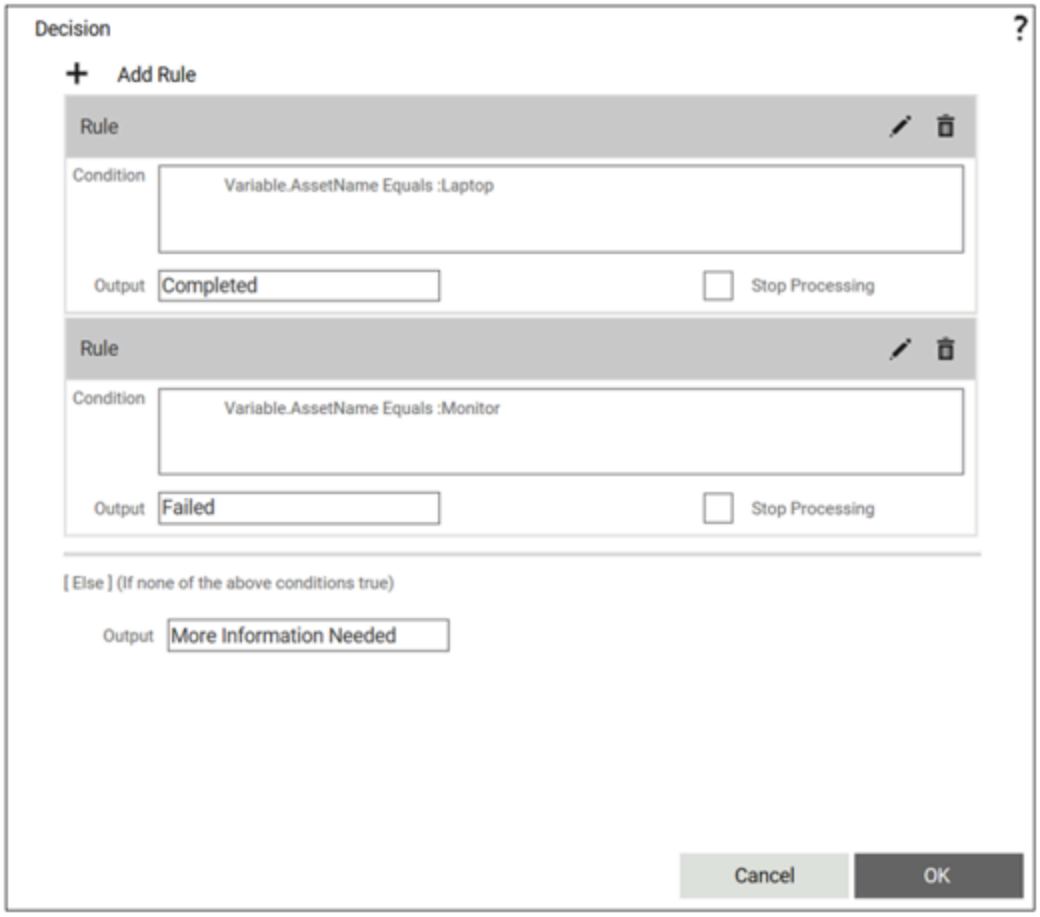
Consider that the left hand side of the expression is VirtualAssetName. You can assign the Variables.AssetName as the value for the right hand side.



You can select "Build Expression" in the Expression mode combo to open the Expression Editor and build an expression instead of specifying an object on either side of the expression. You can select "Enter a value" in the right hand side combo to return to the default mode, that is, the text box.

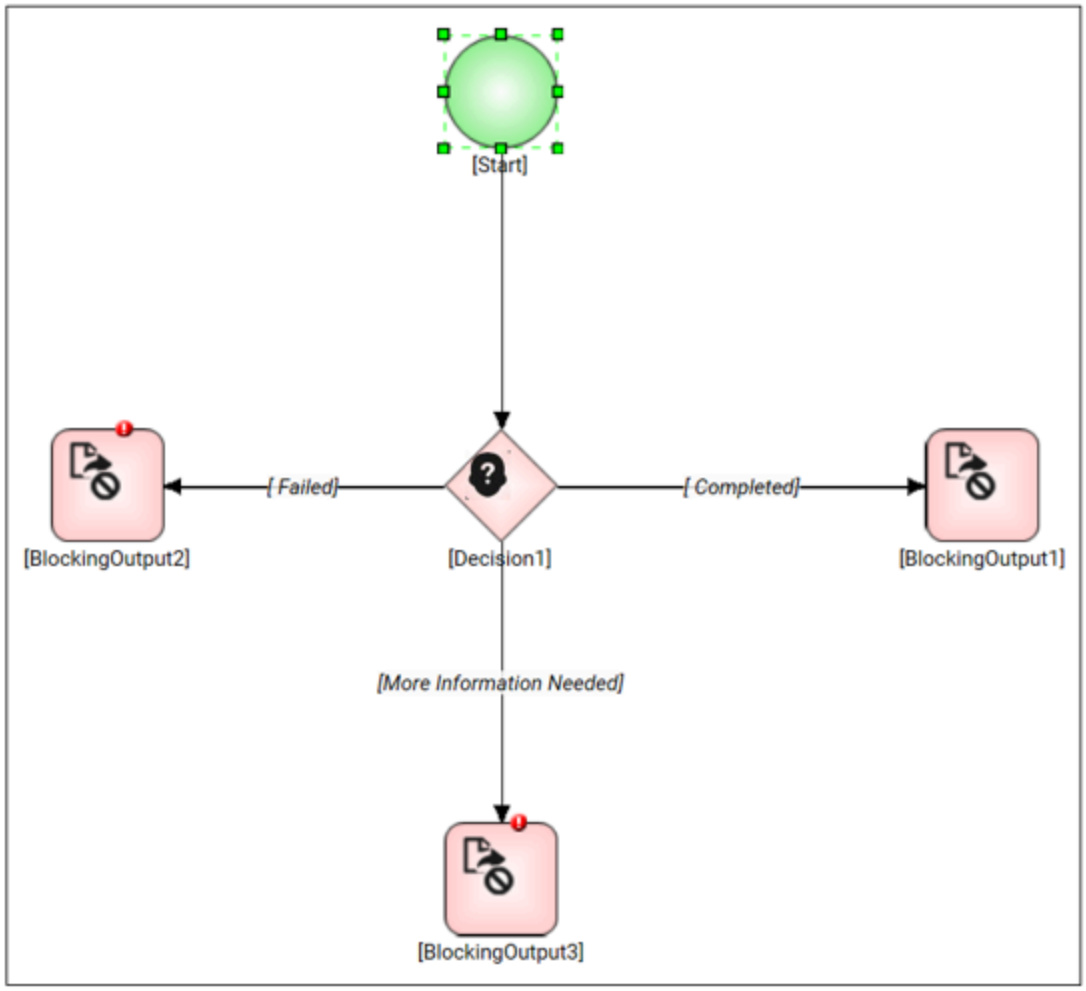
Click **OK** to save the expression and return to the **Decision** property window. The following image shows the window with Decision conditions added:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The Decision activity would be typically used in a workflow as shown below, with each decision output determining a different path for workflow execution.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Decision activity in a typical workflow

To return Value from the C# Code

You must follow the following steps to return any value from the C# code:

1. Declare a variable in the start activity.
2. Initialize the variable in Start activity or using update variable using the C# code.
3. This variable can be used in the Decision activity.

Note: In Decision Activity, when you specify a value for comparing numeric, date, time, or date, time type of variables, you must specify the value in Invariant format.

ForEach-Loop Activity

Activity Description

The ForEach-Loop activity is used to loop an activity through a collection within an XMLIterator Variable. It allows repeated execution of an activity or group of activities. It must always be used along with the Next Loop Activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

You can configure the ForEach-Loop activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **ForEach-Loop** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

Name & Description

Specify the name and description for the activity in this group.

Name - Enter the name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - Enter a brief description for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The Configuration group has a single property for selection of the XMLIterator Variable to be looped through.

XMLIterator - Select the XML Iterator variable to be looped through. All XMLIterator variables as declared in Start activity are available for selection.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Example of XMLIterator Variable in ForEach-Loop

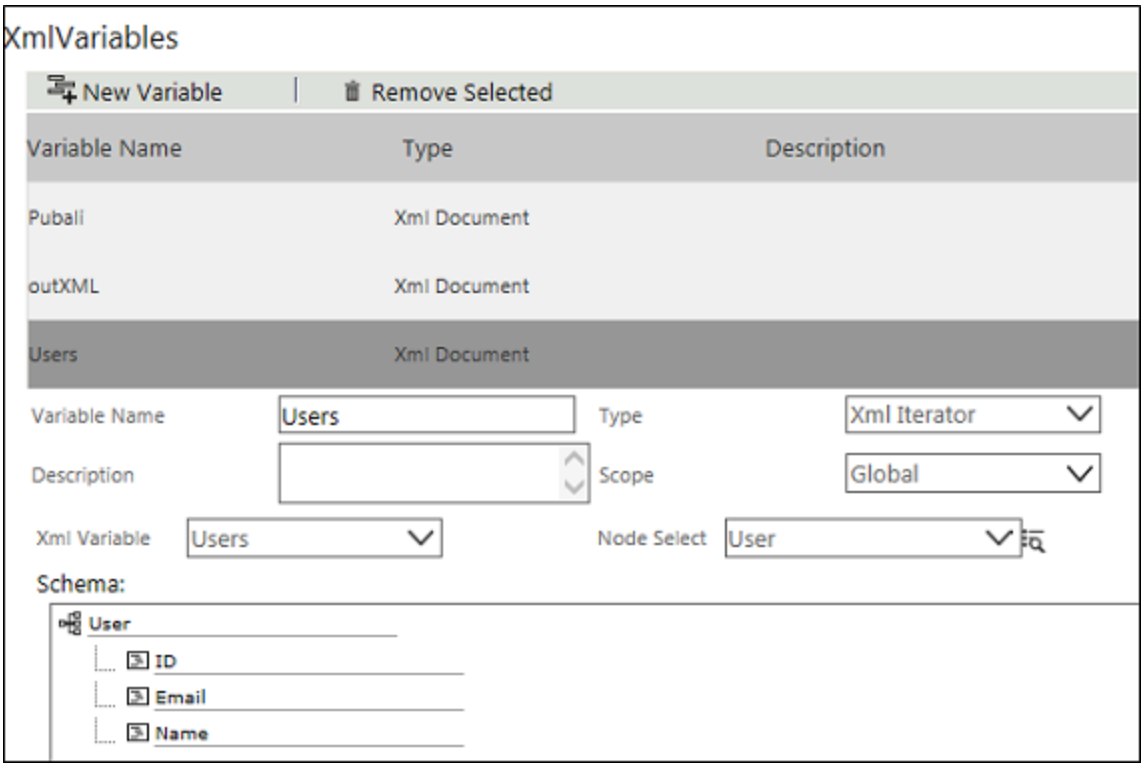
In this example, the XMLIterator Variable is used to get approval from all managers in the Managers.XML file containing the list of all users who are managers.

Managers.XML file

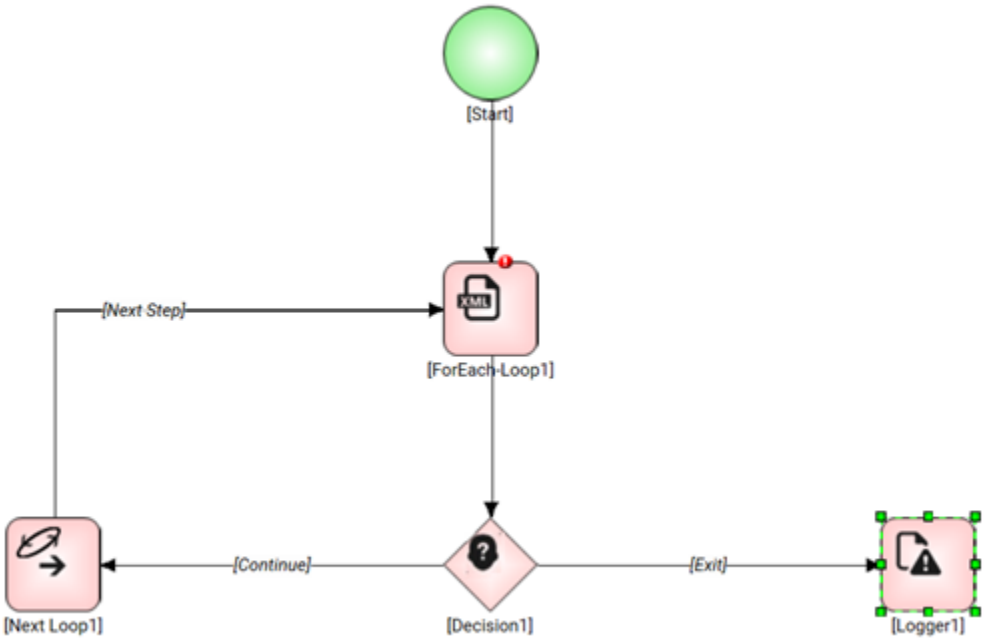
```
<Managers>
 <Users>
<User id=1>
 <Name>John</name>
 </User>
<User id=2>
 <Name>Jim</name>
</User>
<User id=3>
 <Name>Victor</name>
</User>
<User id=4>
 <Name>Andy</name>
</User>
 </Users>
</ Managers >
```

Step 1: XMLIterator Variable is defined for users. The node Users holds the collection of managers.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

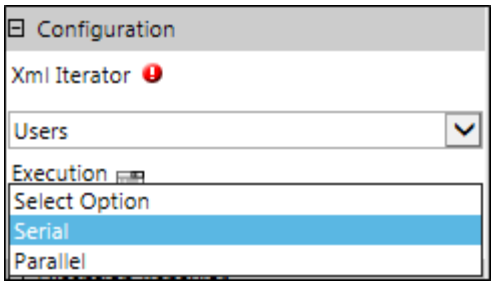


Step 2: The Decision activity is set within a ForEach-Loop activity.



Step 3: The XMLIterator Variable Users is defined in XMLIterator field of ForEach-Loop properties. The Execution type is set to Parallel as the decision is to be processed simultaneously.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Step 4: The XMLIterator Variable Users is specified in the condition for making a decision.

Output: When the above workflow is invoked, Quickflow Engine resolves "XMLVariables.user.users.ID".

For-Loop Activity

Action Description

The For-Loop activity can be used to start a loop within a workflow. It is useful for repeated execution of an activity or group of activities. It must always be used along with the Next Loop Activity.

Activity Properties

You can configure the For-Loop activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **For-Loop** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Steps to set the Loop Count property

1. To specify the loop count, click on the button for the 'Loop Count' property to open the property window.
2. In this window, enter the count value directly in the text editor.
3. If you want to assign the loop count dynamically during run-time, you can do so using the **Field** controls. For example, you can add a variable or content pointing to the loop count from the two drop downs for **Field**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

and click on the **Append** button. This will add the placeholder for the variable or content in the text editor. At run-time this placeholder will be replaced by the corresponding value, that is the loop count.

4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After specifying the count, click the Update button.
7. Click the Close button to close the window without saving the parameter.

See For - Loop Count for a detailed description of the Loop Count property window.

Activity outputs

The For-Loop Activity returns the following values -

Step: Displayed to indicate the loop step

Loop End: Displayed when the end of the loop is reached

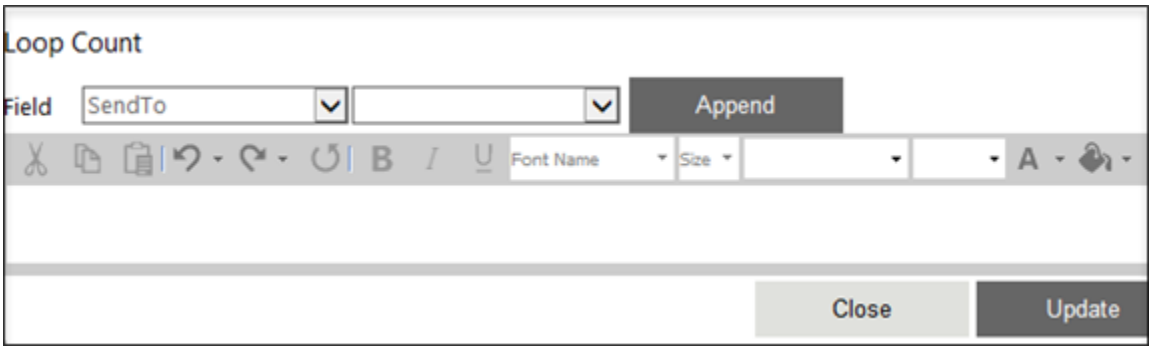
Variable Not Found: Displayed when the specified variable is not found

Invalid Count: Displayed when there is an invalid loop count

For-Loop Count

Loop Count Property Window

You can set the loop count that will be used to determine the iterations of the loop in this property window. To open this window, click the button for the 'Loop Count' property.



The window has a text editor to enter the loop count. You can do this either by entering a discrete value directly in the text editor or by using the **Field** controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify **Variable**, **Content**, **Value** or **XML Variable**.
- The Value text box/drop down - You can specify the actual value of the loop count. The control displayed depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the loop count in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

loop count.

- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the loop count details. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the loop count
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- **Update** - You can click this button to save the loop count
- **Close** - You can click this button to close this window

Logger Activity

Activity Description

The Logger activity logs messages like information, warnings, and errors. The messages are logged to the Logger Console from the workflow.

Activity Properties

You can configure the Logger activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **Logger** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name & Description

The properties in the Name & Description group is described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group is described below. You can use these properties to configure the functionality of the activity.

LogType - This property allows the user to select the type of message to be logged. The drop down lists the following options: Information, Warning, and Error.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

LogMessage - This property allows the user to specify the message to be logged.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Log Message property

See Log - Log Message for a detailed description of the Log Message property window.

Activity outputs

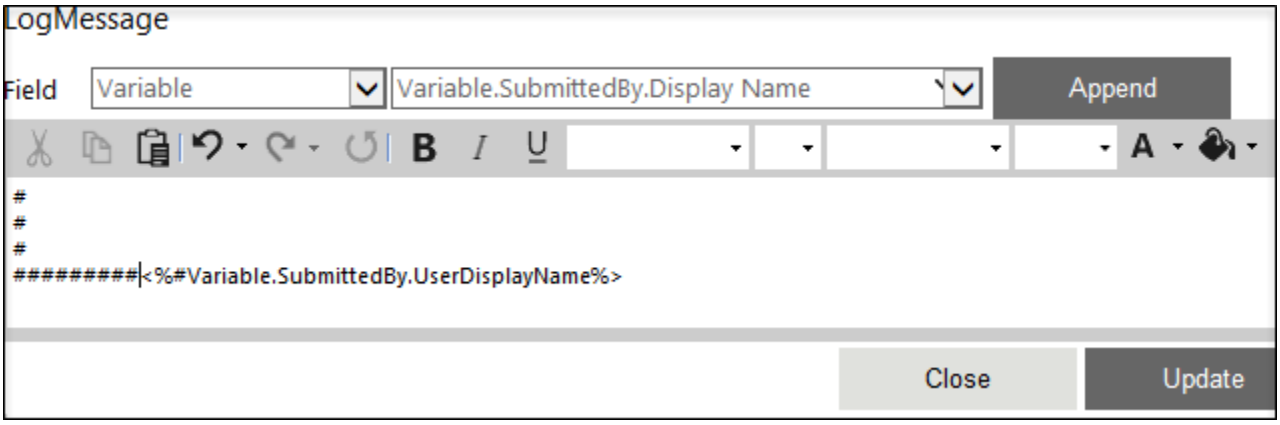
The Logger Activity does not return any value.

Example for a Logger:

Step 1: In the Activity Properties of the Logger1, select LogType as Information, and click Save button of Activity Properties.

Step 2: Log Message - Select the field as **Variable** in the first drop-down. In the second drop-down, select the Variable.SubmittedBy.DisplayName, and then click **Append**.

Step 3: The entry would be as this: <%#Variable.SubmittedBy.DisplayName%>. The '#' Symbol is just for quick identification of data in the Logger.



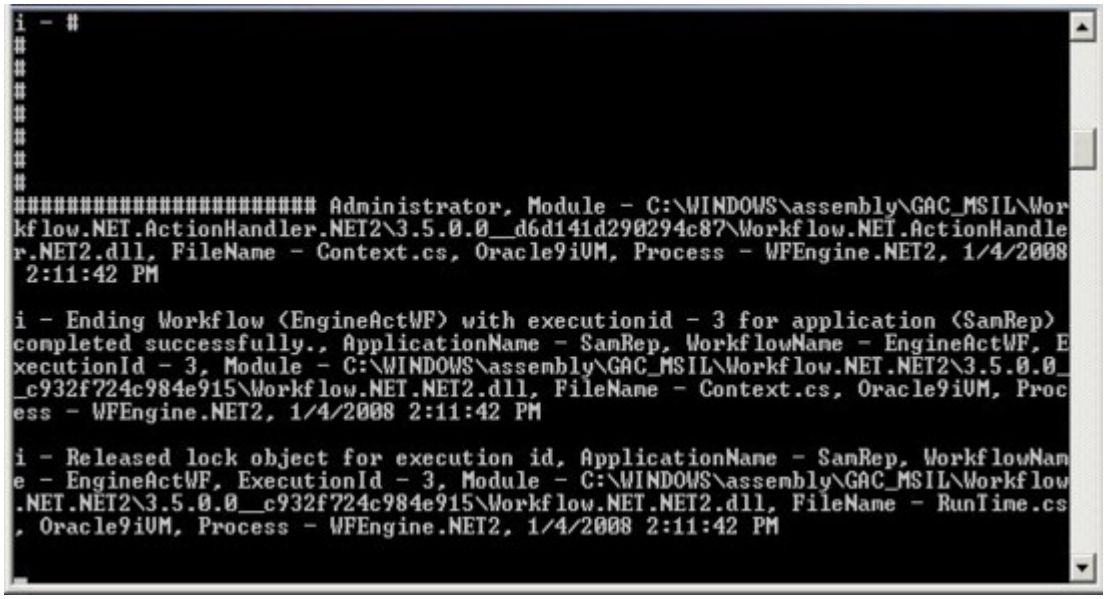
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: Before publishing the workflow, start the Logger Console.

Output:

In the Logger, you can see the contents of the variable Variable.SubmittedBy.UserName with the '#' symbols.

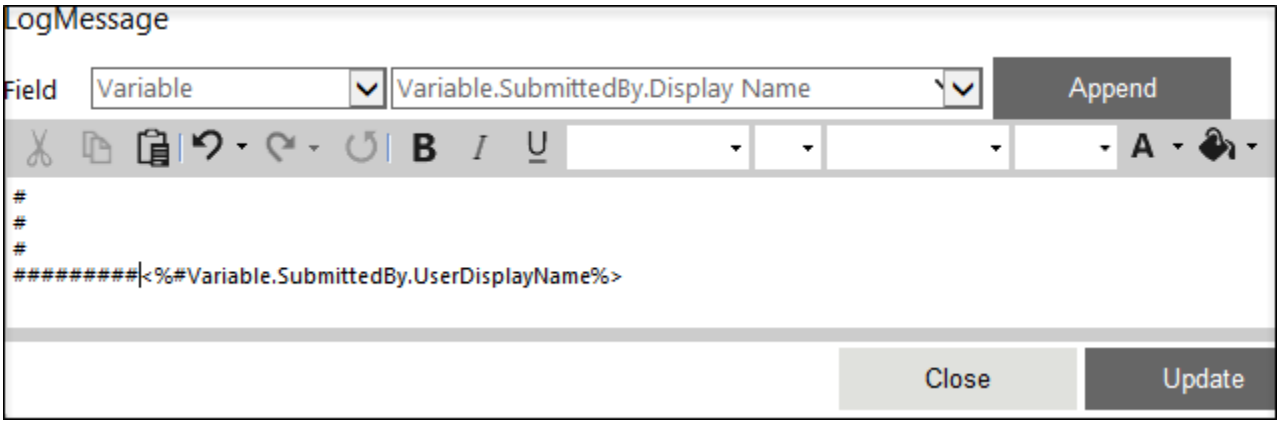
In the below figure you can see the name Administrator at the end of the # symbols.



Log - Log Message

Log Message Property window

You can specify the Log Message, which should be logged to the Log Console from the workflow, in this property window. To open this window, click the button for the 'Log Message' property.



The window has a text editor to enter the Log Message. You can do this either by entering a discrete value directly in the text editor or by using the Field controls at the top of the window. These controls are described below:

- The Value type drop down - You can select the type of value here. You can specify Variable, Content, Value or XML Variable here.
- The Value text box/drop down - You can specify the actual value of the Log Message. The control displayed

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

depends on the option selected in the first drop down. If you had selected "Value", it will display a text box here. You can enter the Log Message in this box. If you had selected "Variable", "Content", or "XMLVariables" instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option to assign the Log Message.

- **Append** - You can click this button to add the specified value, variable or content to the text editor.

The text tool bar above the text editor allows you to manage and format the Log Message details. It has the following text tools –

- Cut (Ctrl+X) - to cut text
- Copy (Ctrl+C) - to copy text
- Paste (Ctrl+V) - to paste text
- Undo (Ctrl+Z) - to undo the last action
- Redo (Ctrl+Y) - to undo the last action
- Repeat Last Command (F4) - to repeat the last command that was executed

It also has the following formatting & other tools –

- Bold (Ctrl+B) - to make text bold
- Italic (Ctrl+I) - to italicize text
- Underline (Ctrl+U) - to underline text
- Font Name - to select the font for the Log Message
- Size - to specify the font size
- Paragraph Style - to select a paragraph style
- Zoom - to zoom in and zoom out the view
- Foreground Color - to set text color
- Background Color - to set the background color
- Print - to print the text

At the bottom of the window you will find the following buttons:

- Update - You can click this button to save the Log Message
- Close - You can click this button to close this window

Next Loop Activity

Activity Description

The Next Loop activity can be used as the terminal activity in a loop initiated using a For-Loop Activity. It will check if the count value has been reached. If the value has not been reached it will increment the 'Loop Variable' set in the For-Loop Activity and repeat the execution of the activities in the loop.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

You can configure the Next Loop activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **Next Loop** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Activity outputs

The Next Loop Activity returns the following value -

Next Step: Displayed to indicate the next step in the loop

Rule Activity

Activity Description

The Rule activity allows complex rules or expressions to be built into the workflow. The expression will either result in 'true' or 'false'. The workflow engine evaluates the expression and assigns the result (true/false) to the output. The engine determines the path of the workflow from the output.

Activity Properties

You can configure the Rule activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **Rule** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration

The property in the Configuration group has been described below. You can use this property to specify the decisions.

Condition - This property allows the user to frame complex expressions for the workflow execution. The expression results either in 'true' or 'false'. The workflow engine, after evaluating the expression and getting the result, decides on the path for the activity, as per the process design.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the Condition property

1. To specify the condition for the Rule activity, click the button for the 'Condition' property to open the property window.
2. In this window, build a complex expression for the rule using the fields provided. Select a type of value (Variable, Content, XMLVariables) from the first drop down list, then select the corresponding value from the second drop down. This forms the left-hand side of the rule condition expression. Next, select the operator for the condition from the third drop down. Finally, select the type of value (Value, Variable, Content, XMLVariables), and the corresponding value for the property from the respective drop downs to add the right-hand side of the rule condition expression. Select "End" from the last drop down to complete the expression. If you are building a compound expression, you can select one of the logical operators to connect the different conditions that form the expression.
3. Add the condition to the expression by clicking **Add**. This will display the expression in the text area.
4. You can remove a condition already added to the expression by clicking **Undo**.
5. You can also clear an existing expression if necessary by clicking **Clear**. This is useful if you want to clear large expressions in the text area in order to build a new expression.
6. After building the expression, click **Update** to save it.

See Rule - Condition for a detailed description of the Condition property window.

See the Using XML Variables section in the About the XML Variable User Interface topic for details on how to use the XMLVariables to build the expression.

Activity outputs

The Rule Activity returns the following values -

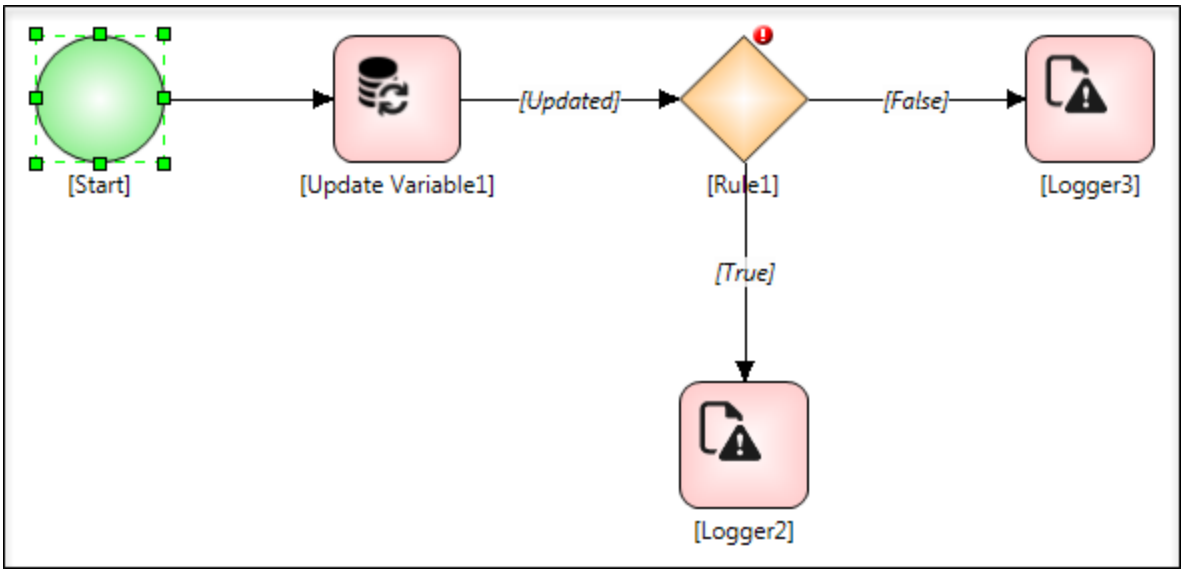
True: Displayed when the rule condition is true

False: Displayed when the rule condition is false

Example for a Rule Activity

Step 1: Design a workflow as shown in the following figure.

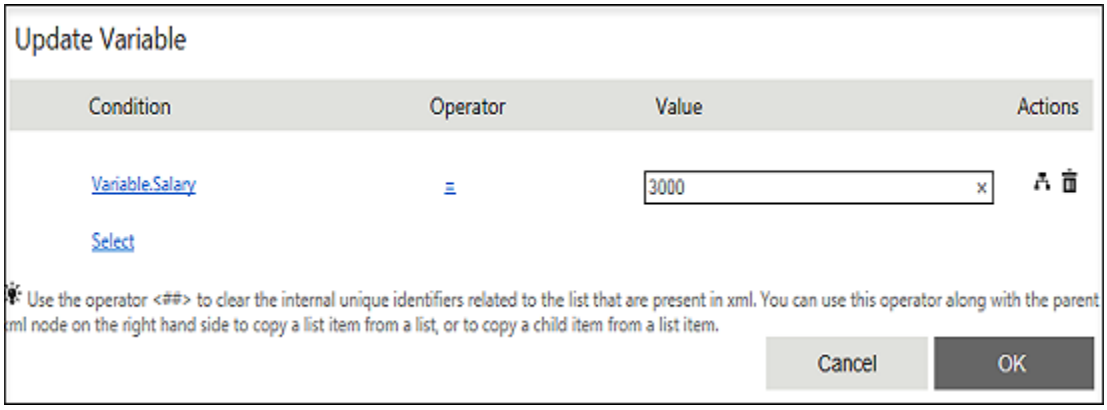
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Step 2: In the Start Activity declare a variable Salary.

Step 3: In the 'Set Variable' property of Update variable activity, perform the following actions:

- In the first drop-down, select Variable.Salary.
- Type a value. For Example, 3000.
- Click **OK**.



Step 4: In the Condition property of Rule Activity, perform the following actions:

- In the first drop-down, select Variable.Salary.
- Second drop-down, select '>' .
- Third drop-down, select 'Value'.
- In the adjacent text box, enter a variable. For Example, 2000.
- In the last drop-down, select AND.
- Click **Add**.
- Once again, in the first drop-down, select Variable.Salary.
- Second drop-down, select '<'.
- Third drop-down, select 'Value'.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- In the adjacent text box, enter a variable. For Example, 6000.
- In the last drop-down, select END.
- Click **Add**.
- Click **Update**. The entry would look as the following:

[Variable.Salary]>2000 And Variable.Salary<6000

Step 5: On the link connecting the right logger, select False and display an error message.

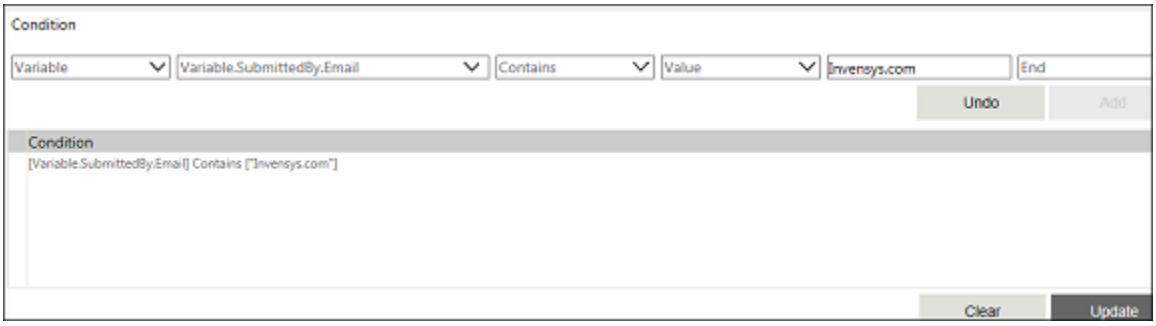
Step 6: On the link connecting the left logger, select True and display the value of the variable.Salary.

Output: The Variable.Salary will be updated to 3000, and in the rule condition the condition will be checked. The rule activity will return true and execute the logger at the bottom of the rule activity displaying the value of the variable Salary in the Logger Console.

Rule - Condition

Condition Property Window

You can set the condition corresponding to the Rule activity in this property window. To open this window, click the button for the **'Condition'** property.



The window has a set of drop downs at the top to select the different parameters required to build an expression for the condition. You can also find related buttons next to the drop downs.

- The first Value type drop down - You can select the type of value that will form the left-hand side the rule condition expression. You can select **Variable**, **Content**, or **XMLVariables** from this drop down.
- The first Value drop down - You can specify the actual value corresponding to the value type selected in the preceding drop down. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Operator drop down - You can select the operator for the expression. E.g., **=**, **<=**, **IN**, **Contains**.
- The second Value type drop down - You can select the type of value to be matched with the value specified in the left-hand side of the expression. You can specify **Variable**, **Content**, **Value**, or **XMLVariables**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The Value text box/ drop down - You can specify the actual value to assign to the attribute. The control displayed depends on the option selected in the preceding drop down. If you select **Value**, it displays a text box. You can enter a discrete value in this box to complete the expression. If you select **Variable**, **Content**, or **XMLVariables** instead, there will be a drop down instead of the field. This drop down will have the variable collection or content collection or XMLVariables collection from which you can select an appropriate option.
- The Logical operator drop down - You can select logical operators from this drop down to build compound expressions. The options available are **End**, **And**, **Or**, **)**, **(**. You can select **End** to complete the expression.
- **Add** - Click this button to add the new expression to the text area.
- **Undo** - Click this button to remove a condition you have added to the expression.

At the bottom of the window you will find the following buttons:

- **Clear** - Click this button to clear the expression shown in the text area. This is useful when you want to build a new expression.
- **Update** - Click this button to save the expression that has been added to the text area.

Set Workflow Output Variable

Activity Description

The Set Workflow output variable activity can be used to set the output of the parent workflow from the child workflow. You can use Set Workflow output variable of Quickflow to update a Workflow only when both the parent and child are of the same type (either Workflow or Quickflow).

Activity Properties

You can configure the Set Workflow output variable activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **Set Workflow output variable** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

Name & Description

You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

You can use this property to configure the functionality of the activity.

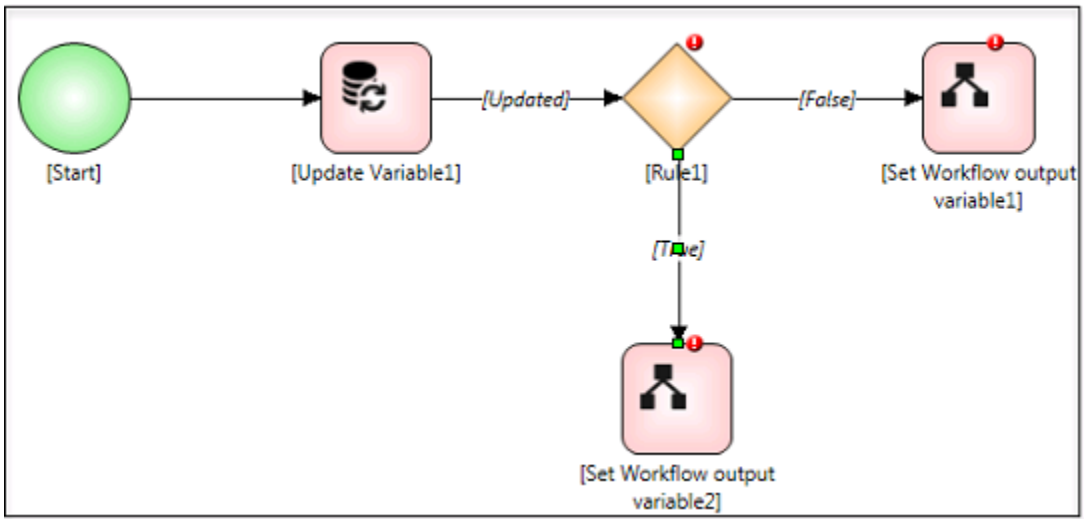
Set Workflow Output - This property specifies the output of the workflow. You can specify the output value directly here or click the icon to open the Expression Builder and build an expression for the output.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

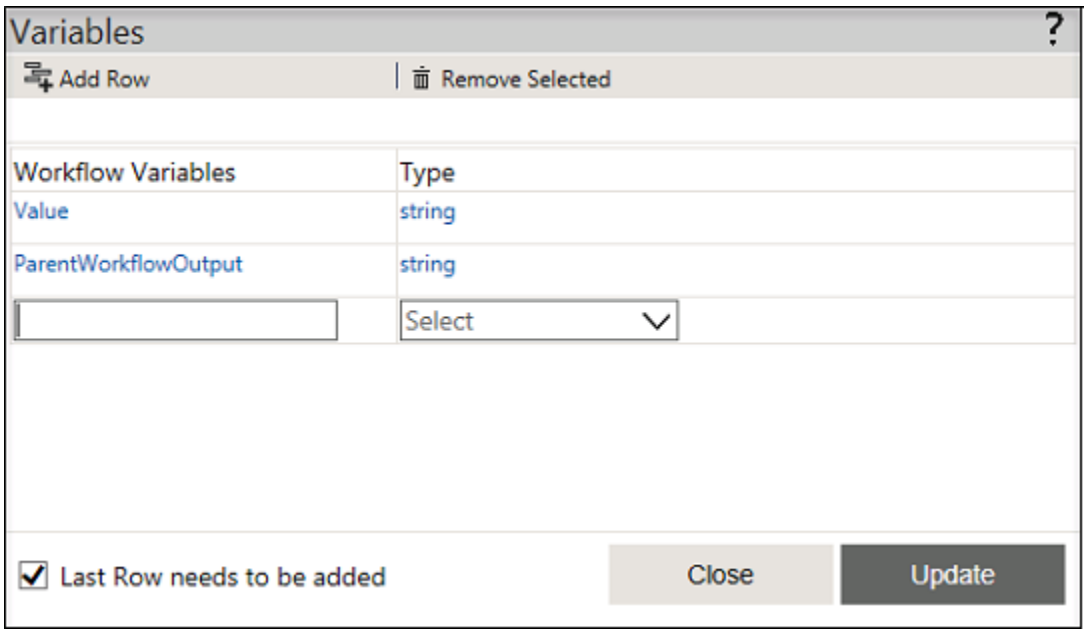
Example for setting Workflow Output Variable

1. Design a Quickflow as shown in the below figure and name it as ChildQF.



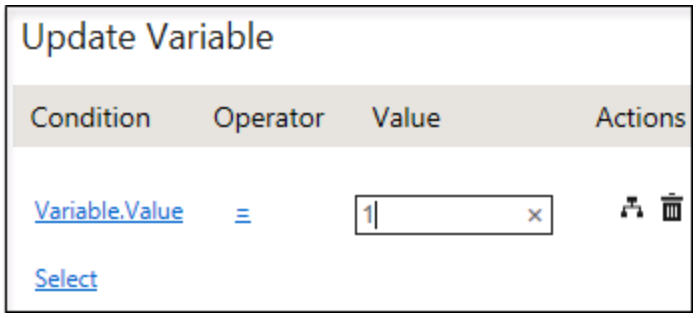
2. Perform the following actions:
   a. In the Start Activity, declare two variables Value and ParentWorkflowOutput as String.

   Note: The user must declare a same variable name in the child workflow which was declared for the Parent workflow. In the above example, user must declare by the name "ParentWorkflowOutput" of type "String" in the child workflow.



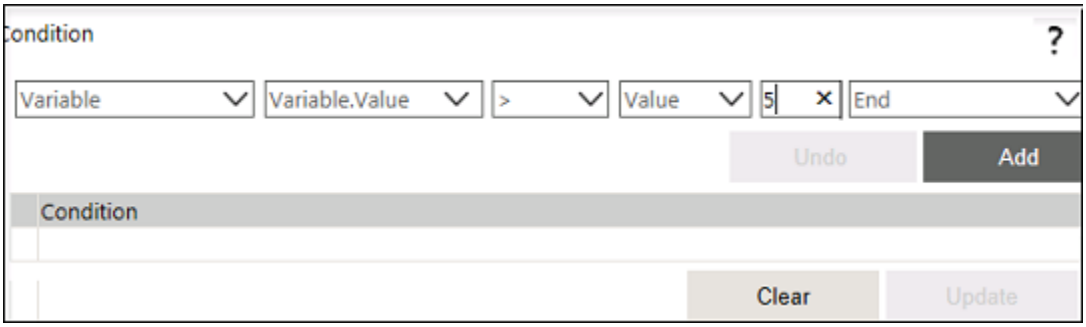
   a. In the Set Variable property of Update Variable set the value of variable Value as 1 as shown in the figure.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



b. On the link connecting Update Variable and Rule activity set the output as Updated.

c. In the Configuration property of Rule, set the condition [Variable.Value]>["5"] as shown in the below figure.



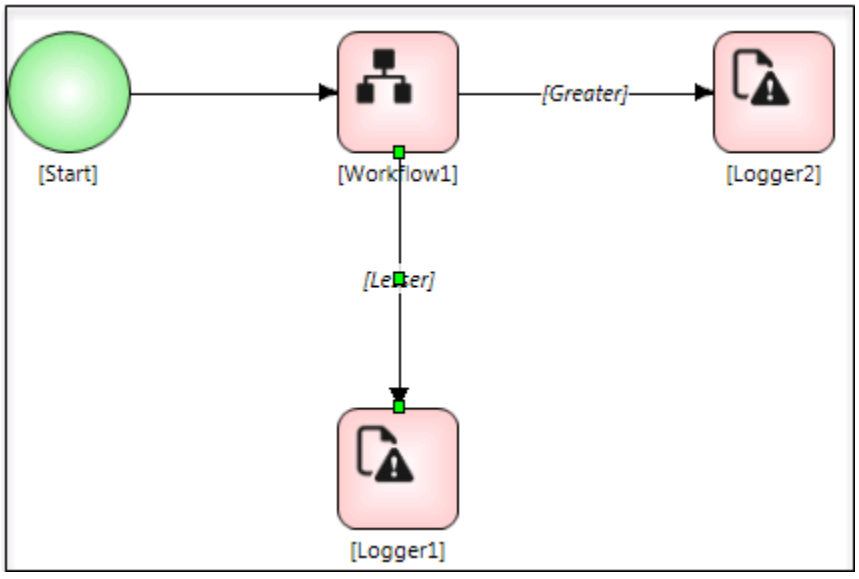
d. On the link connecting **Rule** and **Set Workflow output variable2** Activity set TRUE.

e. In the **Set Workflow Output** property of **Set Workflow output variable2** Activity set it as Greater.

f. On the link connecting Rule and **Set Workflow output variable1** Activity set FALSE.

g. In the **Set Workflow Output** property of **Set Workflow output variable1** Activity set it as Lesser.

3. Design a Quickflow as shown in the below figure and name it as ParentQuickflow.



4. Perform the following actions:

a. In the Repository property of Workflow Activity, enter Repository Name. For Example: SamRep.

b. In the Workflow property of Workflow Activity, select Previously created Workflow Name. For Example: ChildQuickflow.

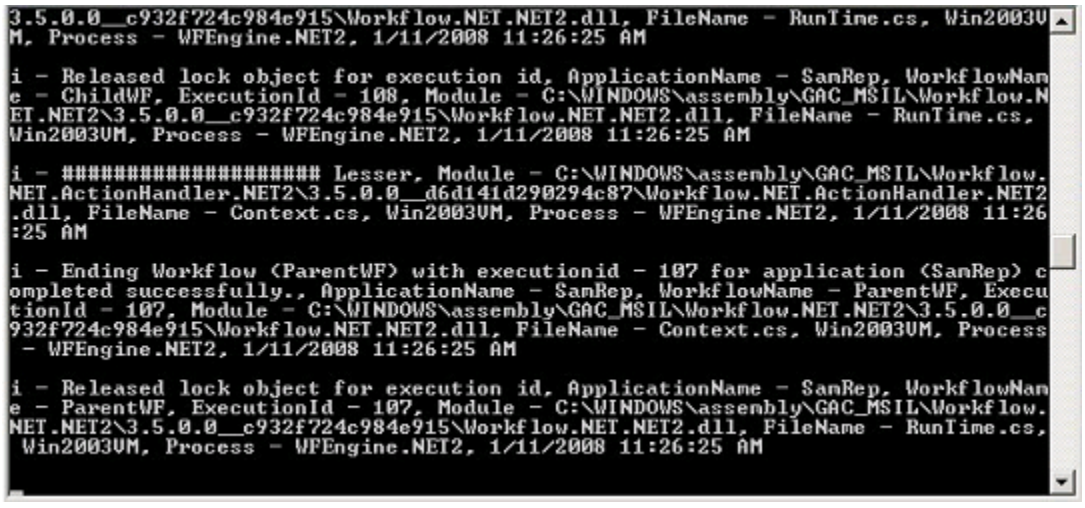
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

c. In the Wait for the Child Workflow property, select "YES".
d. In the Execution Mode property, select "Asynchronous".
e. In the Return Values property of Workflow Activity:
    a. Click Add Row and enter as "Greater"
    b. Click Add Row and enter as "Lesser"
    c. Click Update to save and click Close to close the window.
f. In the property LogType of Logger1, select Information.
g. In the Log Message of Logger1, enter as the following:
   '################ Lesser'
h. In the link connecting Logger1 and Workflow Activity1 set Lesser as output.
i. In the property LogType of Logger2, select Information.
j. In the Log Message of Logger2, enter as the following:
   '################ Greater'
k. In the link connecting Logger2 and Workflow Activity1, set Greater as output.

Summary: When you publish the parent workflow, the child workflow is also triggered and the path selected after the Rule activity is stored in the variable "ParentWorklfowOutput". The output of child workflow can be traced in the parent workflow.

Output: The ParentQF is published, and the workflow1 Activity invokes the ChildQF. Value of the variable Value is set to 1 in Update Variable Activity.

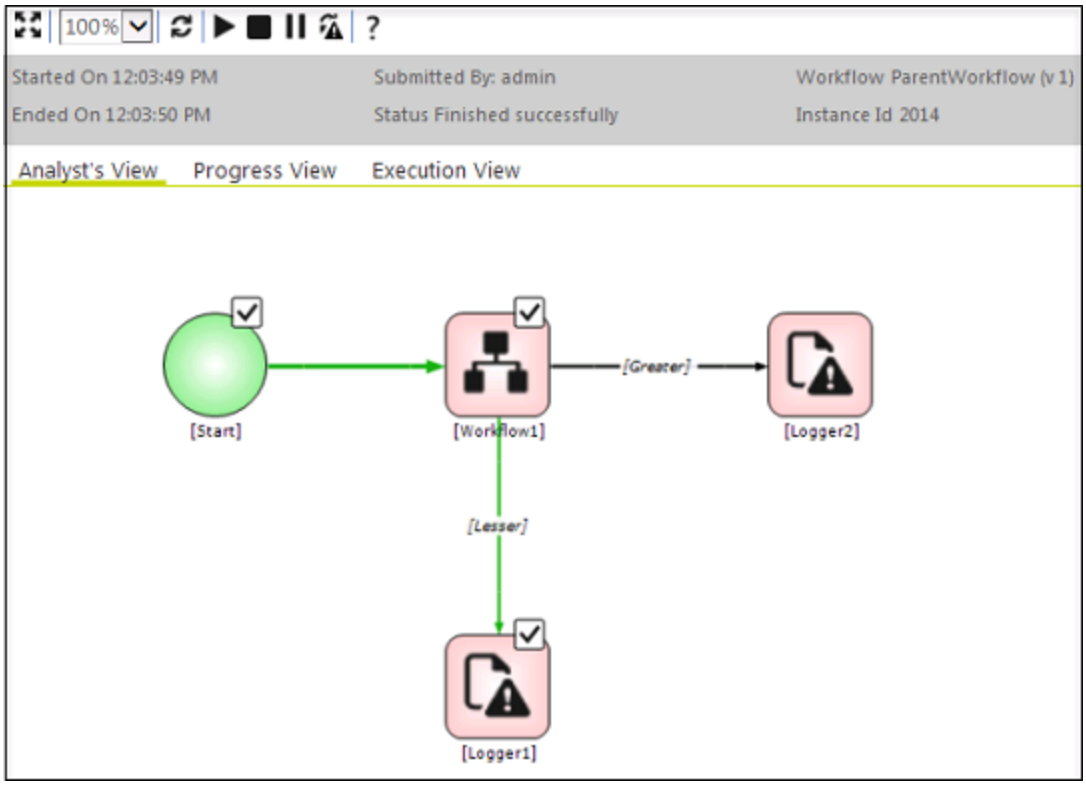
Since the value of Variable value is less then five, the rule activity output is False. So the Set Workflow Output Variable2 is executed and the control is passed to the ParentQF. The value updated in Parent Workflow variable is Lesser and hence the output of the Workflow Activity is Lesser and the Logger1 is executed.



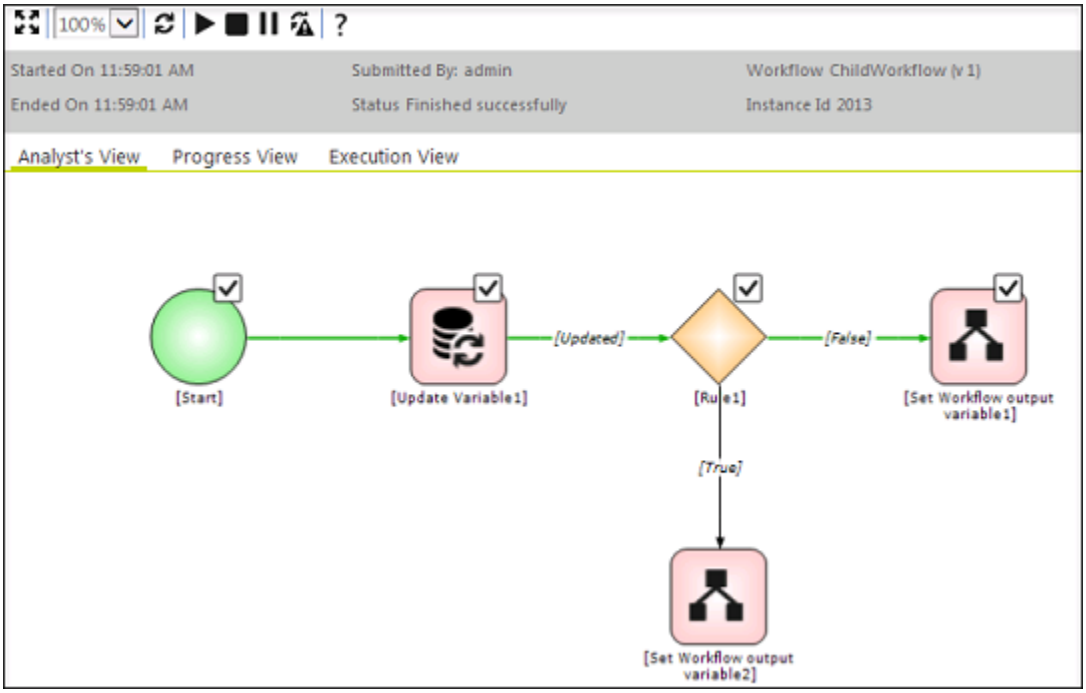
Start Activity in BAM

Parent Quickflow in BAM:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Child Quickflow in BAM



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Synchronize Activity

Activity description

The Synchronize Activity synchronizes the execution of different activities. The engine does not proceed with the execution of further actions until all the actions leading to this activity are completed.

Activity Properties

The Synchronize activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Synchronize activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

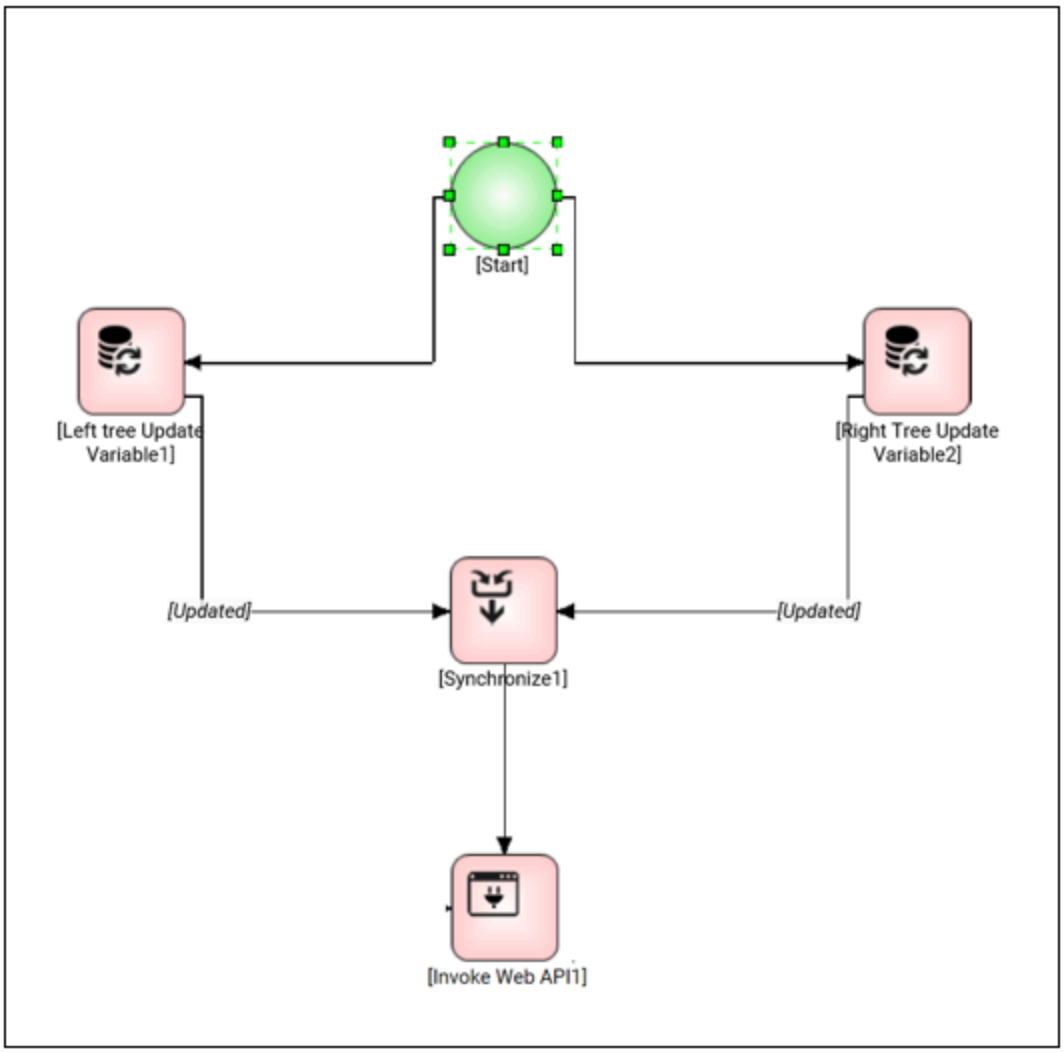
Activity outputs

The Synchronize Activity does not return any value.

Example for a Synchronize Activity:

Suppose you want to submit the updated value of the two variables which are executed at different branches to the WebAPI, then these two values has to be synchronized. In the following workflow, the two variables are connected to the synchronize activity. When both the variables are updated using the update variable activity, the value gets submitted to WebAPI only after the synchronize activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Update Parent Variables Activity

Activity description

In AVEVA Work Tasks, a parent workflow variable can be updated from a child workflow provided both child and parent workflow belong to the same repository. This can be achieved by using Update Parent variables activity.

Activity Properties

You can configure the Update Parent Variables activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **Update Parent Variables** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

Name & Description

Use these properties to specify the name and description for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

Use these properties to configure the functionality of the activity.

Parent Workflow Application Name - This property specifies the name of the application to which the parent workflow belongs. You can specify the application name directly here or click on the icon to open the Expression Builder and build an expression for the application name.

The parent workflow should belong to the same repository of the child workflow. This property of a child workflow is exposed for backward compatibility.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Parent Workflow Name - This property specifies the name of the parent workflow. You can specify the workflow name directly here or click on the icon to open the Expression Builder and build an expression for the workflow name.

The parent workflow should belong to the same repository of the child workflow.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

List Parent Variables - This property specifies the parent workflow variables to be updated. Click on the edit icon to open the

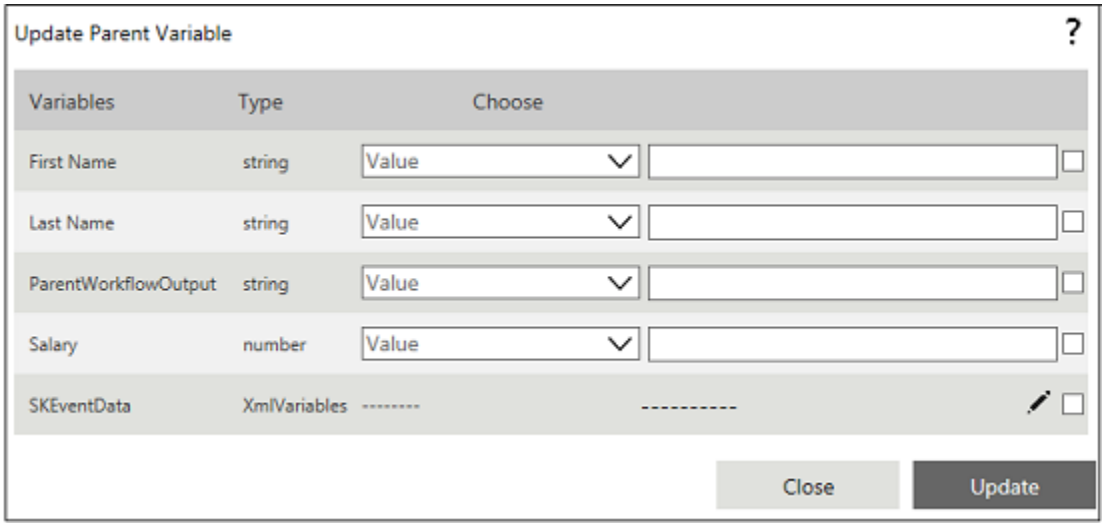
'List Parent Variables' property window.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Set the List Parent Variables property

To update the value of variable:

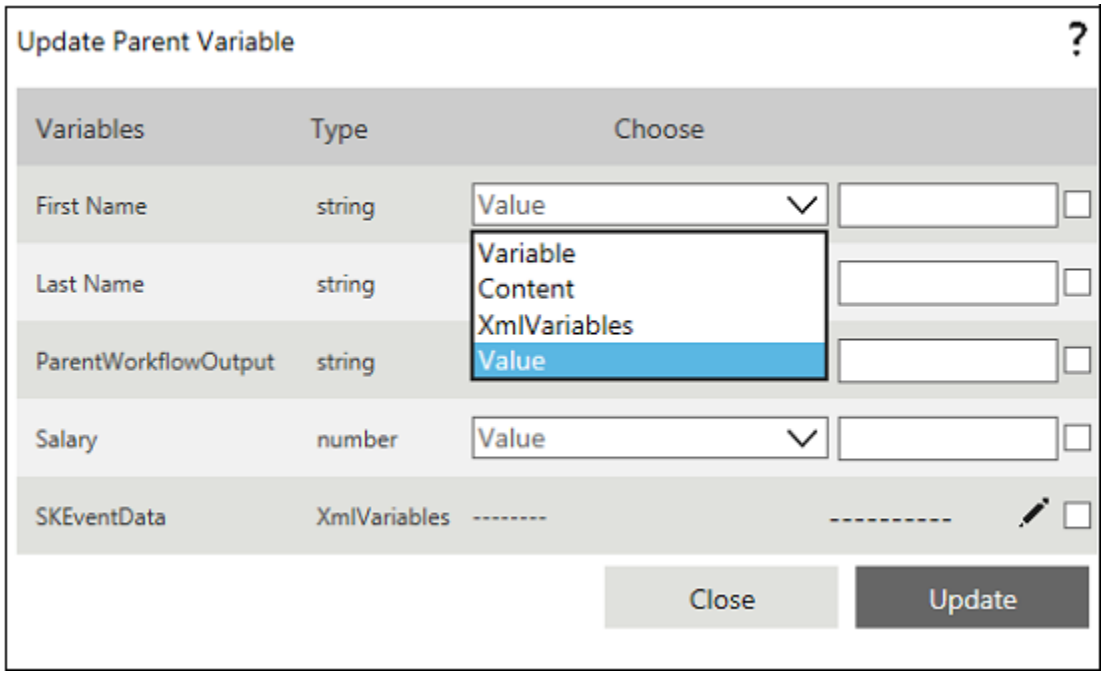
1. Click the edit button of the **List Parent Variables** property. The **Update Parent Variables** window appears. Specify the parent workflow variables in this window.
2. Select the check box of the respective parent variable.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Choose the option using which the parent variable to be updated. The parent variable values can be updated using the Value, Variable, Content, or XML Variable of the child workflow.
4. Depending on the choice, select the respective options or node, and then click **Update** to save.

Note: The value can also be hardcode using the Value option as shown in the following image.

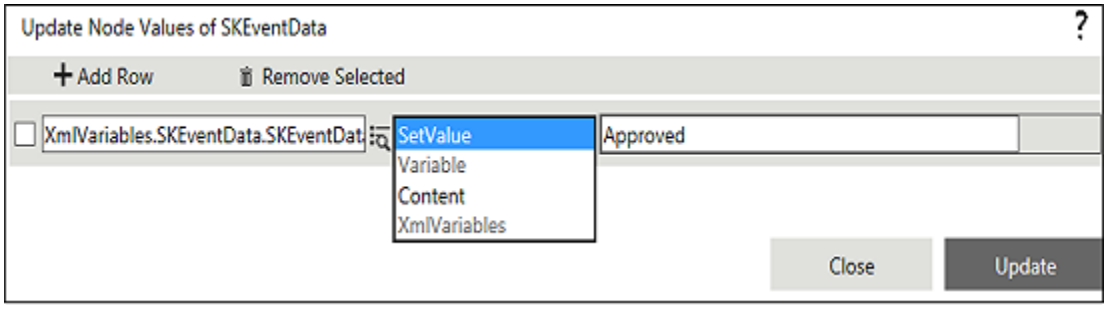


5. After selecting and assigning values to the variables, click **Update** to save.

To set the value of XML Variable

1. Select the check box of the respective parent variable.
2. Clicking the icon on the left hand side of the check box opens the window to specify the node to be updated of the respective parent XMLVariable.
3. Select the node to be updated from the tree view.
4. Choose the option using which the parent XML variable node to be updated. The parent XML variable node value can be updated from the variable, content or XML variable of the child workflow.
5. Depending on the choice select the respective option/node from the RHS.

   Note: The value can also be hardcoded using the SetValue option as shown in the following figure.



6. You can specify multiple nodes for updating by using the **Add Row** option.
7. In order to remove any node that is set from updating, select the respective entry by selecting the check box

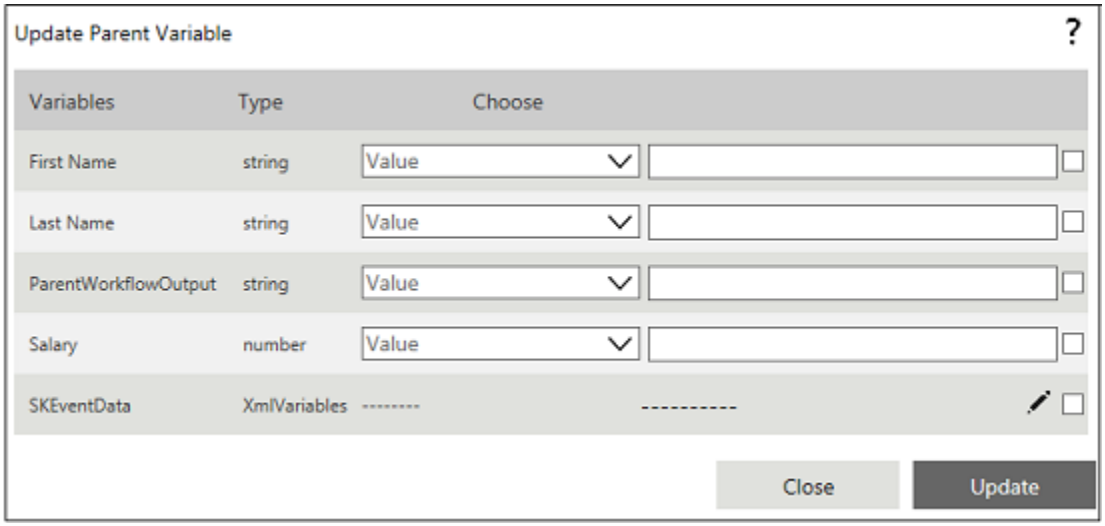
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

in the left hand side and click **Remove Selected**.

See Update Parent Variables - List Parent Variables for a detailed description of the List Parent Variables property window.

Update Parent Variables - List Parent Variables

You can set the List Parent Variables for this activity in the **Update Parent Variable** property window.



Update Parent Variable property Window

The **Update Parent Variable** property window has a grid that lists all the variables belonging to the parent workflow which are specified in the Parent Workflow Name property. Each row in the grid represents a parent workflow variable, showing the variable name and type under the respective columns. The window also contains the following controls to specify the value to update the variable with:

- The Value type drop-down menu: Select the type of value to be assigned to the parent variable. Values that you can specify are Variable, Content, Value, or XMLVariables.
- The Value text box/drop-down menu: Specify the actual value to assign to the parent variable. The control displayed depends on the option selected in the Value type drop-down menu. If you select "Value", a text box appears. You can enter a discrete value in this box to complete the expression. If you select "Variable", "Content", or "XMLVariables", a drop-down menu appears in this field. This drop-down menu contains variable collection, content collection, or XMLVariables collection from which you can select an appropriate option.
- The check box: Select this check box to update the value of the corresponding parent variable at the workflow execution time.
- To set values for an XML Variable, click the icon, and then enter the required values in the **Update Node Values** window. For more information, see How to Update a Parent Workflow Variable from a Child Workflow?.

Note:
- For number data types, the minimum and maximum values are not validated. Enter only valid numeric values for number data types. For numeric, DateTime, Date, or Time data types, ensure that the respective

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

value is based on the culture set. You can enter the input or use the Date and Time picker to enter the required date and time. The time format supports long data type to enable you to enter seconds. If you enter a valid value for a numeric, date, time, or datetime data type, and if you enter unwanted strings after the valid values, then while submitting the values, the extra strings at the end are truncated. For example, for a DateTime value entered as 01/31/2015 10:11:12ababa, then ababa is truncated by default.
- Only single instance of Update Parent Variables activity can be used in child Quickflow.

How to Update a Parent Workflow Variable from a Child Workflow?

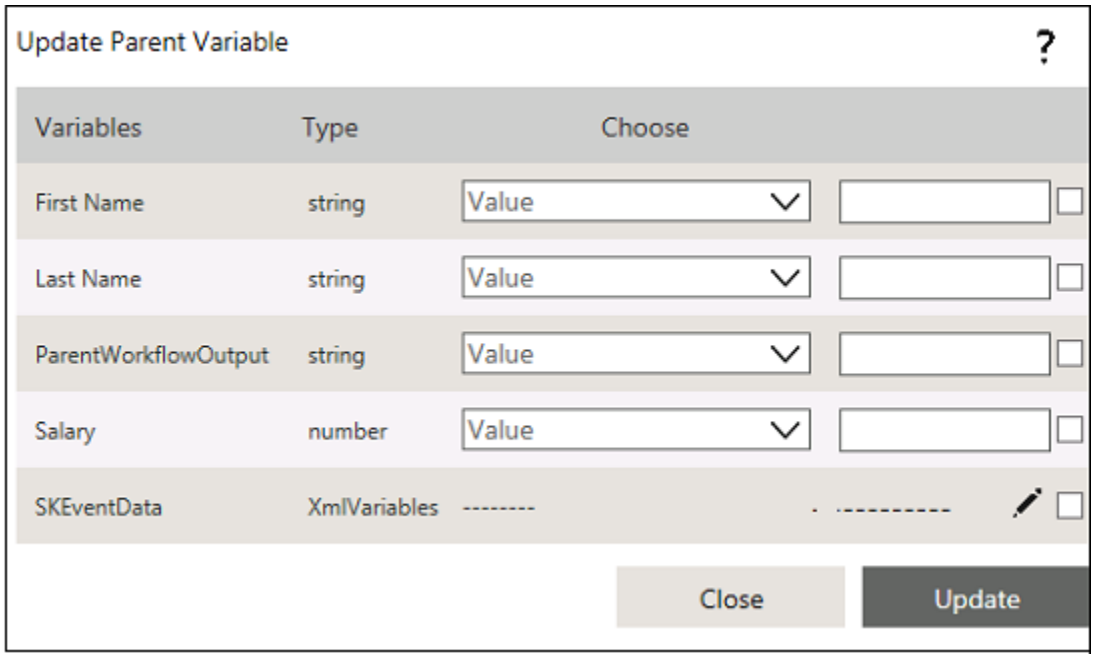
In AVEVA Work Tasks, a parent workflow variable can be updated from a child workflow provided both child and parent workflow belong to the same repository. This can be achieved by using **Update Parent Variables** activity.

To update parent workflow from a child workflow

1. Right-click **Update Parent Variables Activity** and select **Activity Properties**.
2. Enter the parent workflow's repository name in the **Parent Workflow Application Name** property. The parent workflow should belong to the same repository of the child workflow. This property of the Update Parent Variable activity is exposed for backward compatibility.
3. Enter the parent workflow name in the **Parent Workflow Name** property. The parent workflow should belong to the same repository of the child workflow.
4. Click **List Parent Variables** property to update the variables in the parent workflow.

To update the value of variable

1. To specify the parent workflow variables, click the button for the **'List Parent Variables'** property to open the property window.
2. In the **Update Parent Variables** window, select the check box of the respective parent variable.

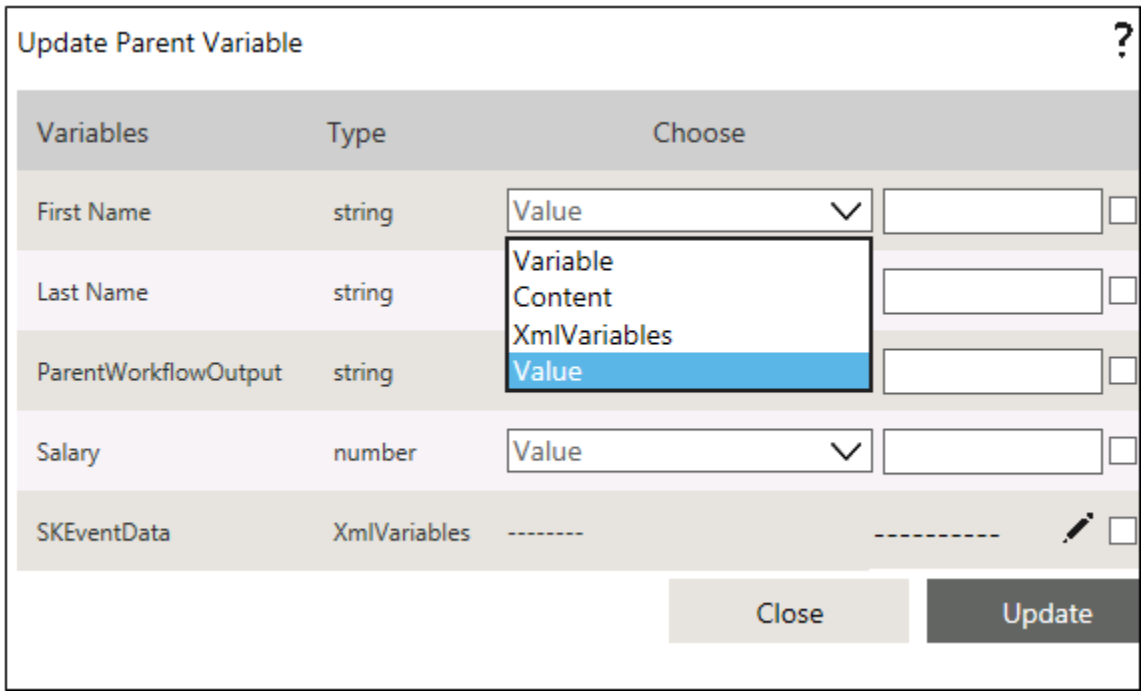


3. Choose the specific option from the list using which the parent variable to be updated. The parent variable value can be updated from the Value, Variable, Content, or XML Variable of the child workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Depending on the choice, select the respective text box/options/node from the RHS and click **Update** to save.

Note: The value can also be hard coded using the Value option as shown in the following figure.

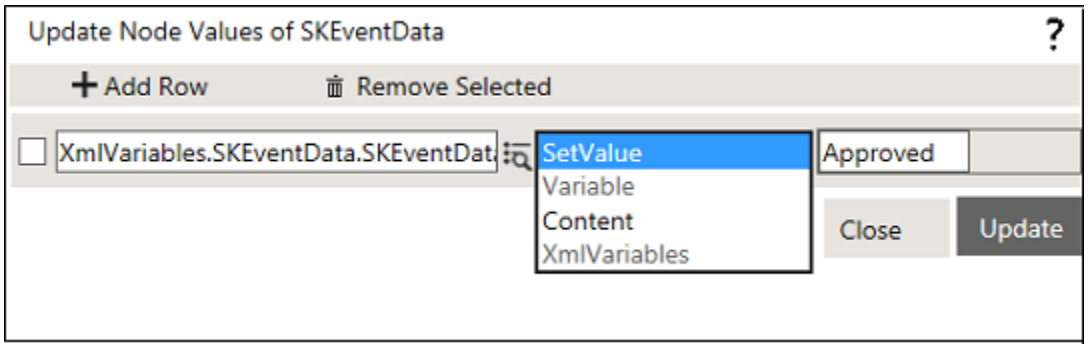


5. After selecting and assigning values to the variables, click Update to save.

To set the value of XML Variable

1. Select the check box of the respective parent variable.
2. Click the icon on the left hand side of the check box. The **Update Node Values** window appears. Specify the node to be updated for the respective parent XML Variable.
3. Select the node to be updated from the tree view.
4. Choose the option using which the parent XML variable node to be updated. The parent XML variable node value can be updated from the SetValue, Variable, Content, or XMLVariable of the child workflow. Depending on the choice, select the respective value/options/node from the RHS.

   Note: The value can also be hard coded using the SetValue option as shown in the following figure.



5. You can specify multiple nodes for updating by using the **Add Row** option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. To remove any node that is set from updating, click the respective check box of the entry to be deleted, and then click **Remove Selected**.

Update Variable Activity

Activity Description

The Update Variable activity can be used to update the value of Variables and XmlVariable that can be used in the subsequent activities. Using this activity, the variable for For-Loop activity can also be initialized.

Activity Properties

You can configure the Update Variable activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **Update Variable** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The property in the Configuration group has been described below. You can use these properties to configure the functionality of the activity.

Set Variable - This property allows the user to specify the values for existing Variables or XMLVariables.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

See Variables for updating Variables.

See Forms Xml Variables to work with Forms XMLVariables.

Workflow Activity

Activity Description:

In AVEVA Work Tasks, a workflow can be called from another workflow provided both workflows belong to the same repository. In this case, the calling workflow is termed as **Parent Workflow** and the workflow to be called is termed as **Child Workflow**.

The Workflow activity enables the execution of a second (child) workflow from a (parent) workflow. The

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Workflow activity can be used to call a child workflow from a parent workflow. The **Variables for the child workflow** property and **XMLVariables for the child workflow** property enable parent workflow to pass variable and XML variable information to the child workflow respectively.

Activity Properties:

You can configure the Workflow activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **Workflow** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The properties in the Configuration group have been described below. You can use these properties to configure the functionality of the activity.

Repository - This property specifies the application or repository name that has the workflow to be run. The child workflow should belong to the same repository of the parent workflow. The repository property of a child workflow is exposed for backward compatibility.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Workflow - This property specifies the workflow to execute. This is an editable combo, which drops down a list with the names of all available workflows for the specified application. You can select one of the workflows as the child workflow. Alternatively, you can enter a name in the field to create a new child workflow. Now when, you right-click the workflow activity icon and select the "Open Sub Process" option, a new Process Designer window will be opened with the Application name and Workflow name specified here. You can create the new sub-process in this window.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Version - This property specifies the version only if any specific version of the (child) workflow to be executed. Leave this as blank, if the latest published version of (child) workflow has to be executed.

Property Type: Optional (This property needs to be set only if required.)

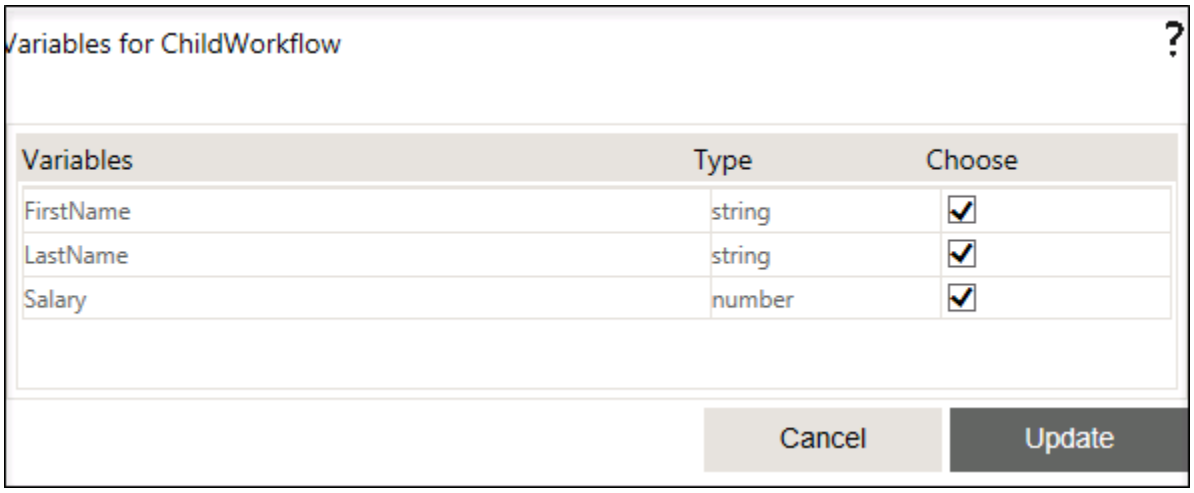
Variables for ChildWorkflow - This property specifies the variables to be passed on to the workflow. This window shows the list of variables defined for the current (parent) workflow. Select the variables that need to be passed to the child workflow.

Property Type: Optional (This property needs to be set only if required.)

Steps to set the Variables for ChildWorkflow property

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. To specify the variables to be passed to the child workflow, click the button for the **Variables for ChildWorkflow** property to open the property window.
2. In the **Variables for ChildWorkflow** window, select the required variables to be passed to the child workflow.



3. After specifying the data click the **Update** button.

   Note: These variables are expected to exist in the child workflow. Make sure to define variables with same name and type as defined in parent workflow.

   See Workflow - Variables for ChildWorkflow for a detailed description of the Variables for ChildWorkflow property window.

XMLVariables for ChildWorkflow - This property specifies the XML variables to be passed on to the child workflow. This window shows the list of XML variables defined for the current (parent) workflow. Select the XML variables that need to be passed to the child workflow.

Property Type: Optional (This property needs to be set only if required.)

Steps to set the XMLVariables for ChildWorkflow property:

1. To specify the XML variables to be passed to the child workflow, click the button for the 'XMLVariables for ChildWorkflow' property to open the property window.
2. In the XMLVariables for ChildWorkflow window, choose the required variables to be passed to the child workflow.
3. Click the **Update** button.

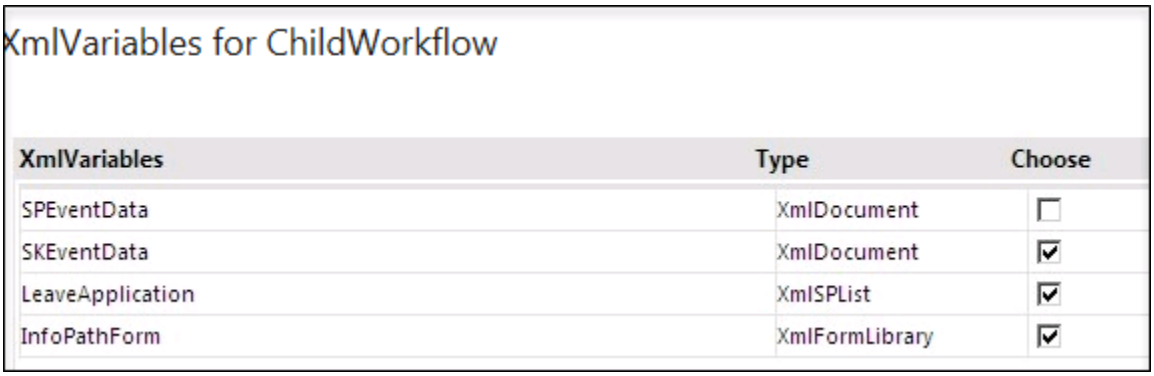
Note:
- These XML variables are expected to exist in the child workflow. Make sure to define XML variables with same name and type as defined in parent workflow.

- To work with any XML variables or resource activities, SKEventData XML variables must be passed in the child workflow, where SKEventData contains some events information related to parent workflow which needs to be passed to child workflow. If SKEventData variable is not created, user can create XML variable of type XMLDocument with the same name 'SKEventData' by locating the 'SKEventData.xml' from the following path:*[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\BPMUITemplates\Default\XMLSchemas*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example:

The following image shows that XMLVariables named SPEventData and LeaveApplication need to be passed to the child workflow. In this case, the child workflow is expected to have SPEventData and LeaveApplication XML variables defined.



See Workflow - Workflow - XmlVariables for ChildWorkflow for a detailed description of the XMLVariables for ChildWorkflow property window.

Output Values

The property in the Output Values has been described below. You can use this property to specify the output values of the activity.

Return Values - This property can be used to add all the possible output values from the child workflow, for conditional routing. The child workflow should set a value to the 'ParentWorkflowOutput' variable. This variable will hold the output value from the child workflow. The value of the 'Wait for the child workflow?' property should be set to 'Yes'.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Return Values property

1. To specify the Return Values, click the button for the 'Return Values' property to open the property window.
2. In this window, enter the return value in the Return Values field.
3. After entering the return value details, click Add Row to add this return value and display a new row for entering another return value.
4. Remove rows if necessary by selecting them and clicking Remove Selected.
5. After specifying all the return values, click Update.

See Workflow - Return Values for a detailed description of the Return Values property window.

Activity outputs:

The Workflow Activity does not have any standard return value. It will output the value specified by the user in the Return Values property.

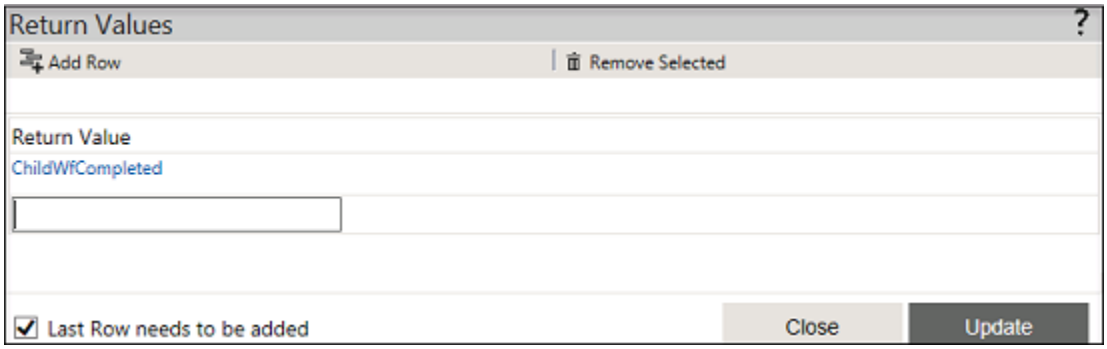
Example for a Workflow Activity:

Configuring Parent Workflow to wait till Child Workflow Completes Execution

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to be followed for Parent Workflow configuration:

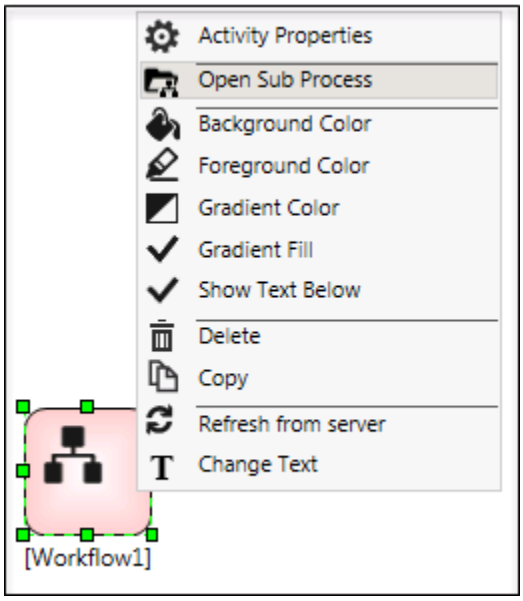
1. Set Wait for the child workflow property as Yes to wait for the child workflow to complete the execution.
2. Use Return Value property to set the routing condition after completion of child workflow. Example: ChildWfCompleted.



This will dynamically create a link output named 'ChildWfCompleted' for the activity. Once the child workflow complete its execution, it returns back to the parent workflow with the return value specified in the child workflow (In this case, child workflow should return 'ChildWfCompleted' value) and then it will continue with further execution.

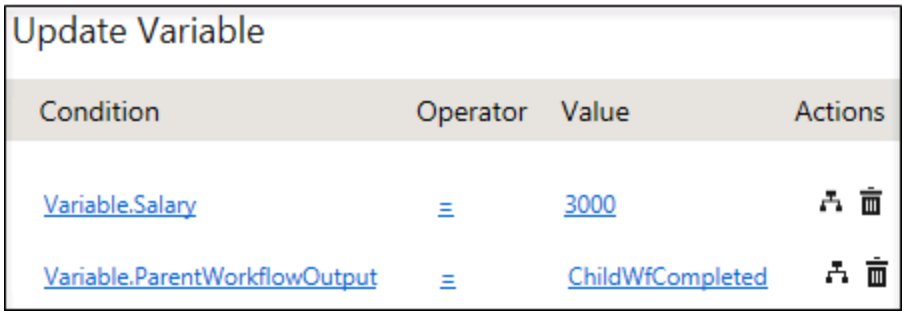
Steps to be followed for Child Workflow configuration:

1. Right-click the workflow activity to open child workflow using the context menu item Open Sub Process.



2. Declare a variable named 'ParentWorkflowOutput' of type string in the child workflow Start action.
3. Use Update Variable action to set the variable 'ParentWorkflowOutput' value as 'ChildWfCompleted' (This value should be same as the value for the Return Values property in parent workflow action properties).
4. The Update variable action to update the parent workflow output should be placed after the logical conclusion of child workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



After configuring child workflow, follow the steps given below:

1. Go back to the parent and set routing condition based on the output 'ChildWfCompleted'
   a. Parent workflow goes to sleep mode after the child is executed.
   b. Child completes execution with Update Variable as the last action.
   c. Once the child completes execution, the value set in Update variable is given as output for the workflow action in parent.

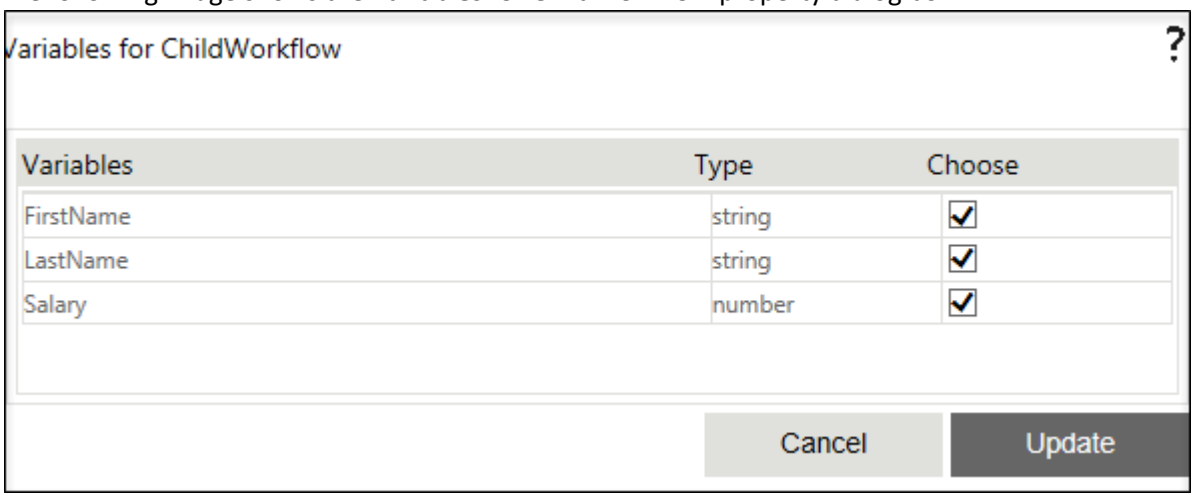
Note: If the child workflow needs to update any of the parent workflow variables, it is a must for Parent workflow to wait for the child workflow to complete.

Workflow - Variables for ChildWorkflow

Variables for ChildWorkflow Property Window

You can specify the variables to be passed to the child workflow in this property window. To open this window, click the button for the 'Variables for ChildWorkflow' property.

The following image shows the **Variables for ChildWorkflow** property dialog box:



The window has a grid that lists the parent workflow variables that can be passed to the child workflow. It has the following columns – **Variables**, **Type**, and **Choose**. Select the **Choose** check box to select the variable. Click **Update** to set the selected variables to be passed.

Note: You can create a local variable (that is, variables prefixed with '@', for example, @EmpId) which can be passed to the child workflow. For the child workflow to get this variable value, it should have a variable with the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

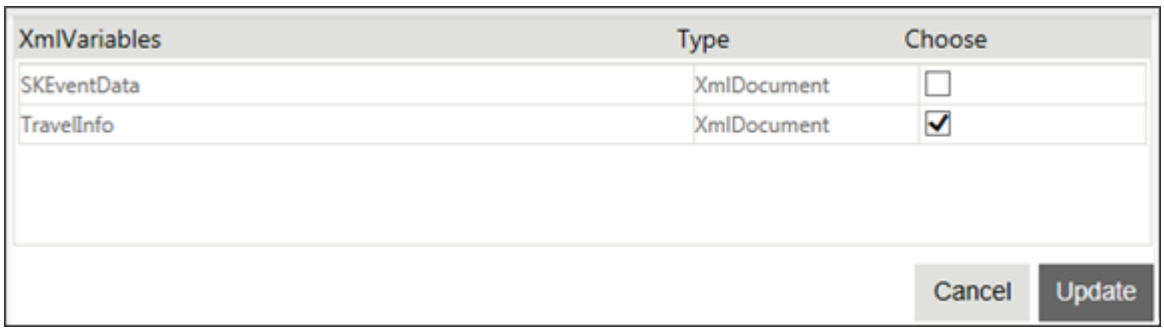
same name but without the '@'. For example, if a parent workflow has a local variable '@EmpId', then the child workflow should have a variable 'EmpId' to get its value.

Workflow - XmlVariables for ChildWorkflow

XmlVariables for ChildWorkflow Property Window

You can specify the XmlVariables to be passed to the child workflow in this property window. To open this window, click the button for the 'XmlVariables for ChildWorkflow' property.

The following image shows the **XmlVariables for ChildWorkflow** property dialog box:



The window has a grid that lists the parent workflow XMLvariables that can be passed to the child workflow. It has the following columns –**XmlVariables**, **Type**, and **Choose**. Select the **Choose** check box to select the XmlVariable(s). Click **Update** to set the selected XmlVariable(s) to be passed.

Note: For **XMLVariables** of type **XMLDocument**, the **Internal Storage** should be the same for both the parent and the child workflow.

How to Pass Variables from Parent Workflow/Quickflow to Child Workflow?

The Workflow Activity enables the execution of a second (child) workflow from a (parent) workflow/quickflow. The workflow activity can be used to call a child workflow from a parent workflow/quickflow. The **Variables for the child workflow** property and **XMLVariables for the child workflow** property enable parent workflow/quickflow to pass variable and XML variable information to the child workflow respectively.

To pass variables from parent workflow to child workflow, follow the steps given below:

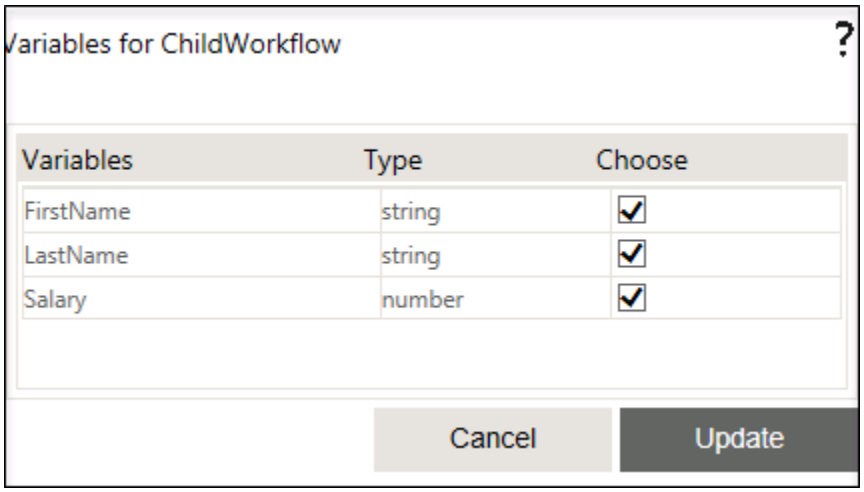
1. In the parent workflow/quickflow, right-click **Workflow Activity**, and then select **Activity Properties**.
2. Enter the repository name for **Repository** property. The child workflow should belong to the same repository of the parent workflow/quickflow. The repository property of a child workflow is exposed for backward compatibility.
3. Enter the name of the workflow to be executed in the **Workflow** property.
4. Enter the version only if any specific version of the (child) workflow to be executed in the Version property.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Leave this as blank, if the latest published version of (child) workflow has to be executed.

Steps to set the Variables for ChildWorkflow property:

1. To specify the variables to be passed to the child workflow, click the button for the 'Variables for ChildWorkflow' property to open the property window.
2. In the **Variables for ChildWorkflow** window, select the required variables to be passed to the child workflow.



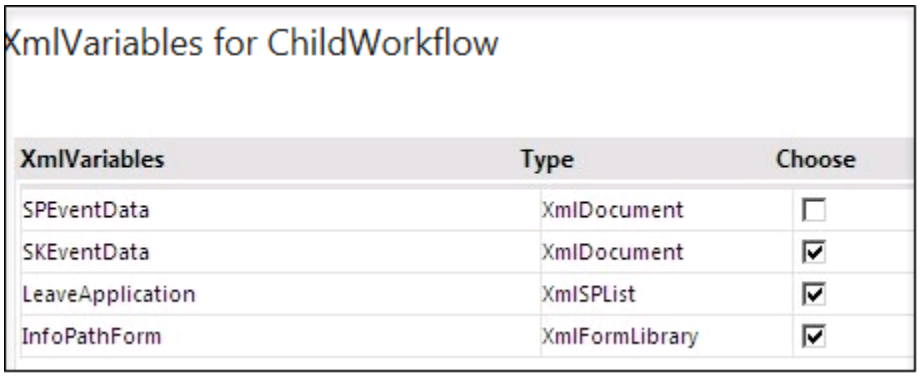
3. After specifying the data, click Update.

Note: These variables are expected to exist in the child workflow. Make sure to define variables with same name and type as defined in parent workflow/quickflow.

Select the XML variables that need to be passed to the child workflow from XMLVariables for ChildWorkflow property.

Steps to set the XMLVariables for ChildWorkflow property:

1. To specify the XML variables to be passed to the child workflow, click the button for the **'XMLVariables for ChildWorkflow'** property to open the property window.
2. In the **XMLVariables for ChildWorkflow** window, choose the required variables to be passed to the child workflow.



3. Click the Update button.

Note: These XML variables are expected to exist in the child workflow. Make sure to define XML variables with same name and type as defined in parent workflow/quickflow.

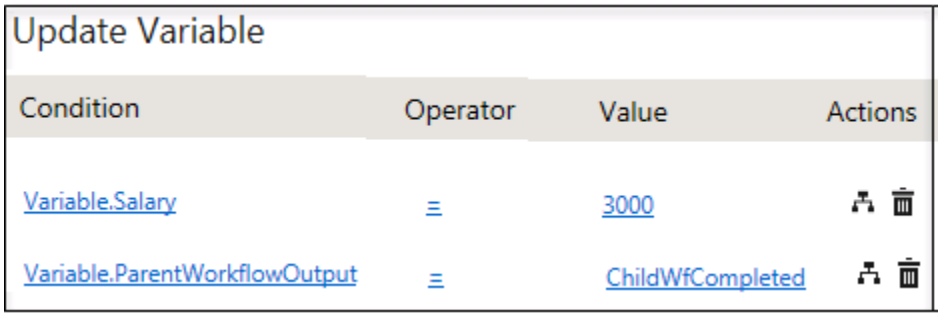
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To work with any XML variables or resource activities, SKEventData XML variables must be passed in the child workflow, where SKEventData contains some events information related to parent workflow/quickflow which needs to be passed to child workflow. If SKEventData variable is not created, user can create XML variable of type XMLDocument with the same name 'SKEventData' by locating the 'SKEventData.xml' from the following path: *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\BPMUITemplates\Default\XMLSchemas*

Configuring Child Workflow to route Parent Workflow/Quickflow

Steps to be followed for Child Workflow configuration:

1. Right-click the workflow activity to open child workflow using the context menu item Open Sub Process.
2. Declare a variable named 'ParentWorkflowOutput' of type string in the child workflow Start activity.
3. Use **Update Variable** activity to set the variable 'ParentWorkflowOutput' value as 'ChildWfCompleted' (This value should be same as the value for the Return Values property in parent workflow action properties).
4. The **Update Variable** activity to update the parent workflow output should be placed after the logical conclusion of child workflow.



Steps to be followed for Parent Workflow configuration:

1. Use Return Values property to set the routing condition after completion of child workflow. Example: 'ChildWfCompleted'.

   This will dynamically create a link output named 'ChildWfCompleted' for the activity. Once the child workflow complete its execution, it returns back to the parent workflow with the return value specified in the child workflow (In this case, child workflow should return 'ChildWfCompleted' value) and then it will continue with further execution.
2. Go back to the parent and set routing condition based on the output 'ChildWfCompleted'.
3. Parent workflow goes to sleep mode after the child is executed.
4. Child completes execution with Update Variable as the last action.
5. Once the child completes execution, the value set in Update variable is given as output for the workflow action in parent.

Getting Execution ID of the Child Quickflow

You can get Execution ID of the Child Quickflow from the Parent Quickflow as follows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To Access Execution ID of the Child Quickflow

1. Create a variable in the parent Quickflow with a name such as **"ChildQFExecutionId" + <<Child Quickflow Action Name>>**.

   For example: **ChildQFExecutionIdQuickflow1**
2. Access the variable **<%#Variable.ChildQFExecutionIdQuickflow1%>**.

   The variable will return the value of the Child Quickflow Execution ID.

XML Navigator

Activity Description:

The XML Navigator activity is used to either locate or navigate to a specific record in a collection within an XMLIterator Variable. This activity enables identification of a specific record for execution of an activity or group of activities.

See also *Example of XML Navigator Activity*

Activity Properties:

You can configure the XML Navigator activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **XML Navigator** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

Name & Description

Specify the name and description for the activity in this group.

Name - Enter the name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - Enter a brief description for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Configuration

The Configuration group has properties for selection of the XMLIterator Variable to be looped through.

XMLIterator - Select the XML Iterator variable to be navigated. All XMLIterator variables as declared in the Start activity are available for selection.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Location - This field identifies the location to be navigated to. It defaults to First. The options are:

First: Navigates to the first record in the collection.

Last: Navigates to the last record in the collection.

Next: Navigates to the next record based on the current location.

Previous: Navigates to the previous record based on current location.

Index: Navigates to the record identified in the Index field.

Property Type: Mandatory (This property must be set if the action is to be executed.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Index - Enter the index number of the record to navigate to. This field is mandatory if the Location field is set to Index.

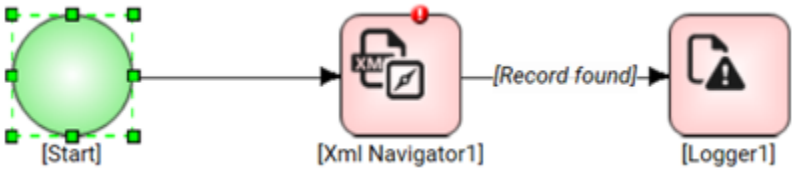
Property Type: Optional (This property needs to be set only if necessary.)

Example for XML Navigator

Consider already an XML file called Sample.XML is exists. The contents of it are as shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<root>
<Item>
<itemdID>I001</itemdID>
<desc>some stuff1</desc>
<qty>10.20</qty>
</Item>
<Item>
<itemdID>I002</itemdID>
<desc>some other stuff2</desc>
<qty>20.20</qty>
</Item>
<Item>
<itemdID>I003</itemdID>
<desc>some more stuff3</desc>
<qty>30.20</qty>
</Item>
</root>
```

Step 1: Design a Workflow as shown in the below figure.



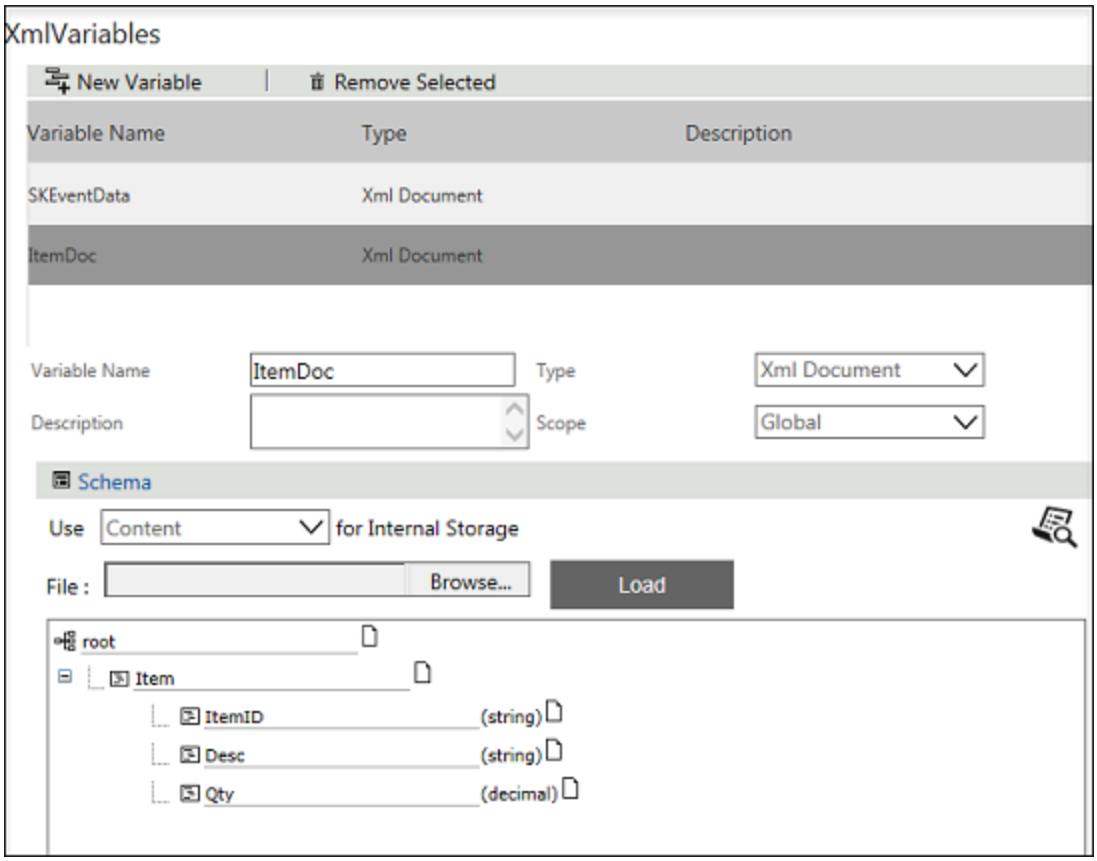
Step 2: In the Start Activity, declare two XML Variables.

1. Click the New Variable link. Under Schema enter the following details:

   Variable Name: ItemDoc, **Type**: XML Document, and **Scope**: Global

   Use **Content** for internal storage and browse the above created Sample.XML file and load it.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

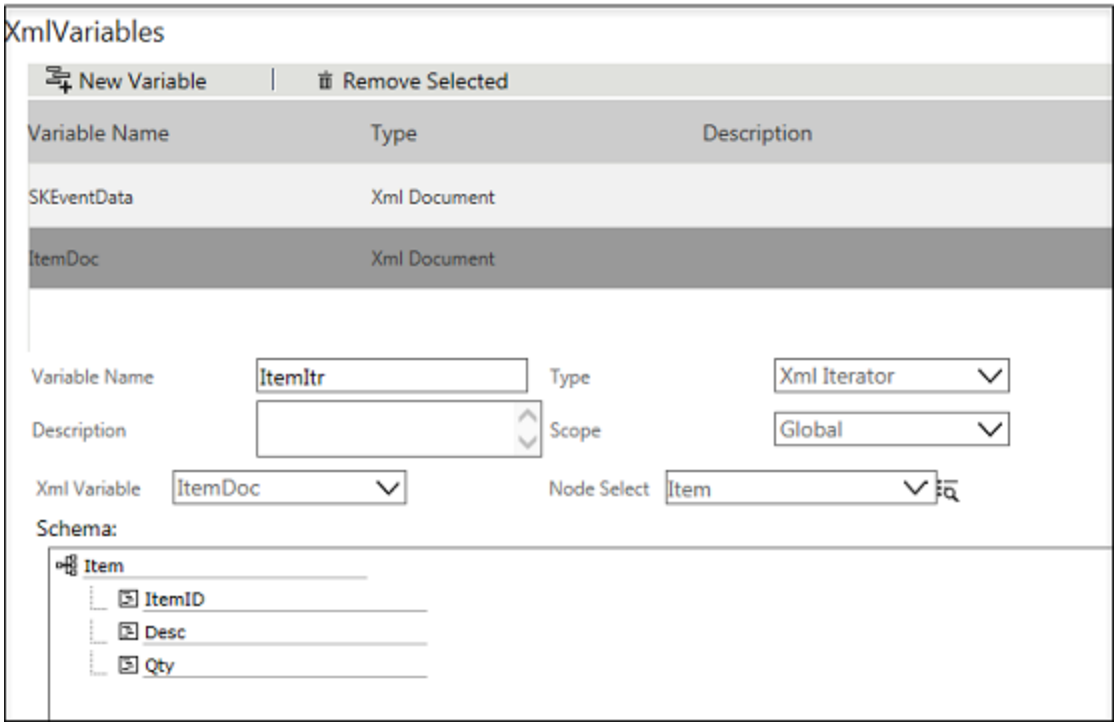


2. Click the **New Variable** link. Under Schema, enter the following details:

   Variable Name: ItemItr,
   Type: XML Iterator,
   Scope: Global,
   XML Variable: ItemDoc
   Node Select: Item

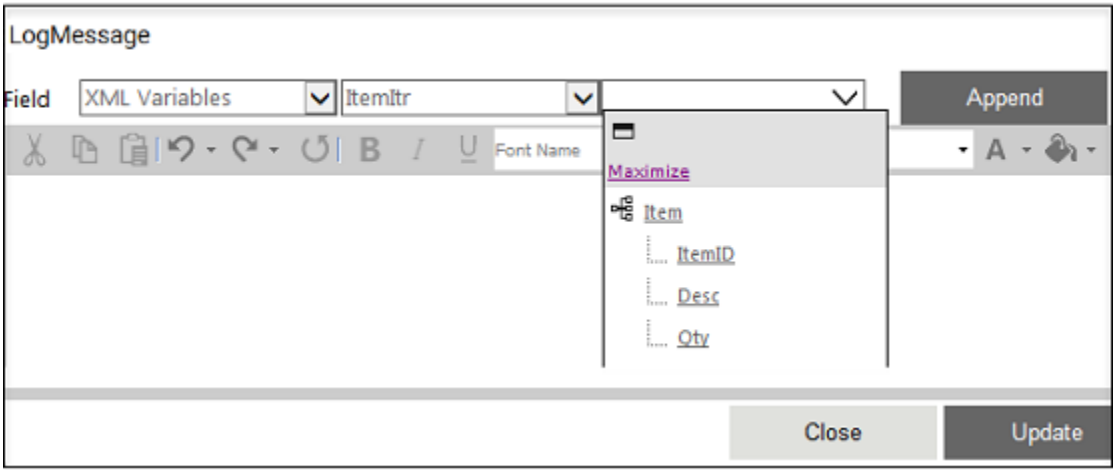
3. Click **Save Variable** and close the window

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Step 3: In the location property of XML Navigator Activity, select "Last". For the XML Iterator property, select the Iterator created that is, ItemItr. In the Logger Activity, the ItemItr is displayed in LogMessage column of the work item.

Step 4: Click the LogMessage property of Logger Activity, following window will open.



Select XML Variable form the first drop-down. In the second drop-down, select the Iterator created, that is ItemItr. In the third drop-down, select the Item ID from the tree displayed. Click **Append**, and then click **Update**.

Step 5: On the link connecting XML Navigator Activity and Logger Activity, select "Record Found".

Output:

The item ID will be displayed in the LogMessage Column of the Logger in work item.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Integration Activities

The following are the types of Integration activities:

- Invoke Web API Activity: The Invoke Web API activity is used to invoke a web API from the workflow.
- Script Activity: The Script Activity enables the execution of script written in languages, such as Jscript and VBScript, from the workflow.
- XML Action Activity: The XML Action Activity can be used to assign XML data for activity.

Invoke Web API Activity

Activity Description

You can use the Invoke Web API activity to invoke a web API from the workflow. Use the output of the activity to determine the path of workflow execution.

The output is triggered based on the http Status Code returned by the activity. Alternatively, the output Status Code can be saved in a AVEVA Work Tasks system variable.
The variable has to be prefixed with "^" followed by "Activity Name _StatusCode". For example, if the activity name is Invoke Web API1, then variable name should be "^Invoke Web API1_StatusCode" and the data type should be "string".

The Error Message can be saved in the AVEVA Work Tasks system variable.
The variable has to be prefixed with "^" followed by "Activity Name _WebAPIErrorMessage". For example, if the activity name is Invoke Web API1, then variable name should be "^Invoke Web API1_WebAPIErrorMessage" and the data type should be "string".

The additional error information (i.e. http custom content message) can be saved in the AVEVA Work Tasks system variable.
The variable has to be prefixed with "^" followed by "Activity Name _WebAPIErrorDetails". For example, if the activity name is Invoke Web API1, then variable name should be "^Invoke Web API1_WebAPIErrorDetails" and the data type should be "string".

You can use any HTTP client to invoke your web API. In fact, you can invoke it directly from a web browser. When the Web API framework receives a request, it routes the request to an action. To determine which action to invoke, the framework uses a routing table.

Activity Properties

You can configure the Invoke Web API activity by specifying appropriate values in the Activity Properties area. To access the Activity Properties, click the **Activity Properties** menu in the Tool Bar. Or, right-click the **Invoke Web API** activity and select the **Activity Properties** option. The Activity properties are organized under groups of related properties. The properties under each group are described in this topic.

Name & Description

Use the Name and Description properties to enter the the name and description for the activity. This property is optional, and can be set if required.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Name**: Specify a name for the activity.
- **Description**: Specifies a brief description to be displayed for the activity.

Configuration

Use the configuration properties to configure the functionality of the activity.

Configure Web API: Use this property to configure the Web API to be called. Web APIs are frameworks to build HTTP services for browsers and mobile devices.

For more information about this activity property, see Configure Web API.

Looping Behavior

Use this property to specify the looping behavior of the activity. This property is optional, and can be set if required.

Loop through this action if output is: Specify the condition for looping the activity. Select any one of the following activities from the list:

- No Loop
- BadGateway(502)
- BadRequest(400)
- Error Encountered
- Forbidden(403)
- GatewayTimeout(504)
- HttpVersionNotSupported(505)
- InternalServerError(500)
- MethodNotAllowed(405)
- NotAcceptable(406)
- NotFound(404)
- NotImplemented(501)
- ProxyAuthenticationRequired(407)
- RequestTimeout(408)
- ServiceUnavailable(503)
- Successful
- Unauthorized(401)
- Unsuccessful
- UnsupportedMediaType(415)

Note: For the Invoke Web API activity, the StatusCodes "OK(200)" and 'ConnectFailure(0)' is not saved in the system variable (i.e. "^Activity Name_StatusCode").

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Invoke Web API: Configure Web API

You can configure the Web APIs in the **Set Web API Configurations** window. This window lists all the Web APIs configured for the application or workflow.

Http Method

The **Invoke Web API** activity supports the following HTTP request methods:

- **GET**: The GET method retrieves information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.
- **POST**: A POST request sends data to the server, for example customer information, file upload, using HTML forms.
- **PUT**: A PUT request replaces all current representations of the target resource with the uploaded content.
- **DELETE**: A DELETE request removes all current representations of the target resource given by the URI.

Web API

Web APIs are frameworks to build HTTP services for browsers and mobile devices.

If the Web API is Windows authenticated, then the status of the Windows Authentication must be set to **Enabled** for the **EnterpriseConsole** site under **WonderwareSkeltaBPMSites** site in the **IIS Manager**.

If the Web API requires ArchestrA Security Token Service (ASTS) access token authentication, then the Workflow Engine Service must first be registered as a client in the ASTS server through **Central Configuration** site. Web API with the Access Token enabled Authentication gets listed in the Web API drop-down list, so that it can be used within the workflow.

Request URI

The Request URI is a Uniform Resource Identifier and identifies the resource upon which to apply the request.

Example:

- Request URI without parameters: **api/product/1** for **http://localhost/MySite/api/product/1**
- Request URI with parameters: **api/{product}/{ID}** for **http://localhost/MySite/api**

Use System Authentication

Select the **Use System Authentication** check box to use the access token of the Workflow Engine Service.

Content Type

You can configure the content type of the Post Data. The available options are **JSON**, **XML**, and **URL ENCODED**.

This field is displayed only when the value of **Http Method** is set to **POST** or **PUT**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

POST Data

Passing data to POST and PUT Methods

You can pass data to POST and PUT methods either through parameters or through the request body. If you pass data through request body, use the adjacent text box for the body content. If web API method uses [FromBody] attribute, select the **Yes** option for the **FromBody Attribute** property.

Output Settings

Save Output In

Use the options in this field to save the return value from the Web Method so that you can use it in subsequent activities. You can save the return value as either a **Variable** or as an **XML Variable**.

To save the output as a **Variable**, select the variable that you created in the **Start** activity. To save the output as an XML variable, select the **XML Variable** that you created in the **Start** activity. You can also create the XML variable using the **Create New** option.

Note: Create XML variable will work only if the web API returns valid data. Therefore, for Delete, Put, and Post, use an existing variable.

Invoke Web API: Configure OpenAPI

AVEVA Work Tasks allows you to configure OpenAPI, and it is compliant with Swagger 2.0 and Swagger 3.0.

To configure the OpenAPI

1. Right-click the **Web API Activity**, and then click **Activity Properties** option. In the Activity Properties window, click **Configure Web API**.

   The **Web API Activity Configuration** window opens.
2. In the **Web API** drop-down list, select the Web API list item, which is an OpenAPI.

   This loads the OpenAPI data, which gets stored when OpenAPI list item is created using OpenAPI.
3. In the **Operation Group** dropdown list, select the operation group provided in the JSON file of the selected OpenAPI. These groups are the tags or the logical groupings of the selected OpenAPI.

   Tags are used as Operation Group. Though Tag is an optional element in the OpenAPI Specification, if available, it will be considered as Operation Group item.

   If Tags is empty/not available, then it will not list out in Operation Group, but the operation will be directly listed under All (by default) Operation Group.
4. In the **Operation** drop-down list, select the operation ID of the OpenAPI defined for the selected Operation Group in the JSON file of the selected OpenAPI.

   The OperationId is considered as Operation and will be listed out in the drop-down.

   The OperationId is an optional, unique element as per the OpenAPI Specifications. But if any of the OperationId is duplicate, then the same will be logged in the Logger.

   If the OperationId is not present, then Path_HttpMethod will be displayed as Operation under the Operation dropdown.

   Based on the Operation Group selection, the Operation will be listed in the Operation dropdown.

   If there are no Tags available for some operations, then they will appear in All Operation Group along with all

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

other operations.

When the Operation is selected, the HTTP methods (GET, PUT, POST, DELETE, or PATCH) and the associated parameters are retrieved from the JSON file of the selected OpenAPI.

Note: If Operation ID is not available, the Operations Group tag is considered.

5. In the **HTTP Method**, the HTTP method is displayed when you select Operation, and this is a disabled field.
6. The **Headers** tab is displayed only if the Header Keys and Values are configured in the OpenAPI resource file. You cannot add or delete the Header Keys and Values for the OpenAPI.

   Note:
   - The **Show Header** option is available to hide/unhide the **Headers** tab in the **Web API Activity Configuration** page. Select **Yes** if you want to view, add, and delete the Header Keys and the Values, else select **No**. You can add the Value using Expression Editor.
   - You cannot set the **Optional** property of the **Header**. You cannot provide the Key, but you need to provide the Value. If the **Optional** property is **YES** and the Value is empty, the value will not be sent to the Web API and will be considered null. If the **Optional** property is **NO**, you need to provide the Value.
7. In the **Parameters** and **Body** tabs, enter the details given in the respective tables in this topic.
8. In the **Output settings**, configure the **Save output type** and **Save output in**.

   To save the output in a **Variable**, select the variable that you created in the **Start** activity. To save the output in an **XML Variable**, select the XML variable that you created in the **Start** activity. You can also create the XML variable using the **Create New** option from the **Save output in** dropdown.

   You can overwrite the XML Variable using the **Update XML Variable** option after modifying the OpenAPI configuration.

   Note: If OpenAPI specification JSON file does not have a response for the selected Operation, then it will prompt the user as "The selected operation does not return any value."
9. Click **Save**.

   The OpenAPI gets configured.

Parameters Tab

This tab is displayed if the selected Operation has parameters.

In the Parameter's grid, the following parameters are listed:

- Path parameters: the parameters' value is part of the operation’s URL
- Query parameters: the parameters that are appended to the URL

Note: Header and Cookie parameters are not supported.

The table describes the fields available under **Parameters** tab.

Fields	Description
No	It displays the serial number of the parameter and represents its sequence.
Optional	It displays **YES** or **NO** values. **YES**, denotes that you can either enter the value for

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Fields	Description
	the parameter in the **Value** field or leave it blank. **NO**, denotes that you must enter the value for the parameter in the **Value** field. **Note**: For the path parameter, the Optional field is always NO, but for query parameter it can be either YES or NO.
Name	It displays the name of the parameter.
Data type	It displays the data type of the parameter. The supported data types are boolean, decimal, numeric, and string. **Note**: The OpenAPI-specific datatype formats are not validated.
Value	Enter the value to be passed for the parameter. The value is validated against the data type. The value can be hard-coded or can be passed through the Expression Editor. **Note**: The value passed using Expression Editor is not validated for the data type.

Body Tab

This tab is displayed if the selected Operation is POST, PUT, or PATCH.

It is used to pass PostData to the OpenAPI, and the PostData can be passed as raw format or in the form of parameters, hence two options - **Parameters** and **Raw** are available with the Radio buttons.

Both Parameters and Raw are validated based on the selection of Content Type.

Note: In the **Body** tab, it is mandatory to provide data either in the **Parameters** or the **Raw** option.

Parameters:

The request body is populated in the form of the name, data type, and the value to map individual parameters with a hard-coded value or to a dynamic value that is build-up through the Expression Editor.

The table describes the fields available under **Body** tab for **Parameters**:

Fields	Description
No	It displays the serial number of the parameter and represents its sequence.
Optional	It displays **YES** or **NO** values. **YES**, denotes that you can either enter the value for the parameter in the **Value** field or leave it blank. **NO**, denotes that you must enter the value for the parameter in the **Value** field.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Fields	Description
	Note: For the path parameter, the Optional field is always NO, but for query parameter it can be either YES or NO.
Name	It displays the name of the parameter.
Data type	It displays the data type of the parameter. The supported data types are boolean, decimal, numeric, and string. **Note**: The OpenAPI-specific datatype formats are not validated.
Value	Enter the value to be passed for the parameter. The value is validated against the data type. The value can be hard-coded or can be passed through the Expression Editor. **Note:** The value passed using Expression Editor is not validated for the data type.

On the Body parameters, constitute the PostData, and pass it to the OpenAPI during execution.

Raw:

In this tab, you can provide raw data as PostData, either a hard-coded value or dynamic value that is built up through the Expression Editor.

The table describes the fields available under **Body** tab for **Raw** data:

Fields	Description
Post Data	Enter the value you want to pass to POST, PUT, and PATCH methods through the body. The value should be in the format based on the content type selected in the Content Type dropdown.

Note:
- For more information on PATCH method, see PATCH Method.
- Passing empty/null parameter in RAW JSON/XML format as PostData is not supported. The user needs to provide correct JSON/XML data in RAW format without null/empty values.
- During design, the value provided as raw data to pass to the POST or PUT method through the body is not retained when you change the operation (PUT or POST).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Script Activity

Activity Description

The Script Activity enables the execution of script written in languages, such as Jscript and VBScript, from the workflow. When the Script activity is included in the workflow, the activity calls the Run method in the script. The references to the assemblies in the script are taken from the activity properties. The string value returned from the Run method of the script is assigned to the Output parameter. The Script activity executes the script and gives the return values specified in the properties. The link matching this value is picked up to continue workflow execution.

Features of Script Activity

- Compilation of the script code is made at design time to enable discovery of syntax or other errors during design.
- Loading all scripts defined through script activities as a single assembly. All script activities are compiled as a single assembly and only one assembly is loaded during workflow execution. This significantly improves performance.
- When a workflow is published, the script details are stored in the SWScriptCode table to enable identification of source code and assembly references.

Note: Framework 4.0 has a library named System.Action and results in an ambiguity between the System.Action and Workflow.NET.Action. To avoid this ambiguity, it is recommended to name the Action class with complete namespace as 'Workflow.NET.Action' if it is used in the scripts inside script activity, while running BPM.NET 2011 SP2 in Framework 4.0.

Activity Properties

The Script activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the Tool Bar or by selecting the appropriate option from the right click menu for the Script activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

The following are the Script Activity Properties:

Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

The following image shows the Name & Description properties of Script Activity:

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Programming

The properties in the Programming group have been described below. You can use these properties to program the script activity.

The following image shows the Programming Properties of the Script Activity:

Language - This property specifies the language of the script to be run. This property can be set to 'C#.NET' or 'VBScript' depending on the scripting language required.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Code - This property specifies the code of the script to be run.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the Code property:

1. To enter the code, click on the button for the 'Code' property to open the property window.
2. In this window enter the code directly in the text editor.
3. If you want to assign the code dynamically during run-time, you can do so using the **Field** controls. For example, you can add a variable or content pointing to the code from the two drop downs for **Field** and click on the **Append** button. This will add the placeholder for the variable or content in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the code.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the code, click the **Update** button to save it
7. Click the **Close** button to close the window without saving the code.

See Script - Code for a detailed description of the Code property window.

Reference - This property specifies the global objects that can be referenced in the script.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the References property:

1. To specify the References, click on the button for the 'References' property to open the property window
2. In this window, enter the reference object in the **References** field
3. Next, enter the corresponding assembly path in the **Assembly Path** field
4. After entering the return value details, click the **Add Row** button to add this reference and display a new row for entering another reference
5. Remove rows if necessary by selecting them and clicking the **Remove Selected** button
6. After specifying the references, click the **Update** button.

See Script - References for a detailed description of the References property window.

Output Values

The properties in the Output Values group have been described below. You can use these properties to specify the output values for the activity.

The following image shows the Output Values Properties of the Script Activity:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Return Values - This property allows the user to specify the return values for this action. These values become the conditions for the workflow engine to pick up the next action to execute.

Property Type: Mandatory (This property must be set if the action is to be executed.)

Steps to set the Return Values property

1. To specify the Return Values, click on the button for the 'Return Values' property to open the property window
2. In this window, enter the return value in the **Return Values** field
3. After entering the return value details, click the **Add Row** button to add this return value and display a new row for entering another return value
4. Remove rows if necessary by selecting them and clicking the **Remove Selected** button
5. After specifying all the return values, click the **Update** button.

See Script: Return Values for a detailed description of the Return Values property window.

Action outputs:

The Script Action does not have any standard return value. It will output the value specified by the user in the Return Values property.

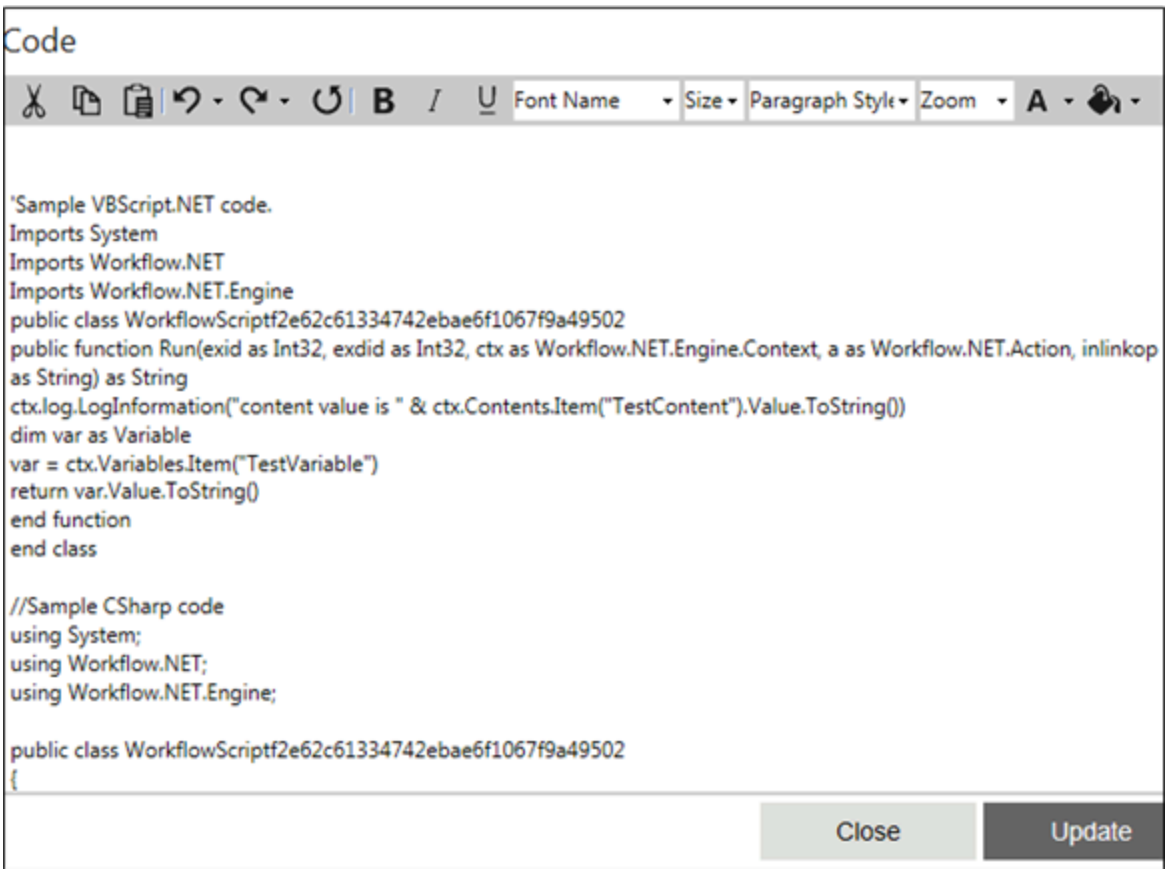
Script: Code

Code Property Window

You can set the code of the script in the **Code** property window. To open this window, click the button for the "Code" property.

The following image shows the script **Code** property window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The **Code** property window contains a text editor to enter the script code. You can do this either by entering a discrete value directly in the text editor. You can use the text tool bar in the text editor to manage and format the code.

The text tool bar contains the following options:

Options	**Shortcut Key**	**Description**
Cut	Ctrl+X	Cut text
Copy	Ctrl+C	Copy text
Paste	Ctrl+V	Paste copied text
Undo	Ctrl+Z	Undo the last action
Redo	Ctrl+Y	Redo an action that you undid
Repeat Last Command	F4	Repeat the last command that was executed
Bold	Ctrl+B	Make text bold
Italic	Ctrl+I	Italicize text
Underline	Ctrl+U	Underline text

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Options	Shortcut Key	Description
Font Name	-	Select the font for the parameter
Size	-	Enter the font size
Paragraph Style	-	Select a paragraph style
Zoom	-	Zoom in and zoom out the view
Foreground Color	-	Set text color
Background Color	-	Set the background color
Print	-	Print the text

The **Code** property window also contains the following tabs:

- **Update**: Use this tab to save the parameter.
- **Close**: Use this tab to close this window.

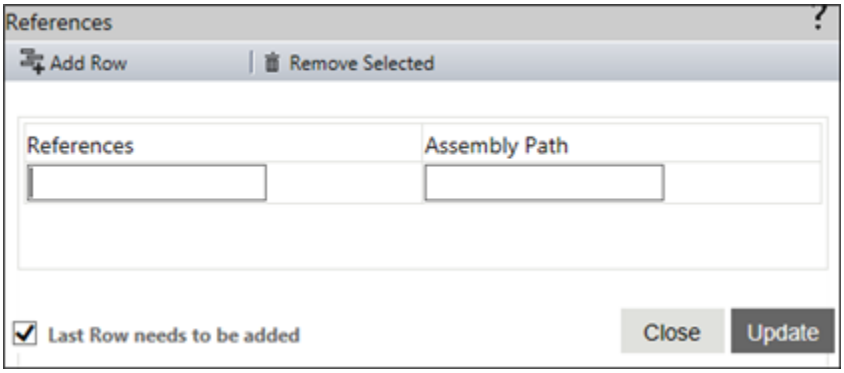
Script: References

References Property Window

You can set the references for the script in the **References** property window.

To open this window, click the button for the "References" property. It displays the references in a grid. The grid contains a blank row where you can enter the parameters for new references.

The following image shows the **References** property window:



The **References** property window contains the following fields and tabs:

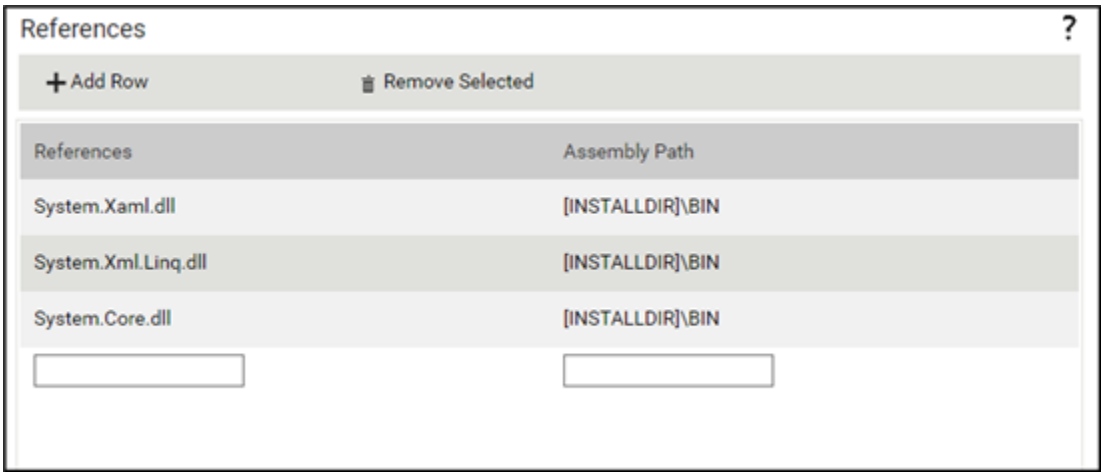
- **Add Row**: Use this tab to add the newly entered reference, and display a new row to the grid where you can add another reference.
- **Remove Selected**: Use this tab to remove the reference selected in the grid.
- **References**: Use this field to enter a new reference object.
- **Assembly Path**: Use this field to enter the assembly path for the new reference.
- **Update**: Use this tab to save the newly added reference(s).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Close**: Use this tab to close the property window without saving the newly added reference(s).
- **Last Row needs to be added**: Select this check box to save the last row added.

Enter the relative path, **[INSTALLDIR]\Bin** as the assembly path to refer to the external assemblies that are referenced in script action and are available in the AVEVA Work Tasks installed Bin folder.

The following image shows the external assembly relative path:



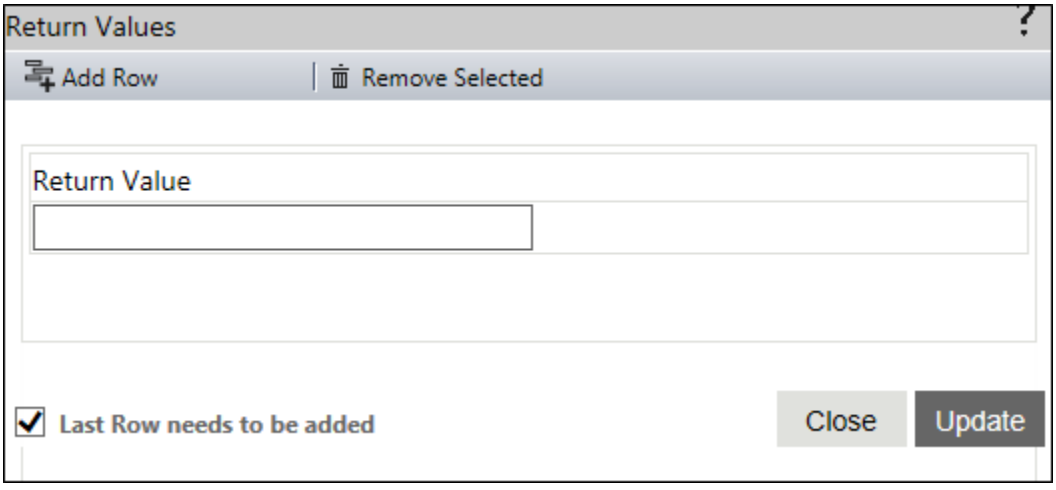
Script: Return Values

Return Values Property Window

You can set the return values for the Script in the **Return Values** property window.

To open this window, click the button for the "Return Values" property. It displays the values in a grid. The grid contains a blank row where you can enter a new return value.

The following image shows the **Return Values** property window:



The **Return Values** property window contains the following fields and tabs:

- **Add Row**: Use this tab to add the newly entered return value, and display a new row to the grid where you can add another value.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Remove Selected**: Use this tab to remove the return value selected in the grid.
- **Return Values**: Use this field to enter the return value.
- **Update**: Use this tab to save the newly added return value(s).
- **Close**: Use this tab to close the property window without saving the newly added return value(s).
- **Last Row needs to be added**: Select this check box to save the last row added.

XML Action Activity

Activity Description

The XML Action Activity can be used to assign XML data for activity. This was possible earlier only by using Contents and Variables defined in the Start activity. The activity has been enhanced to accept XML data containing namespaces. The namespace manager that has been added to the activity handles namespaces within the XML data internally.

View an example of assigning XML data with namespaces to a variable

The namespaces in the following example used to illustrate the extraction of a particular data are highlighted in red.

```
<tr:travelRequest xmlns:xhtml="http://www.w3.org/1999/xhtml" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance"
<tr:travelRequest xmlns:xhtml="http://www.w3.org/1999/xhtml" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance">
<tr:description></tr:description>
<tr:date xsi:nil="true"></tr:date>
<tr:traveler>
<tr:name>
<tr:prefix></tr:prefix>
<tr:givenName>guru</tr:givenName>
<tr:middleName>nayak</tr:middleName>
<tr:surname></tr:surname>
<tr:suffix></tr:suffix>
<tr:singleName></tr:singleName>
</tr:name>
</tr:traveler>
</tr:travelRequest>
```

The XPath expression to retrieve the middle name and assign it to a workflow variable 'aa' is given below. The image shown below illustrates the actual setting of the variable with this XML data.

// tr:travelRequest/ tr:traveler/ tr:name/ tr:middleName

Application Scenarios

The XML Action can be used with other Activities to retrieve required XML data by defining XPath expressions. For example, it can be used with the Checklist activity to retrieve the Answers to Checklist Questions.

The XPath expressions for retrieving answers to Checklist Questions

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The XML Activity can be used with the Checklist Activity to retrieve the Answers to the Checklist Questions. The XML data format for the Checklist Activity to be declared in the Start activity is '^'+ activity name + 'AnswersXMLCheckList' , e.g. '^Checklist1AnswersXMLCheckList'.

The XPath expressions to retrieve the Answers from the Checklist XML string are given below:

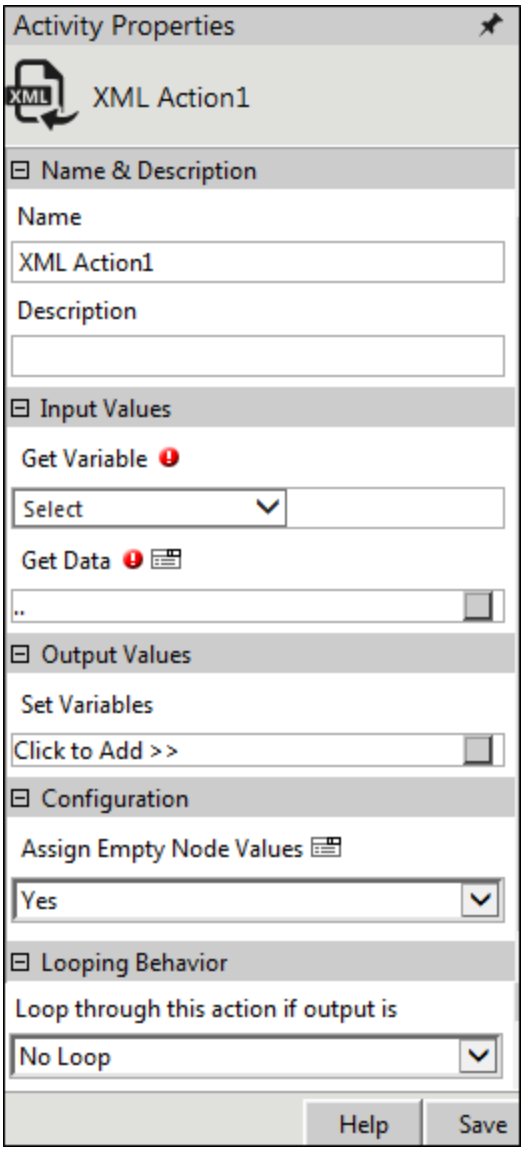
XPath with attribute:

```
//checklist/question[@name="RADIO"]/answer
//checklist/question[@name="DROPDOWN"]/answer
//checklist/question[@name="TEXTBOX"]/answer
//checklist/question[@name="MEMO"]/answer
//checklist/question[@name="CHECKBOX"]/answer
```

Activity Properties

The XML Activity has to be configured by specifying appropriate values for the different properties in the Activity Properties area. The Activity Properties can be accessed by clicking on the Activity Properties tool in the tool bar or by selecting the appropriate option from the right click menu for the XML Activity. The Activity Properties are organized under the following groups of related properties. The properties under each group are described in this topic.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Name & Description

The properties in the Name & Description group have been described below. You can use these properties to specify the name and description for the activity.

Name - This property can be used to specify a name for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Description - This property specifies a brief description to be displayed for the activity.

Property Type: Optional (This property needs to be set only if necessary.)

Input Values

The properties in the Input Values group have been described below. You can use these properties to specify the input values for the activity.

Get Variable - This property allows the user to specify the variable that holds the XML string. This variable is defined in the Start activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Optional (This property supports backward compatibility.)

Get Data - This property allows the user to pass the XML data. This data can be passed in one of two forms: (1) XML String, or (2) XML File (where the complete path of the XML file is specified).

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Get Data property

1. To specify the XML data, click the button for the 'Get Data' property to open the property window.
2. In this window enter the XML data directly in the text editor. You can either specify the file path to an XML file here or enter the XML data directly.
3. If you want to assign the parameter dynamically during run-time, you can do so using the **Field** controls. For example, you can add a variable or content pointing to the XML data from the two drop downs for **Field** and click on the **Append** button. This will add the placeholder for the variable or content in the text editor. At run-time this placeholder will be replaced by the corresponding value, i.e. the XML data.
4. You can manipulate the text using the appropriate tools from the tool bar. You can cut, copy and paste text, undo and redo changes, zoom in and zoom out the view as well as print the text.
5. You can format the text, if necessary, using the formatting tools. You can make text bold or italic or change the font or size, set text color or background color.
6. After adding the XML data, click **Update** to save it.
7. Click to close the window without saving the XML data.

See XML Action: Get Data for a detailed description of the Get Data property window.

Output Values

The properties in the Output Values group have been described below. You can use these properties to specify output values for the activity.

Set Variables - This property allows the user to set the XPath to retrieve the required XML attribute from the XML string that is set in the Get Variable property.

Property Type: Mandatory (This property must be set if the activity is to be executed.)

Steps to set the Set Variable property

1. To set the XML data to selected variables, click the button for the 'Set Variables' property to open the property window.
2. In this window, select the variable from the drop down for the **Variable Description** field.
3. Next select the variable type from the **Type** drop down.
4. Specify the XPath Expression for the XML data to be assigned to the selected variable in the **XPath Expression** field.
5. After entering the details, click **Add Row** to add this variable setting and display a new row for assigning XML data to another variable.
6. Remove rows if necessary by selecting them and clicking **Remove Selected**.
7. After setting the XML data to the variables, click **Update**.

See XML Action: Set Variables for a detailed description of the Set Variables property window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration

The property Assign Empty Node Values in the Configuration group has been described below. You can use this property to configure the functionality of the activity.

Assign Empty Node Values

This property allows the user to choose whether the XML node value can be updated with an empty value or not.

If this property is set as "Yes", then the variables can be updated with the "Empty" node values and the activity will complete with a "Successful" output.

If this property is set as "No", then the variables will not be updated with the "Empty" node values and the activity will complete with an "UnSuccessful" output.

Property Type: Optional (This property needs to be set only if necessary.)

Message Queue Activities

This section provides information on the SendMQData activity. This activity is used to publish messages to Message Queue.

SendMQData

Activity Description

Use SendMQData activity to publish message to Message Queue.

Activity Properties

To use SendMQData activity, follow the given steps:

1. In the **Process Designer** window, select **Message Queue** from the **Activities** drop-down list.
2. Drag and drop the **SendMQData** activity on the **Process Designer** window.
3. Right-click the **SendMQData** activity and then select **Activity Properties**.

Name & Description

Use the name and description properties to specify the name and description for the activity.

- **Name**: Use this property to enter a name for the activity.

  Property Type: Mandatory.
- **Description**: Use this property to enter a brief description to be displayed for the activity.

  Property Type: Optional. Set this property only if required.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configuration

Use these properties to configure the functionality of the activity.

- **Message Queue Connection**: Use this property to configure Message Queue connection. For more information, see Configuring Message Queue Connection.

  Property Type: Mandatory

Activity outputs

The SendMQData activity returns the following values:

Successful: Displayed when the activity publishes the message to Message Queue.

Unsuccessful: Displayed when the activity fails to publish the message to Message Queue.

Configure Message Queue Connection

Use Message Queue Connection Settings property to configure connections to Message Queue.

Based on the **Connection** type, **Message Name** and **Message Type** input fields are visible/invisible.

In case of **WM Object** connection type, these input fields are visible to the user, so that the user can pass this information. In case of other connection types, these fields are not needed, hence they are invisible.

1. Select the name of the Message Queue connection from the Connection list. Connections with Connection Type selected as Send would be listed in the list.
2. Select the Binding type for the Connection.

   The following fields appear based on the Binding Type selected for the Connection.

Binding Type for Connection	Field Name	Description
WM Object	Message Name	Enter the Message name or click the Expression Editor Icon to define an expression in the Expression Editor.
	Message Type	Enter the Message type or click the Expression Editor Icon to define an expression in the Expression Editor.
	Message Details	Enter the message in string format.
XML	Message Details	Enter the message in XML format.
JSON	Message Details	Enter the message in JSON format.

For more information about Expression Editor, see *User Guide > Process Designer > Setting Activity and Link*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Properties > Using the Expression Editor.

Note:
- It is recommended to use Expression Editor to enter or copy multi-line text in 'Message Details' of SendMQData Activity rather than directly copying into the input field.

- In Expression Editor the text has to be enclosed within double quotes " ".

Logic App Activities

This section provides information on the Call Logic App activity. This activity is used to call Logic App Workflows.

Call Logic App Activity

Activity Description

Use Call Logic App activity to call Logic App Workflows.

Activity Properties

To use Call Logic App activity, follow the given steps:

1. In the **Process Designer** window, select **Call Logic App** activity.
2. Drag and drop the **Call Logic App** activity on the **Process Designer** window.
3. Right-click the **Call Logic App** activity and then select **Activity Properties**.

Name & Description

Use the name and description properties to specify the name and description for the activity.

- **Name**: Use this property to enter a name for the activity.
  Property Type: Mandatory.
- **Description**: Use this property to enter a brief description to be displayed for the activity.
  Property Type: Optional. Set this property only if required.

Configuration

Use these properties to configure the functionality of the activity.

- **Logic App Connection and Data**: Use this property to get connections to Logic Apps and pass data, if required.
  Property Type: Mandatory

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity outputs

The Call Logic App activity returns the following values:

Successful: Displayed when the connection execution is successful.

Unsuccessful: Displayed when the connection execution is unsuccessful.

Configure Logic App Connection and Data

Use Logic App Connection and Data property to configure connections to Logic Apps and pass data.

To configure Logic App connection and pass data

1. Create a Logic App Connection. For more information, see Logic App Connections.
2. In the **Logic App Connection and Data** window, select a **Logic App Connection** created earlier.
3. Enter a variable for **Variable Name**.

   This variable stores the Logic App Alert Key.
4. Get the custom JSON schema from the Logic App to pass relevant data.

   For example:

```
{
"type": "object",
"properties": {
"id": {
"type": "number"
},
"machine": {
"type": "string"
},
"status": {
"type": "boolean"
}
}
}
```

5. Click **Add Logic App Data** to add a row of data.
6. Enter **Name** and select **Data Type** obtained from the custom JSON schema, and then enter a **Value**.

   For example:

   Name: id

   Data Type: Number

   Value: 9876

   Note:
   - The JSON keys are case-sensitive.
   - The supported data types are Boolean, Number, and String.
   - Click **Expression Builder** to generate value from an expression, if required.
   - Single level JSON structure is supported.
7. Repeat **Steps 4 to 5** to enter the required data.
8. Click **Save**, and then click **Close**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note:
- The data is not validated. Ensure name and data type are as specified in the custom JSON schema, and the value matches the data type.
-- In case of duplicate names, value passed is only for the first match.
-- In case of names from the standard JSON schema (e.g., ApplicationName), value passed will overwrite the default data.

Standard JSON Schema

Logic Apps must include the following standard JSON schema to interact with AVEVA Work Tasks.

```
{
"properties": {
"ApplicationName": {
"type": "string"
},
"ExecutionDetailsId": {
"type": "integer"
},
"ExecutionId": {
"type": "integer"
},
"LogicAppAlertKey": {
"type": "string"
},
"WorkflowName": {
"type": "string"
},
"UserName": {
"type": "string"
}
},
"type": "object"
}
```

AttributeReadWrite Activity

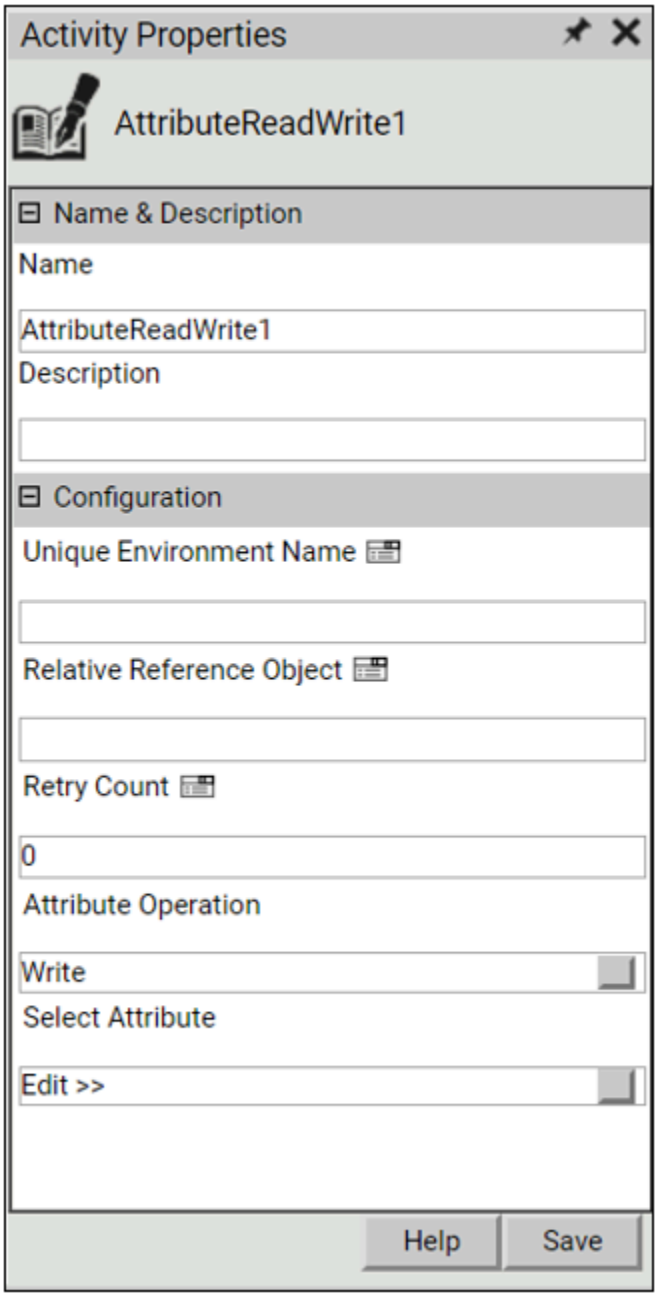
Use the **AttributeReadWrite** activity to perform the read and write operation between AVEVA Work Tasks and ArchestrA IDE.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Activity Properties

You can configure the **AttributeReadWrite** activity by specifying appropriate values for the properties in the **Activity Properties** area. Click **Activity Properties** in the toolbar or select the **Properties** option from the short-cut menu of the **AttributeReadWrite** to access the Activity Properties.

The following image shows the Activity Properties dialog box:



Name & Description

You can use these properties to specify the name and description for the activity.

- **Name**: Use this property to specify a name for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property Type: Mandatory

- **Description**: Use this property to add a brief description about the activity.

  Property Type: Optional. Set this property only if required.

Configuration

You can use these properties to configure the functionality of this activity.

- **Unique Environment Name**: Identifies the Galaxy and Runtime node on which the galaxy is deployed to perform the read or write operation on the System Platform attribute.
  - You can manually enter the value for this property. This value will be considered for further execution of the current activity.

    Or
  - If the property does not have a value, then the value defined for this property in the "ArchestrAXMLData" XML variable is used during the execution of the **AttributeRead/Write** activity. The "ArchestrAXMLData" XML variable is automatically created when you associate a workflow with an ArchestrA event.

    See also: Configure AttributeReadWrite Activity to Perform Read Operation and Configure AttributeReadWrite Activity to Perform Write Operation.
- **Relative Reference Object**: Relative references are especially useful in templates while passing context parameters to workflows.
  - You can manually enter the value for this property. This value will be considered for further execution of the current activity.

    Or
  - If the property does not have a value, then the value defined for this property in the "ArchestrAXMLData" XML variable is used during the execution of the **AttributeRead/Write** activity. The "ArchestrAXMLData" XML variable is automatically created when you associate a workflow with an ArchestrA event.

    For more information, see Relative Referencing.
- **Retry Count**: Use this property to enter the number of retries, in case of a communication error. If the error is due to invalid data, it will not be retried.

  Property Type: Optional. Set this property only if required.
- **Attribute Operation**: Define the attribute operation type. The attribute operation type defines whether the action should be read or write.
- **Select Attribute**: Select the attribute type for the Attribution Operation.

Configure AttributeReadWrite Activity to Perform Read Operation

You can read entire UDA of array type from System Platform and assign the same to variables of type array within a workflow. You can also read specific index of UDA of type array from System Platform and assign the same to variables within a workflow.

To read a specific index of UDA of type array from System Platform use Expression Builder.

Example: To read the 5th Element of the array , enter `return "UserDefined_001.Array1[4]";` in the

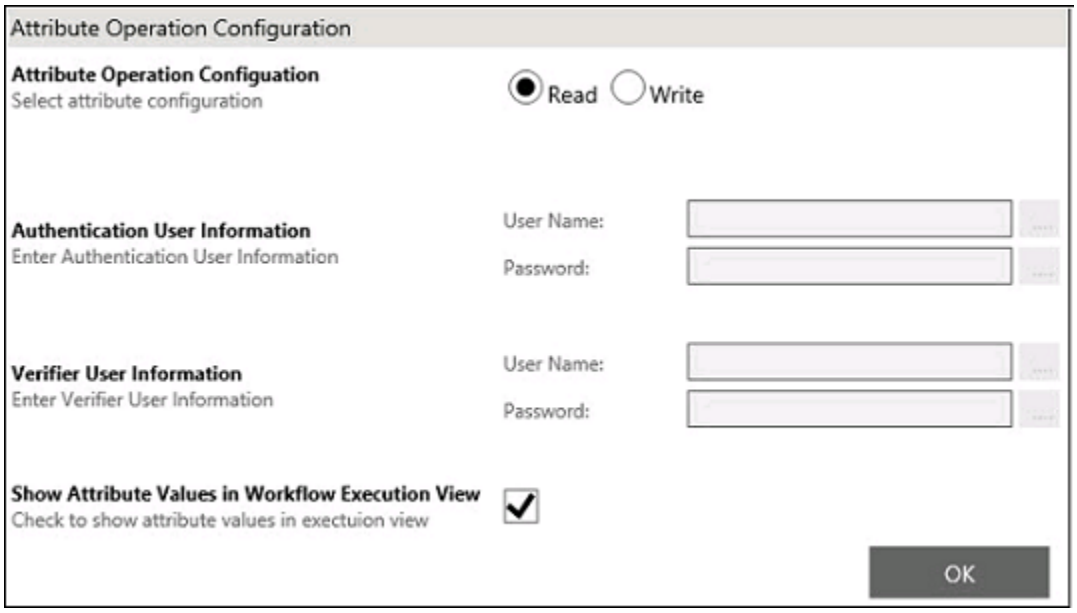
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

expression builder.

To configure AttributeReadWrite activity to perform a read operation

1. Drag-and-drop the **AttributeReadWrite** activity to the Process Designer window.
2. Set the unique environment name from Expression Editor.
3. Click the edit icon in the **Attribute Operation** property.

   The Attribute Operation Configuration dialog box appears.



Note: The **Show Attribute Values in Workflow Execution View** feature is not supported in Quickflow.

4. Select **Read**, and then click **OK**.
5. In the **Condition** drop-down menu, select a variable to store the attribute value.

   You can add the variable and type from **Variables** property in the **Start** activity.

   In a Remote Galaxy, the value of update variable for the **Select Attribute** is in the following format:

   Remote or Paired Galaxy_Name:ObjectName.UDA name

   In a local Galaxy, the value of update variable for the **Select Attribute** is in the following format:

   Me.UDA name
6. In the Actions tree menu, select the **Value** drop-down menu, and then select the Galaxy Browser attributes from which you want the read action to be performed.

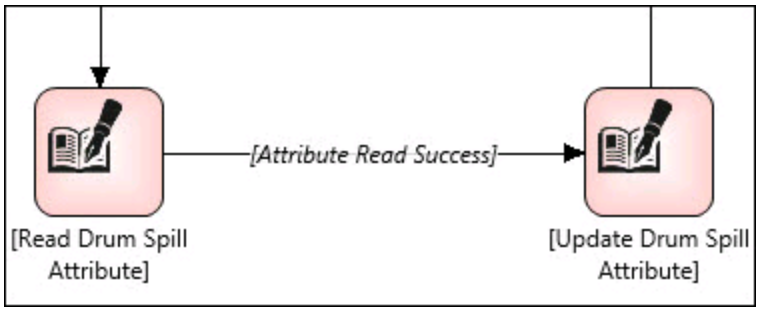
   The Galaxy Browser option appears only if you design the workflow in the IDE context.

   The Relative Reference dialog box appears.

   This dialog box displays support for relative referencing. Except for Me, you must manually enter all other relative references. For more details about relative referencing, see Relative Referencing.

   Create a link between the next activity based on the output options **Attribute Read Success** *or* **Attribute Read Failure**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



7. Publish the workflow.

   You can view the workflow in the Process Execution View dialog box.

   In the **Execution View**, the details of the Read/Write activity executed in a workflow appears in the **Step Details** tab. The **Execution View** page can be opened by double-clicking a top level row in any of the reports. The **View** represents the workflow up to the activity or execution status where the view is opened.

If any error occurs during Read operation, the error message will be logged with an appropriate error code generated from System Platform in the ArchestrA Logger.

For more information about the error logs, see *MxStatusDetail ValuesSystem* under the section *Status and Error Codes* of the *MXAccess Toolkit User's Guide* of *System Platform* documentation.

Configure AttributeReadWrite Activity to Perform Write Operation

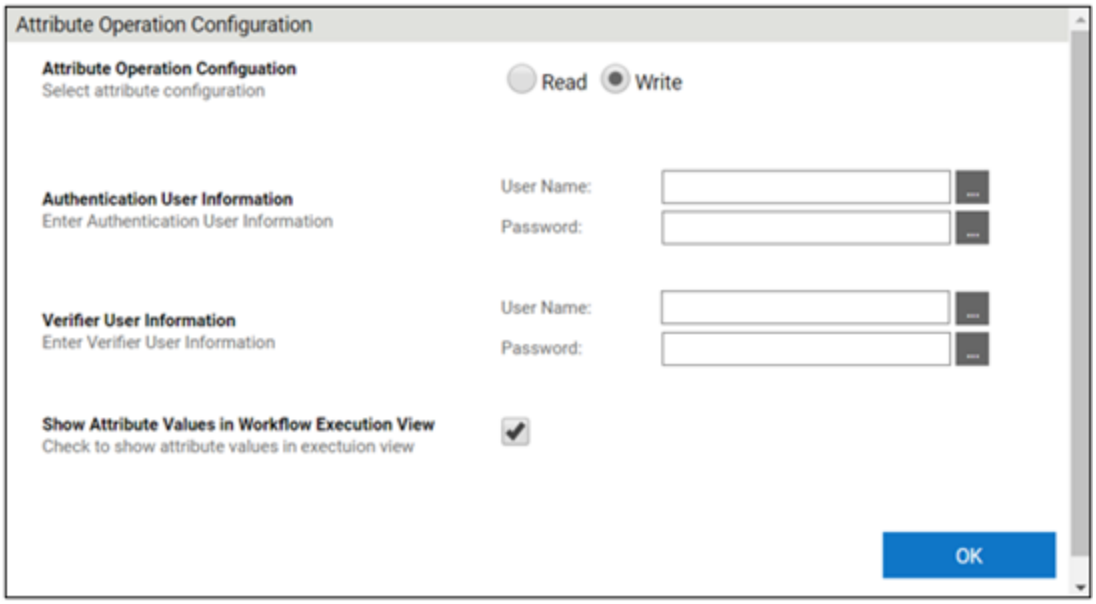
You can write entire UDA of array type to System Platform using a variable of array type within a workflow. However, unlike Read operation, you cannot write specific index of UDA of array type.

To configure AttributeReadWrite activity to perform a write operation

1. Drag-and-drop the AttributeReadWrite activity to the **Process Designer** page.
2. Set the unique environment name from Expression Editor.
3. Click the edit icon in the **Attribute Operation** property.

   The Attribute Operation Configuration window appears.
4. Select **Write** option in the Attribute Operation Configuration window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



5. Enter the **User Name** and **Password** for **Authentication User Information** to authenticate user information. This credential information is needed for **Operate**, **SecuredWrite**, and **VerifiedWrite** user accounts.
6. Enter the **User Name** and **Password** for **Verifier User Information** to verify user information. This credential information is needed for **VerifiedWrite** user account.
7. Click **OK**.

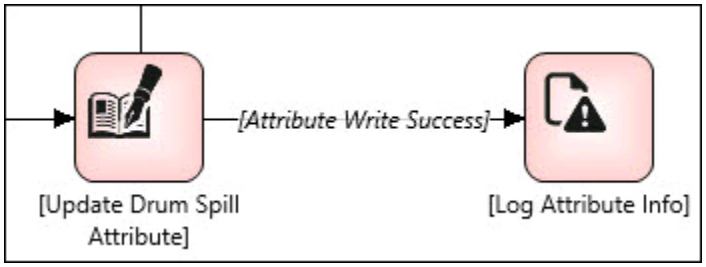
   Note: The **Show Attribute Values in Workflow Execution View** feature is not supported in Quickflow.
8. Set the authentication information from **Variable**, **Content**,or **XML variable**.

   Note: The **Content** option to set the authentication information is available, but not supported in Quickflow.
9. Select the desired **Type** from the drop-down list, and click **Update**.
10. Select the attribute to which the write action needs to be performed in the **Condition** drop-down list. Using expression builder or a variable get the value.

    The variable and type can be added from **Variables** property in **Start** activity.

    For write operation, the Galaxy Browser appears in the Condition drop-down list. The browser appears only if the workflow is designed in the IDE context.

    The Relative Reference dialog box appears. This dialog box displays support for relative referencing. Except for Me, you must manually enter all other relative references. For more details about relative referencing, see Relative Referencing.
11. Create a link between the next activity based on the output options **Attribute Write Success** or **Attribute Write Failure**.



12. Publish the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The workflow can be viewed in the Process Execution View page. In the **Execution View**, the details of the Read/Write activity executed in a workflow appears in the **Step Details** tab. The **Execution View** page can be opened by double-clicking a top level row in any of the reports. The **View** represents the workflow up to the activity or execution status where the view is opened.

If any error occurs during Write operation, the error message will be logged with an appropriate error code generated from System Platform in the ArchestrA Logger.

For more information about the error logs, see *MxStatusDetail ValuesSystem* under the section *Status and Error Codes* of the *MXAccess Toolkit User's Guide* of *System Platform* documentation.

Relative Referencing

Relative references are especially useful in templates while passing context parameters to workflows. You can use relative references like *MyContainer, MyPlatform, MyEngine, MyArea, and Me* to refer to contained objects within that container.

Relative referencing supports event context at the parameter level and the attribute read or write at the **AttributeReadWrite** activity level.

The following valid relative references refer to the current object:

- Me
- MyContainer
- MyArea
- MyPlatform
- MyEngine

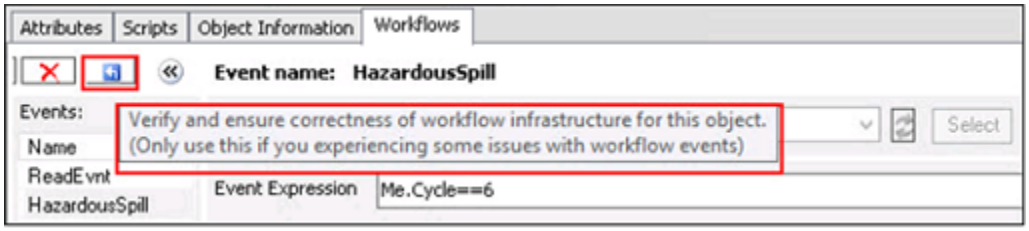
When you use relative references, like *MyContainer*, you can refer to contained objects within that container. For example, a reference to *MyContainer.InletValve.PV* is equivalent to *Tank1.InletValve.PV* in the following hierarchy:

- Tank1: Cannot reference at this level because this is not contained.
- InletValve: Can reference at this level because this object is contained.

Example:

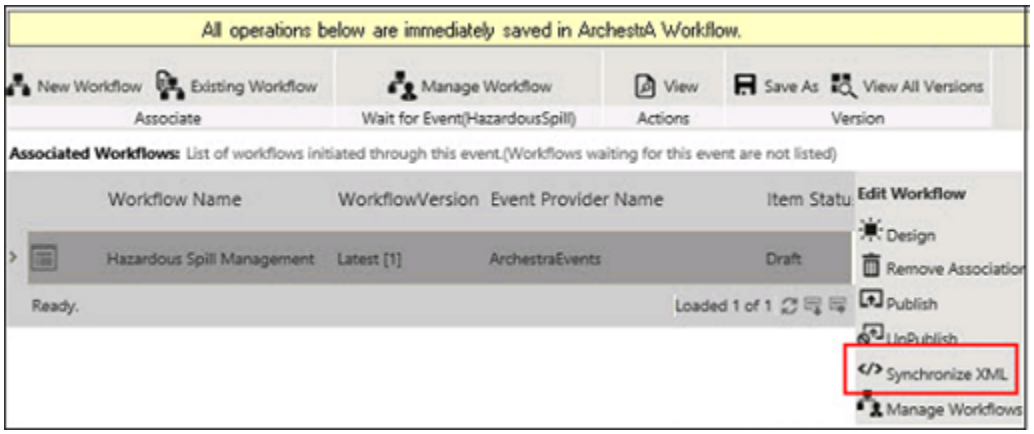
If System Platform-Workflow Connector Version 1.1.1222.0 is used and events to trigger workflows are configured, then do the following steps to use the relative referencing support in the **System Platform-Workflow Connector Version 1.5.1229** for the existing events:

1. Select an existing event, and then click **Verify** to ensure correctness of workflow infrastructure for this object icon.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Select a workflow, and then click the **Synchronize XML** option for that workflow.



3. Deploy the existing gateway object again.

Custom Activity

Follow the steps below to create Custom activity in Quickflow:

1. Modify the action xml or add the new action xml and then add the Quickflow related information as shown below

```
<action name="Custom Action" supportforquickflow="true">
<quickflow>
<definitionProvider class="CustomAction.CustomActivityQuickflowDefinitionHandler"
assembly="c:\CustomAction.dll" />
<runtimeHandler class="QuickflowCustomActivity.CustomActivity"
assembly="c:\netcoreapp2.2\QuickflowCustomActivity.dll" />
</quickflow>
<description>Execute Custom Act</description>
<image resourcename="dummyaction.png" />
<handler classname="CustomAction.CustomActionHandler" assembly="bin\CustomAction.dll"
/>
<properties>
<property category="cat_configuration" name="CustomProp" displayname="Custom
Property" type="string" helpstring="Build Property" mandatory="false"
supportforquickflow="true"></property>
</properties>
<return>
<value helpstring="Set">Set</value>
<value helpstring="Not Set">Not Set</value>
</return>
</action>
```

2. Create a class library targeting .Net Framework 4.8 which can read all the activity related information from the database. In the above example, CustomAction.CustomActivityQuickflowDefinitionHandler is a class which creates the parameters needed for the execution.
3. Decorate the class with the attribute DefinitionHandler(). Make sure the name passed as the argument to this attribute is same as the name mentioned in the action xml.
4. Refer the following assemblies. These can be found in the GAC & in the installation path (Bin)
   a. Skelta.Bpm.Actions.Core.dll

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

   b. Skelta.Bpm.Quickflow.Bridge.dll
5. Provide the class name and the assembly path under the definitionProvider node.
6. Create a class library targeting .Net 6 which executes the activity. In the above example, QuickflowCustomActivity.CustomActivity is a class which executes the activity.
7. Refer the following assemblies. These can be found under the <InstallPath>\Web\Quickflow folder.
   a. Skelta.Bpm.Actions.Core.dll
   b. Skelta.Bpm.Quickflow.Interfaces.dll
   c. Skelta.Bpm.QuickFlow.Logger.dll
8. Decorate the class with the attribute RuntimeHandler (). Make sure the name passed as the argument to this attribute is same as the name mentioned in the action xml.

Process Designer

This section provides information on how to export and import a process definition. It also explains how to use Expression Editor to build a wide variety of expressions with different types of objects.

- Export a process definition
- Import a process definition
- Build an expression

Export Process Definition

You can export a process definition as an XML file. This makes the process design portable and accessible from other applications.

To export process definition as an XML file

1. From the **Tools** menu, select the **Export** option.
2. Click **Save**.

   The **Windows Save As** dialog box appears
3. Click **Save** to save the XML file. You can also browse to a new location for saving the file or change its name before saving it.
4. In the **File Download** dialog, click **Open** to view the XML process definition.

See also, Export.

Import Process Definition

You can import a process design saved as valid XML files. This can be used to import and modify a process design to create a new process design.

To import process definition XML file

1. From the **Tools** menu, select the **Import** option.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. In the window that opens, enter the name of the XML Process definition file in the field provided. Alternatively, click on the **Browse** button to select a valid process definition XML file.
3. After selecting a process definition XML file, click on the **Import** button to import it.

   This imports the selected XML process definition.

See also, Import

Note:
- The workflows with activities not supported in the Quickflow cannot be imported.
- The workflow with properties not supported in Quickflow can be imported.

Build Expression

Expression Editor

The Expression Editor is a web control, which is closely integrated with the Process Designer for building a wide variety of expressions with different types of objects. The Expression Editor supports only the C# syntax for building expressions.

Note: Ensure that you do not create or use any HTML tag in the Expression Editor.

The Expression Editor window is shown in the image below. It has the Object Browser at left. This panel can be displayed or hidden using the toggle at the top. A refresh button is also available to refresh the contents. The default objects available in the Object Browser are **Date**, **Math**, **String**, and **Workflow Instance (XmlVariables** and **Variables)**. Each node expands to display a list of methods, properties and functions that can be used to build expressions.

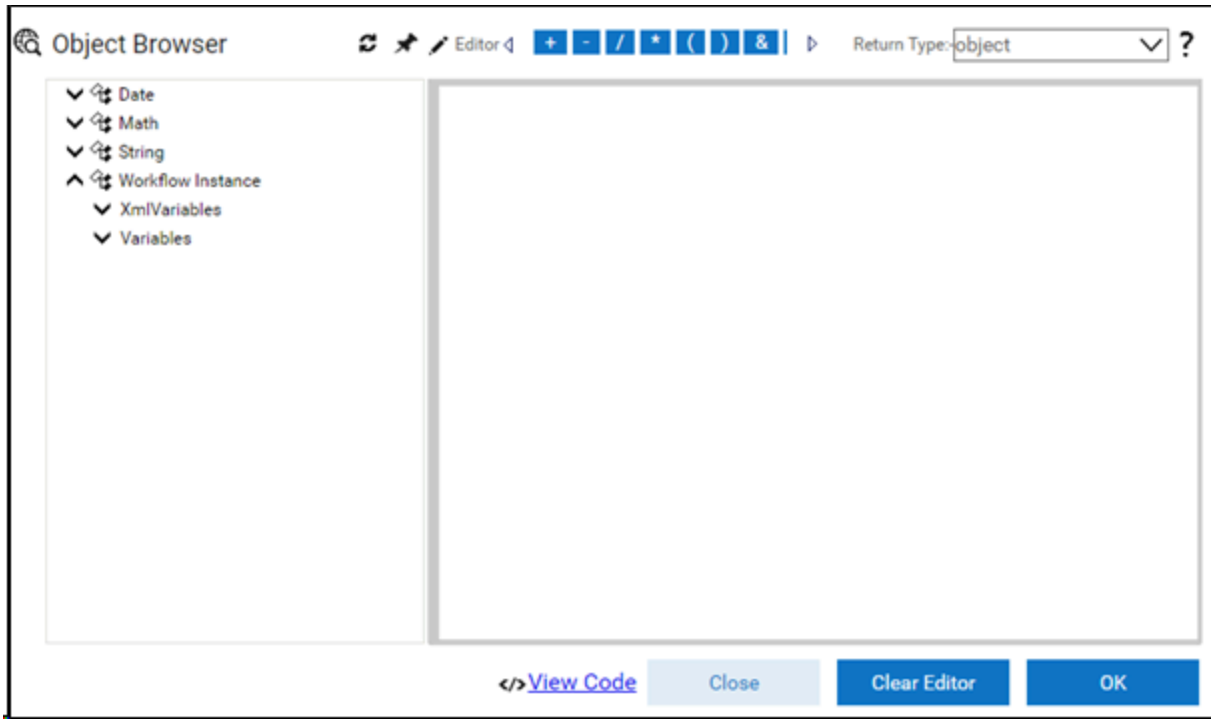
The Editor is located at the right of the window. It has buttons to add the following operators at the top: **+, -** , **/**, *****, **(**, **)**, **&**, and **|**.

The **OK** button at the bottom of the Expression Editor window can be used to save the expression.

The **Close** button can be used to exit the window without saving the expression.

The **Return Type** field to the right enables selection of the value type to be returned.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Support

This section provides information on how to run the diagnostics tests and determine if AVEVA Work Tasks is functioning properly. It also lists the limitations and provides troubleshooting information.

- Diagnostics
- Limitations
- Troubleshooting

Diagnostics

The **Diagnostics** feature allows you to run the diagnostics tests and determine if AVEVA Work Tasks is functioning properly.

You can select the services and run the diagnostics test to know the status of the selected services. The status of the selected services is displayed in the Logs section of the Diagnostics page. You can also filter the logs based on Pass or Fail status of the services, export the Logs, and maximize the Logs view.

You can select the Log levels to display the relevant messages. The following Log levels are available:

Level 1: This level shows minimal information in the Logs section. The Logs section displays information related to the start and end of a workflow. Error messages are also shown in this mode.

Level 2: This level provides detailed information of background activities. It will show all the messages logged from the actions.

Level 3: This level displays messages indicating whether an action is executed.

The following services can be diagnosed to check their availability in AVEVA Work Tasks:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Mobile Notification Service**: Tests whether Mobile Notification Service is running.
- **Azure Service Bus Relay**: Tests whether Azure Service Bus Relay is configured and the response is received from Http Web Request.
- **Message Broker (amqp/amqps)**: Tests whether the repository is provisioned and the Message Broker (amqp/amqps) exist in the repository.
- **Quickflow**: Tests whether Quickflow is deployed, AMQP protocol is validated on the Message Broker, and Quickflow Host is connected.
- **Service Communication**: Tests whether Enterprise Server service and all the AVEVA Work Tasks services are running.

To run the diagnostics test

1. On the Windows **Start** menu, point to **Programs**, point to **AVEVA**, point to **Work Tasks**, and then **Central Configuration**.

   The AVEVA Work Tasks **Central Configuration** page appears.
2. Click **Diagnostics Settings**.

   The **Diagnostics** page appears.
3. Select the check box against the items for which you want to run the diagnostics tests and then click **Test** button.

   After the diagnostic test is complete, the status of the selected items appears in the **Logs** section of the **Diagnostics** page.

To reset IIS

After you change the log level, click the **Reset IIS** button in the **Diagnostics** page to reset IIS and update the log level. On IIS reset all the AVEVA Work Tasks sites will get restarted.

To restart all services

After you change the log level, click **Restart Service** button in the **Diagnostics** page to restart all the AVEVA Work Tasks services and update the log level.

Note: If you modify the log level, ensure to restart services and reset IIS to update the log level.

Limitations

- Quickflow does not support XML Variables of XML List type. Hence, Archestra Workflow Information list will not be updated with the event information for Quickflow.

Troubleshoot

The following error message is displayed when Quickflow is not deployed properly.

Error while deploying Quickflow in one of the machines. For more information, refer the logger console and Troubleshooting section in Quickflow help document.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Possible Causes

- The certificate generated by System Management Server has been used for both Quickflow and AIM.
- The port number is used by other application.
- Connectivity to Message Broker must have failed.

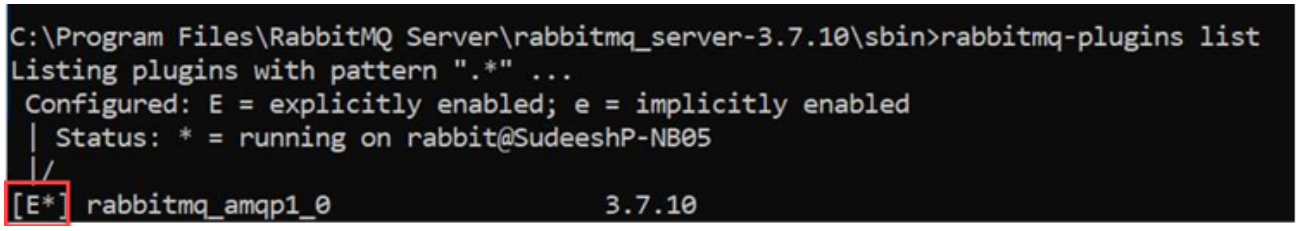
Resolution

- Ensure the AIM certificate and the Quickflow certificate are not same.
- Ensure the port number is not 443 if AIM is installed on the same machine.
- Ensure Message Broker Connection is proper. Refer, *Resolving Message Broker Connection* below to resolve the connectivity failure.

Resolving Message Broker Connection

- Verify whether the AMQP plugin is enabled. Run the command in the Command Prompt:

  `C:\Program Files\RabbitMQ Server\rabbitmq_server-3.7.10\sbin>rabbitmq-plugins list`

- If AMQP plugin is not enabled, run the command:

  `C:\Program Files\RabbitMQ Server\rabbitmq_server-3.7.10\sbin>rabbitmq-plugins enable rabbitmq_amqp1_0`

- Ensure all the listeners are listed in the **Configure RabbitMQ Listener Details** page. To ensure all the listeners are listed, run the command:

  `C:\Program Files\RabbitMQ Server\rabbitmq_server-3.7.10\sbin>rabbitmqctl status`

While executing the above commands displays any error like “node unreachable”, follow one of the steps:

- Restart the RabbitMQ windows service.
- Restart the machine where RabbitMQ is installed.
- If none of the above steps resolve the error, repair/reinstall RabbitMQ.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

User Guide

The *User Guide* provides instructions how to create and manage AVEVA Work Tasks key components, such as workflow, calendars, queues, and work items. This guide also provides tips and warnings that you need to be aware of while using AVEVA Work Tasks.

It assumes you have a working knowledge of the principles and customary practices of your business area.

Administration

The *Administrator Guide* provides detailed instructions to:

- Install and activate the product
- Configure the product
- Set up repositories
- Common system administration procedures
- Uninstall the product and the connectors

For more information about installing and configuring AVEVA Work Tasks, see *Administrator Guide*.

Prerequisites

The prerequisites to access the features of **AVEVA Work Tasks** are as follows:

1. Install AVEVA Work Tasks.
2. Create Repository from the Central Configuration Site and define your organization as AVEVA Work Tasks Repository. For example, if you are designing workflows for Telematics organization, define your Repository as *Telematics*. A **Repository Database** contains workflow definitions, queues, calendars, settings, and workflow execution data for an application. For more information about Creating and Managing Repositories, see *Administrator Guide*.

Get Started

This section provides information that will help you to get started with using AVEVA Work Tasks by creating and using workflows and forms.

Create a Workflow

In this article you will learn how to create your first AVEVA Work Tasks workflow. The "Hello World” workflow that you create will send an information to the workflow initiator with the subject line “Hello World”.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Prerequisites

- Download and install the latest AVEVA Work Tasks version.
- Create a repository.
- Log on to the repository using Administrator credentials.

This topic can be broadly categorized into:

- Designing a Workflow
- Publishing and Executing a Workflow
- Viewing the information in the actors inbox

To design a Workflow

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, and then click **Workflows**.
   The **Workflows** page appears.
3. On the action bar, click **New,** and then click **Workflow.**
   The **New Workflow** wizard appears.
4. In the **Title** box, type a name for the workflow, and in the **Description** box, type a description for the workflow.
5. Click **Save & Continue**.
   The workflow is created, and the **Workflow Options** wizard appears.
6. Do one of the following.
   a. Design the workflow
      i. Click **Design the Workflow now**.
      ii. Click **Finish**. The **Process Designer** page appears.
      iii. Design the workflow.
   b. Cancel further operations.
      i. Click **No task to be performed now**.
      ii. Click **Finish**. The workflow is saved.
7. Open the workflow in **Process Designer**.
8. Drag an **Information** activity from the **Human Activities** to the design area.
9. Right-click **Information activity**, and then click **Activity Properties**.
   The **Activity Properties** window appears.
10. Configure the **Information** activity properties.
    a. Enter a **Name** and **Description** for the activity.
    b. Click the edit icon near the **Assign Actor(s)** property.
       The **Assign Actor(s)** appears.
    c. You have to assign the Information task to the user who has triggered the workflow. For this, you have to specify a small expression:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

i. Select the required provider from the **Providers** list.
ii. Select **ID** from the first drop-down list.
iii. Keep the operator as default "=".
iv. Select **Variable** from the next drop-down list.
v. Select **Variable.SubmittedBy.ID** in the last drop-down list and click **Add**.
vi. Click **Update** to save the property value.

d. Configure the **Subject** property:
i. Click the edit icon near the **Subject** property. The **Subject** appears.
ii. Enter the subject in the text area as "Hello World!".
iii. Click **Update**.

e. Configure the **Body** property:
f. Click the edit icon near the **Body** property. The **Body** page appears.
g. Enter the subject in the text area as "This is my first Information Activity", and then click **Update**.
h. Click **Update**.

11. Click **Save** in the **Activity Properties** pane to save all the data entered.
Every time you input a value for a property, ensure to save the data. Click the **Save** button on the **Activity Properties** window. You need to save the property so that the subsequent properties in the activity can inherit the required information.
Click the **Link** icon to add a link. The **Link Action** appears.
12. Select the **From Action** as **Start** (the name of the activity from which the link begins) and To Action as MyNewInformationActivity (the name of the activity at which the link ends).
13. Click **OK** to establish the link. Your first workflow is created.

To publish and execute the workflow

1. Click **Publish** from the menu of the Process Designer.
A message appears to confirm that the workflow is published successfully.
2. Click **Close**.
The **Process Designer** closes and the **Workflows list** appears. The list now displays one new workflow.
3. Right-click **MyFirstWorkflow**, and then click **Publish and Test Run**.
This option publishes the workflow and then test runs it. You can skip step 1 and directly proceed to this step.
4. The **Process execution view** appears where you can see the execution status.
You will notice that the **Start Activity** and the **Information Activity** are executed successfully.

Note: If the **Status** is **"Execution Pending"** in the Process execution view window, then click the **Refresh** button to get the latest status.

To view the Inbox

1. In the **Enterprise Console**, click menu icon, and then click **Inbox**.
In the **Inbox** you will find an information task with the Subject "Hello World!".

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Click on the **Information** task in the Inbox.

   The details of the work item appears.

   You have successfully created your first workflow.

Use Variables in Workflow

AVEVA Work Tasks supports storage types that are mainly categorized as **Variables**, **Contents**, and **XmlVariables.**

- Create a variable
- Assign a value to the variable
- Use the variable in an activity

Prerequisites

- Download and install the latest AVEVA Work Tasks version.
- Create a repository.
- Log on to the repository using Administrator credentials.

To create a Variable

1. Log on to the **Enterprise Console**.
2. In the **Enterprise Console**, click menu icon, and then click **Workflows**. The **Workflows** page appears.
3. On the action bar, click **New**, and then click **Workflow**. The **New Workflow** wizard appears.
4. In the **Title** box, type a name for the workflow, and in the **Description** box, type a description for the workflow.
5. Click **Save & Continue**. The workflow is created, and the **Workflow Options** wizard appears.
6. Select **Design the Workflow now**, and then click **Finish**. The **Process Designer** appears. While designing a workflow, if you get a pop-up blocker warning message, then to select Allow Blocked Contents to proceed further.
7. Right-click the **Start** activity, and then click **Activity Properties**. The **Activity Properties** appears.
8. Click the edit icon near the **Variables** property to declare the variable. The **Variables** dialog box appears. All Variables added to the workflow are displayed in rows in the grid.
9. Enter "EmployeeName" in the **Workflow Variables** text box, and then select the **Type** as "String".
10. Click **Update** to save the variable.
11. Click the **Save** button in the **Activity Properties** pane to save all the data entered.

To update the Variable

To update the value of the variable in the subsequent activities, we have to add the Update Variable activity from the Engine Activities list.

1. Select **Engine Activities** from the **Activities** drop-down list.
2. Drag the **Update Variable** activity to the Design area.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Right-click the **Update Variable** activity in the Design area, and then click **Activity Properties**.
4. Configure the Update Variable activity.
   a. Enter a **Name** and **Description** for the activity.
   b. Configure the *SetVariable* property.
      i. Click the edit icon in the Set Variable property to set a value for the variable. The Update Variable dialog box appears. Click Select. The tree view containing the list of Variables and XmlVariables is seen.
      ii. Click **Select**. The tree view containing the list of Variables and XmlVariables is seen.
      iii. Click the icon near the Variables view. You will see the variable that we created in the Start activity.
      iv. Click the variable “Variable.Employee” and enter a value for it. Enter the name “John”.
      v. Click **OK** to update the value assigned
   c. Click the **Save** button in the Activity Properties pane to save all the data entered.
5. Click the **Link** icon in the toolbar to link the two activities, that is the Start and the Update Variable.

   The **Link Action** appears.
6. Click **OK** to establish the link.

To use the variable in the activity

1. Select **Human Activities**.
2. Drag the **Information** activity to the Design area.
3. Right-click the **Information** activity in the Design area, and then click **Activity Properties**.
4. Configure the Information activity.
   a. Enter a **Name** and **Description** for the activity.
   b. Configure the Actor to whom you want to send the information:
      i. Click the edit icon near the **Assign Actor(s)** property. The **Assign Actor(s)** appears.
      ii. You have to assign the Information task to the user who has alerted the workflow. For this, you have to specify a small expression:
      iii. Select the required provider from the **Providers** list.
      iv. Select **ID** from the first drop-down list.
      v. Keep the operator as default "=".
      vi. Select **Variable** from the next drop-down list.
      vii. Select **Variable.SubmittedBy.ID** in the last drop-down list, and then click **Add**.
      viii. Click **Update** to save the property value.
   c. Configure the Subject property:
      i. Click the edit icon near the Subject property. The Subject appears.
      ii. Enter the subject in the text area.
      iii. To customize the subject and append the subject by selecting some XML variable and building an expression follow the next set of steps:
      iv. Type 'The Employees Name is:' in the text area.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

v. Select **Variables** from the first drop-down list.
vi. Select the Variable you had declared in the **Start** activity from the second drop-down list, that is EmployeeName.
vii. Click **Append**.
viii. Click **Update**

d. Click the **Save** button in the **Activity Properties** to save all the data entered.

5. Click the Link icon to add a new link between the two activities, "EmployeeVariableUpdation" and "DisplayingEmployeeInformation".
6. Click **OK** to establish the link.
7. Click **Publish and Test Run**. The **Process execution view** appears where you will notice the Start, Employee Variable Updation, and Display Information activity has been executed successfully.

Note: If the Status is "Execution Pending" in the Process execution view, then click the **Refresh** button to get the latest status.

To view the Inbox

To view the information task assigned to the actor who alerted the workflow.

1. Select the **Inbox** from the **Enterprise Console** menu. A task with the subject “The Employee name is John” is available in the users **Inbox**.
2. Click the **Information** task in the Inbox to view the details for the task. You have successfully created the **VariableWorkflow**.

AVEVA Work Tasks provides you other storage types, like Contents and XmlVariables.

For more information about using variables, see the *Start Variables* topic in the *Developer Guide*.

Use XML Variables in Workflow

You can create an XML Variable, loop through the XML across the document, and use XML Variable in an activity.

- Familiarizing with Workflow Management XML Variables
- Loop Through an XML Document

Prerequisites

- Download and install the latest AVEVA Work Tasks.
- Create a repository.
- Log on to the repository using administrator credentials.
- Basic knowledge of how to create a simple workflow.

Familiarize with XML Variables

Perform the following steps to familiarize with AVEVA Work Tasks XML Variables:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Create an XML Variable and define the schema
- Updating the XML Variable
- Display the value in an Information activity

To create an XML Variable and defining the schema

1. Create a workflow.
2. Open the **Process Designer**.
3. Right-click the **Start activity**, and then click **Activity Properties**. The **Start Activity Properties** appears.
4. Click the icon near the **XmlVariables** property in the activity properties to declare the variable.

   The XmlVariables window appears. All XML Variables added for the workflow are displayed in rows in the grid. The variable properties that are specific to an XML Variable are displayed in the Variable Properties section for an XML Variable.
5. Click **New Variable** to add a variable. The **Variable Name** is enabled.
6. Enter the name of the new XML Variable in the **Variable Name** field and a description in the **Description** field.
7. Select `XMLDocument` from the **Type** list. For more information about XML Variables and the different types, see XML Variables topic in *Developer Guide*.
8. Build the XML Schema in the XML Variable interface in the following format:

```
<Employee>
<Name>
</Name>
<Department>
</Department>
</Employee>
```

9. Click the icon near the “root” element. In the dialog box that appears, click **Edit** to edit the root element.
10. Enter the Name as “Employee”, and select the Type as “element”. Click **OK** to save the element value.
11. Click the icon near the **Employee** element, and then click **Add** to add the child elements.
12. Enter the *Name* in the **Name** field, and select **Type** as element. Click **OK** to save the data.
13. Similarly, add one more element **Department** to the XML Schema.
14. Select the type for **Internal Storage** as **Variable** since the value for this XML Variable has to be updated and stored as a variable.
15. Click **Save Variable** to save the newly created XML Variable.

To update the XML Variable

To update the value of the XML Variable for the subsequent activities you have to add the Update Variable activity from the Engine Activities list.

1. Select the **Engine Activities**.
2. Drag the **Update Variable** activity to the **Process Design** area.
3. Right-click the **Update Variable** activity, and then select **Activity Properties**.
4. Enter a **Name** and **Description** for the activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Configure the **Set Variable** property as explained:
   a. Click the icon near the **Set Variables** property. The **Update Variable** appears.
   b. Click **Select**. You can see a tree view containing the list of Variables and XmlVariables.
   c. Click on near XmlVariables. Now you will be able to see the XmlVariables that was created in the section **Creating XML Variables and defining XML Schema**.
   d. Click on the node **Name** and enter a value for the variable. For example, enter the name "John".
   e. Click on the node **Department** and enter a value for the variable. For example, enter the department as "Research".
   f. Click **OK** to update the values assigned
   g. Click **Save** in the **Activity Properties** pane to save all the data entered.

To display the XML Variable value

1. Select **Human Activities**.
2. Drag **Information** activity to the **Process Design** area.
3. Right-click the **Information** activity, and then select **Activity Properties**.
4. Enter a **Name** and **Description** for the activity.
5. Configure the Actor to whom the information has to be sent:
   a. Click the icon near the **Assign Actor(s)** property. The **Assign Actor(s)** appears. You have to assign the Information task to the user who has alerted the workflow. For this, you have to specify a small expression.
   b. Select the required provider from the **Providers** list.
   c. Select **ID** from the first drop-down list.
   d. Keep the operator as default =.
   e. Select **Variable** from the next drop-down list.
   f. Select **Variable.SubmittedBy.ID** in the last drop-down list, and click **Add.**
   g. Click **Update** to save the property value.
6. Configure the **Subject** property as explained:
   a. Click the icon near the **Subject** property. The **Subject** appears.
   b. Enter the subject in the text area. To customize the subject and append the subject by selecting some XML variable and building an expression follow the next set of steps.
   c. Type **'The Employees Name is:**' in the text area.
   d. Select **XML Variables** from the first drop-down list.
   e. Select the XML Variable you had declared in the **Start** activity from the second drop-down list, that is **Employee**.
   f. Select the **Name** node from the **XML Schema** in the third drop-down list, and click **Append**.
   g. Similarly, append the **Department** value. Now you have the Subject as:
   h. Click **Update**.
7. Click the **Save** button in the **Activity Properties** to save all the data entered.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

8. Click the **Link** icon in the tool bar to create the following links:
   a. Start Activity to **Employee XML Variable Updation** activity
   b. **Employee XML Variable Updation** activity to **Displaying Employee Information** activity.
9. Right-click on the link between **Employee XML Variable Updation** activity to **Displaying Employee Information** activity, and then select the set the link the output value as **Updated**.

To publish the Workflow

1. Click **Publish and Test Run** from the menu bar. The **Process Execution** page appears.
2. Verify all the activities that have been executed successfully.

Note: If the Status is "Execution Pending" in **Process execution**, then click the **Refresh** button to get the latest status.

View the Inbox

To view the Information task assigned to the actor who alerted the workflow, select the **Inbox** from the **Enterprise Console**. You have successfully created a workflow that uses an XML Variable and implemented it in your activity.

Reference topic

Loop Through an XML Document

Loop Through an XML Document

Perform the following steps to loop through the elements of an XML Document:

- Creating an XML Variable and loading a predefined schema
- Creating an XML Variable of type Xml Iterator and looping through the Schema
- Adding a script activity to create the XML and get the data
- Looping through the Elements of the XML Document
- Displaying the Value of the XML Iterator

To create an XML Variable and loading a predefined schema

1. Create a new workflow, *LoopXmlVariableWorkflow*.
   The **Process Designer** window appears.
2. Right-click the **Start** activity, and then click **Activity Properties**.
   The **Activity Properties** window appears.
3. Click on the edit icon near the **XmlVariables** property.
   The **XmlVariables** window appears.
4. Click **New Variable** to add a new variable.
5. In the **Variable Name** field, enter the name and description for the variable.
6. Select **XmlDocument** from the Type list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. Create an XML file **EmployeeData.XML**.

   This file is required to build the XML Schema.

8. Copy the following schema to create the XML file.

```
<Employees>
<Employee>
<Name>
</Name>
<Department>
</Department>
</Employee>
<Employee>
<Name>
</Name>
<Department>
</Department>
</Employee>
</Employees>
```

9. Click **Browse** and load **EmployeeData.XML** file from the folder where it is saved.
10. Click **Load** to load the schema.

    The XML Variable of type XML Document has been created.

    You will see the schema in the **Variable** properties area as in the XML document.

11. Click Save Variable to save the variable.

To create an XML Variable of type XmlIterator and looping through the Schema

Use XML Variable of type XmlIterator to define variables for repeating XML elements and attributes. In this scenario, *Employee*, *Name*, and *Department* are repeating elements.

1. Create an **XML Variable** of type **Xml Iterator**.
   a. Click **New Variable** and enter the name in the **Variable Name** field.
   b. Select the type as **Xml Iterator**.

      The **Xml Iterator** type option is available only if another variable of type Xml Document exists. Two additional properties appear.

   c. Select the parent **XML Variable** as *Employees*.

      The parent variable of an **Xml Iterator** variable can be another Xml Iterator Variable also.

   d. Select **Node** as *Employee*.

      The node should represent the variable. Only nodes in the parent variables (that is, Employees) are displayed.

   e. Click **Save Variable** to save the variable.

      The XML Variable has been created.

To add Script Activity

In a real time application, you can get XML Data by different means such as invoke form, from a Web Service, or through Script activity. In this scenario you will create an XML using a Script Activity.

1. Drag a **Script** activity from the **Integration Activities** list to the **Process Designer** area.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Right-click on the **Script** activity in the **Design** area and select **Activity Properties**. The **Activity Properties** appears.
3. Enter a **Name** and **Description** for the activity.
4. Select the **Programming Language** as **CSharp.NET** in the **Language** property.
   a. Click on the edit icon in the **Code property** to write the code.

   To set the value for the XML, you can use the following sample code:

```
//Sample CSharp code
using System;
using Workflow.NET;
using Workflow.NET.Engine;
public class WorkflowScript59e4b2aadf064757a8089c95c630794f
{
public void Run(int ExecutionId, int ExecutionDetailsId,
Workflow.NET.Engine.Context ctx, Workflow.NET.Action action, string inlink)
{
string XmlValues = "<Employees><Employee><Name>Tim</Name><Department>Sales";
XmlValues +=
"</Department></Employee><Employee><Name>John</Name><Department>Marketing";
XmlValues += "</Department></Employee></Employees>";
ctx.XmlVariables["Employee"].RawXml = XmlValues;
ctx.SaveVariables();
}
}
```

   a. In this script action you are building the values for XML Schema as per the Schema defined for the XML Variable *Employee* in the Start activity, that is,

```
string XmlValues =
"<Employees><Employee><Name>Tim</Name><Department>Sales";
XmlValues +=
"</Department></Employee><Employee><Name>John</Name><Department>Marketing";
XmlValues += "</Department></Employee></Employees>";
```

   a. *ctx* is the Workflow.NET.Engine.Context object which encapsulates all the information about the executing workflow. The Context object will have all the Variable, Content, XML Variable Details in the runtime.

   b. Assign the runtime value to the XmlVariable *Employee* as:

```
ctx.XmlVariables["Employee"].RawXml = XmlValues;
```

   ctx.SaveVariables(): Persists the assigned value to the XML variable to the database. After calling the *ctx.SaveVariables()*, the value of the XML Variable will be available in the runtime for the workflow.

   Now the XML Variable *Employee* will contain the details of two employees, Tim and John (in XML format) which can be used in the workflow.

   b. After writing the code, click **Update** to save it.

```
using System;
using Workflow.NET;
using Workflow.NET.Engine;
public class WorkflowScript50e4b2aadf064757a8089c95c630794f
{
public void Run(int ExecutionId, int ExecutionDetailsId,
Workflow.NET.Engine.Context ctx, Workflow.NET.Action action, string inlink)
{
string XmlValues = "<Employees><Employee><Name>Tim</Name></Department>Sales";
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
XmlValues +=
"</Department></Employee><Employee><Name>John</Name><Department>Marketing";
XmlValues += "</Department></Employee></Employees>";
ctx.XmlVariables["Employee"].RawXml = XmlValues;
ctx.SaveVariables();
}
}
```

To loop through the Elements of the XML Document

The **For-Each Loop Activity** is used loop through the collection within an **Xml Iterator Variable**. It allows repeated execution of an activity or a group of activities.

1. Drag the **For-Each Loop** activity from the **Engine Activities** list to the **Process Designer** area.
2. Right-click on the **For-Each Loop** activity in the **Design** area and click **Activity Properties** from the shortcut menu.

   The **Activity Properties** appears.
3. Enter a name and description for the activity
4. Select **EmployeeIterator** from the **XML Iterator** property list.
5. Click **Save** to save all the data entered.
6. Link the **Script** activity to the **For-Each Loop** activity.

To display the Value of the XML Iterator

To display the looped XML value, an Information activity is required.

1. Select **Human Activities**.
2. Drag **Information** activity to the **Design** area.
3. Right-click the **Information** activity in the Design area and click **Activity Properties** from the shortcut menu.

   The **Activity Properties** appears.
4. Enter a name and description for the activity.
5. Configure the **Actor** to whom the information has to be sent:
   a. Click the edit icon near the **Assign Actor(s)** property.

      The **Assign Actor(s)** appears.
   b. Assign the Information task to the user who has alerted the workflow.

      For this, you have to specify a small expression.
   c. Select the required provider from the **Providers** list.
   d. Select **Id** from the first drop-down list.
   e. Keep the operator as default "=".
   f. Select **Variable** from the next drop-down list.
   g. Select **Variable.SubmittedBy.ID** in the last drop-down list and click **Add**.
   h. Click **Update** to save the property value.
6. Configure the Subject property as explained:
   a. Click the **edit** icon near the **Subject** property.

      The **Subject** appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

b. Enter the subject in the text area.

   To customize the subject and append the subject by selecting some XML variable and building an expression follow the next set of steps.

c. Type **'The Employees Name is:'**. in the text area.

d. Select **XML Variables** from the first drop-down list.

e. Select the XML Variable you had declared in the **Start** activity from the second drop-down list that is, *Employee*.

f. Select the **Name** node from the XML Schema in the third drop-down list and click **Append**.

g. Similarly, append the **Department** value.

h. Click **Update**.

7. Click **Save** to save all the information you have entered.
8. Link the **For-Each Loop** activity and the **Employee Information** activity you have created.
9. Right-click on the link to select the link action. Select **Step** as you have to loop through the XML values and get the value in the **Information** activity.

To link to the Next-Loop Activity

Just as you use the for-loop in normal programming, check whether the count value has been reached. If not, then you have to increment the count.This task is achieved through the **Next-Loop** activity.

Note: For every For-Each loop activity, you need a Next-Loop activity.

The Next-Loop activity will check the count and repeat the execution within the loop. Since you have to loop through the XML variables, add a Next-Loop Activity to the workflow.

1. Drag the **Next-Loop** activity from the **Engine Activities** list to the Design area.
2. Link the **Employee Information** activity and the **Next-Loop** activity.
3. Add a link from the **Next-Loop Activity** to the **For-Each Loop Activity** which completes the loop.
4. Set the link action as **Next Step** which is needed to advance the loop.

To publish the LoopXmlVariableWorkflow

1. Click the **Publish** option from the menu bar.

   A message workflow published successfully appears.

2. Select and right-click the **LoopXMLVariableWorlfow** from the workflow list. Click **Publish and Test Run** from the shortcut menu.

   The **Process execution view** appears wherein you can see the execution status. You will notice that all the activities have been executed successfully.

Note: If the Status is "**Execution Pending**" in the **Process Execution**, then click the **Refresh** button to get the latest status.

To view the Inbox

1. Select the **Inbox** option from the **Enterprise Console** menu.
2. In the Inbox, you can find two Employee Information tasks with the Subject "*The Employee Name is John and Employee Department is Marketing"* and "*The Employee Name is Tim and Employee Department is Sales*. You

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

have successfully created the **LoopXmlVariableWorkflow**.

For more information about XML Variables and their types, see *XML Variables* topic in the *Developer Guide*.

Use Forms

Designing the User Interface is one of the most important aspects of developing a business application. Customized Forms are integral to process workflows and in fact often initiate various organizational workflows. AVEVA Work Tasks Forms simplifies the process of designing forms and linking them to a business process.

The forms that are created are mapped to relevant steps in the process workflow. Data mapping between the form fields and process attributes can easily be associated in the control property sheets. The end users interact with this while performing the tasks, and the data entered in these forms is passed back to and from the process seamlessly using XML.

For more information about Forms, see *Forms and Lists User Guide*.

Invoke Web Service from Workflows

You can invoke a web service from a workflow.

Complete the following steps to invoke a web service from a workflow:

- Invoking a Hello World web service
- Calling the HelloWorld () method from the web service
- Displaying the return value in the Information activity

Prerequisites

- Download and install the latest AVEVA Work Tasks version.
- Create a repository.
- Log on to the repository using Administrator credentials.
- Create a Hello World web service

Understanding Invoke Web Service Activity

Step 1: Creating a Workflow

1. In the **Enterprise Console**, click menu icon, and then click **Workflows**. The **Workflows** page appears.
2. On the action bar, click **New**, and then click **Workflow**. The **New Workflow** wizard appears.
3. In the **Title** box, type the name for the workflow, and in the **Description** box, type the description for the workflow.
4. Click **Save & Continue**. The workflow is created, and the **Workflow Options** wizard appears.
5. Select **Design the Workflow now**, which is the default option, and then click **Finish**. The **Process Designer** appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

In this example, you will invoke a web service that contains a method which displays “Hello World”. To store the return value of the web service, create a workflow variable.

The following steps show how to create a variable in a workflow:

Step 2: Creating a Variable

- Right click the **Start** activity, and then select **Activity Properties**.
- To declare the variable, click on the edit icon of the **Variables** property. The **Variables** dialog box appears.
- Enter the variable name as "NewVariable” and select the **Type** as “String”.
- Click **Update** to save the variable.
- Click **Save** in the Activity Properties pane to save all the data entered.

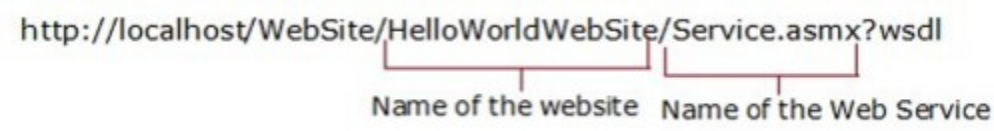
Step 3: Invoke Web Service Activity

To call a web service, you must add an InvokeWebService activity to the Process Designer.

1. Select **Integration Activities** from the **Activities** pane.
2. Drag and drop an **InvokeWebService** activity to the Process Designer.
3. Right click **InvokeWebService** activity to get the Activity Properties pane.

Adding a Web Service

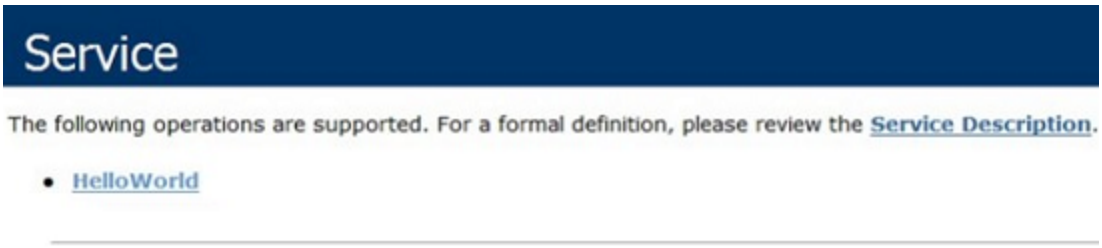
1. Click on the edit icon of the **Web Service Details** property. The **Web Services** dialog box appears. You can add or edit the web services to be invoked in this dialog box.
2. Click **Add** to add a web service.
3. Enter the **WebService Name**. As you would call this web service to display **"Hello World**", you can name the web service as **HelloWorldWebService**.
4. Enter the **WebService URL**. The URL must be the WSDL URL of the web service that you have created. For example, it can be as the following image:



5. If authentication is required, you must select the **Authenticated** check box. This displays additional authentication fields.fields, enter the Login Name, Password, Domain name, and Proxy URL. This step is necessary only if the web service being added requires authentication.
6. In the authentication fields, enter the **Login Name**, **Password**, **Domain name**, and **Proxy URL**. This step is necessary only if the web service being added requires authentication.
7. Click **Save**.

The method associated with the web service is **HelloWorld.** You can get this web method when you view the web service in a browser as shown below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



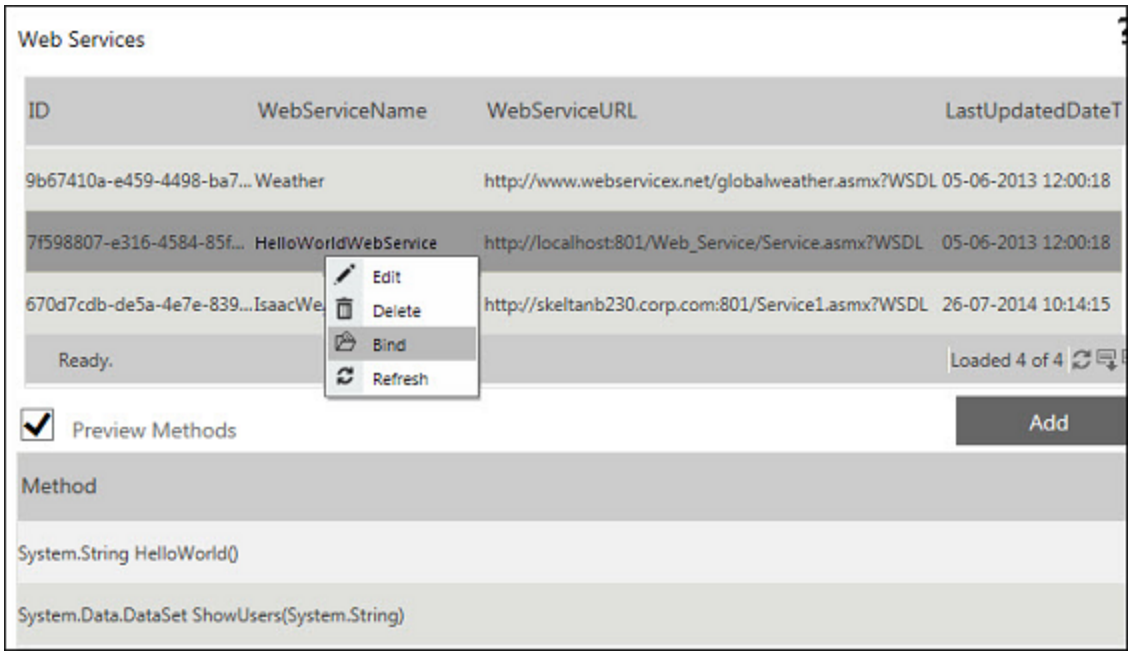
If you invoke this method, you get the output in an XML format as shown in the following image. You can display this in the **Information Activity**.

After you save the details, you can see the web service listed. Click the **Preview Methods** check box to preview the methods in the web service.

Binding a Web Service

1. Right click the web service that you have added. You get options like **Edit**, **Refresh**, **Delete**, and **Bind**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



2. Use the **Bind** option to bind the selected web service to the database. The **Method** list dialog box appears.
3. Click a **Method**, and click the **Bind Parameters**. A window appears which lists out the Parameters to be passed. As the current method just displays a value, the **Parameter** list will be empty.
4. You must save the return value from the Web Method to a variable so that you can use it in subsequent activities. So, select the output type as "Variable" from the Save Output in drop down. Select the variable created in the **Start activity**.
5. Give the **Login Name** and **Password**, if required.
6. Click **Update** to save all the data.
7. In the Activity Properties window, click **Save** to save all the details entered. The **Invoke Web Service** activity is added successfully.

Step 4: Displaying the return value of the Web Service method in the Information Activity

1. Select **Human Activities**.
2. Drag and drop an **Information** activity to the process design area.
3. Right-click the **Information** activity, and then select **Activity Properties**. The **Activity Properties** pane appears.
4. Configure the following properties:
   a. **Name & Description** of the activity. For example, enter name as Display Web Service Data.
   b. The **Actor** to whom this information should be sent. For example, select Id = Variable Variable.SubmittedBy.ID.
   c. The **Subject** of the actual information. For example, select *<%#Variable.NewVariable%>.*
5. Click **Save** in the Activity Properties pane to save all the data entered.
6. Link all the activities, **Start**, **InvokeWebService**, and **Display Web Service Data**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To publish InvokeWebServiceWorkflow

1. Click **Publish** from the menu bar. The **Publish** successful dialog box appears.
2. Clicking **Close**. You will be redirected to the page showing the Workflows list.
3. Run the published workflow. Right-click the workflow, and then click **Publish and Test Run**. The **Process execution view** window appears which shows the execution status of the workflow. You will notice that the **Start**, **InvokeWebService**, and **Display Web Service Data** activities are executed successfully.

Note: If the Status is **Execution Pending** in the Process execution view window, then click the **Refresh** button to get the latest status.

To view the Inbox

To view the information task assigned to the actor who alerted the workflow, see the activity in the Inbox.

1. in the **Enterprise Console**, click menu icon, and then click **Inbox**. The **Inbox** page appears.
2. In the **Inbox**, you can find the **Display Web Service Data** activity with the subject "Hello World".
3. Click the Information task in the Inbox to get a pop-up details view for the task.

   You have successfully created the **Invoke Web Service Workflow**.

Interfaces

AVEVA Work Tasks functions are performed through the following interfaces:

Workflow List: The Workflow List is a list of all workflows defined for a repository. This list provides options for categorizing, designing, and publishing the workflows. Options in the Workflow List are wizard-driven with wizards provided for all the core activities.

About the Process Designer: The Process Designer provides a Visio-type interface with drag and drop functionality for process modeling. For more information about the Process Designer, see About the Process Designer.

Configure Queue Management: The Queue Management window enables creation and management of queues for allocation of workflow activities. To learn more about Queues, see Working with Queues.

Calendar: The Calendar interface enables creation of hierarchical organization and resource calendars. Calendars are used to determine availability of resources, calculate time outs and perform other scheduling functions.

Work Items List: The Work Item List is a web-based interface to which workflow activities are delivered. To learn more about the Work Item List, see Working with Work Item List.

Business Activity Monitoring Alert: The BAM Dashboard enables monitoring of Workflow execution and provides an array of real-time reports on Workflow status and performance. To learn more about using the Business Activity Dashboard, see Working with the Business Activity Monitoring dashboard.

Access Features

AVEVA Work Tasks features and interfaces can be accessed through the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Enterprise Console

The Enterprise Console is a central portal with options for workflow design, workflow management, user creation, BAM analysis, work item lists, queues and calendar management. The Enterprise Console is mapped to one or more AVEVA Work Tasks Repositories. You can gain access to all the above-mentioned functions for a specified repository by logging into the Enterprise Console. The Enterprise Consoles are created from the Farm Server Configuration Manager and mapped to repositories in the Repository Settings page of the Central Configuration.

See *Administrator Guide* for detailed information about creating and configuring Enterprise Consoles and Repositories.

See Workflow Management Enterprise Console for working with the Enterprise Console.

Embedding controls in Applications

The Controls for Process Designer, Queue, Calendar, BAM, and Work List can be embedded into an ASP.NET application.

Enterprise Console

The Enterprise Console is a Web-based interface that allows authorized users to build workflow driven applications, and to participate in these applications. Organizations can create and access their business applications using Enterprise Console.

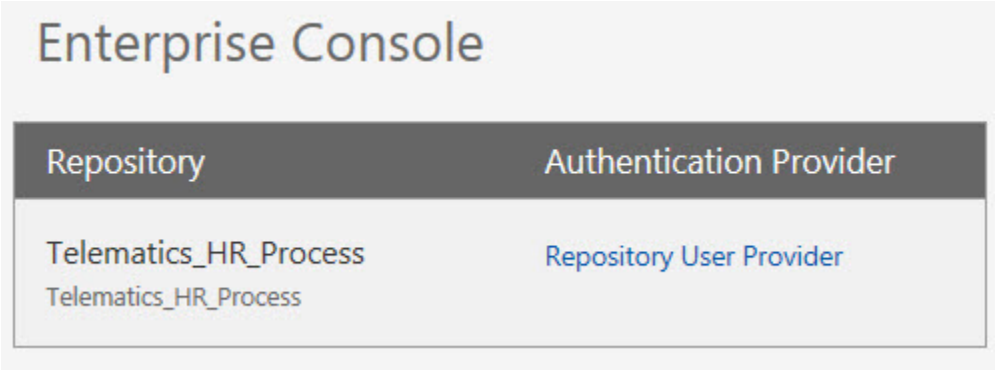
Enterprise Console Authentication

Users, who are a part of the user providers for a repository, can log on to any of the Enterprise Consoles mapped to the repository. For more information about setting up user providers for a repository, see *Administrator Guide*.

To log on to the Enterprise Console:

1. Select the user provider from which you would like to log on to the repository.

   For example, click **Repository User Provider**.



2. Enter **User Name** and **Password** in the **Login** page.

   For the default user name and password, contact the technical support team or refer the *Get Started Guide* in the ISO file of AVEVA Work Tasks.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To protect from unauthorized access or to restrict security vulnerability, the password is not saved even if the remember password option of the browser used has the option to save passwords automatically.

3. Click **Login**.

Note:
- If the Active Directory is selected as Resource Provider and ASTS is also selected, then the authentication is performed by ASTS.
To access the same website from a different machine, you can use the IPv6 address without the scope ID in all the supported browsers.
Example:
- IPv6 with Scope ID = fe80::91fb:becc:2c79:948b%14
- IPv6 without Scope ID = fe80::91fb:becc:2c79:948b

Enterprise Console Home Page

The Enterprise Console home page comprises the following sections:

Work items list:

The Work items list displays a list of activities assigned to the logged in user by workflows in the repository which the user has logged into. The Work items list has an Inbox view and permits users (also known as actors) to respond to the activities assigned.

Enterprise Console menu:

Note: The new UI/UX enhancement to align with Skyline User Interface standard for the navigation menu is applicable to menu items only up to five levels.

The Enterprise Console left menu has the following options:

Dashboard:

This is the default view when a user logs on to the Enterprise Console using the administrator rights. The Dashboard provides an easy-to-read graphical display of webparts, such as the Inbox and Business Activity Monitoring. You can customize the Dashboard according to their specific business requirements.

Inbox:

This option shows the Work items list.

Workflows:

Selecting **Workflows** displays the workflow list which has options to create Workflow, List Workflow, FileWatcher Workflow, BAM Workflow, and Folder. The workflow list also provides options to view, delete, design, edit, publish, manage, and associate workflows and change owner information. You can also apply security at this list level.

Fill Form:

This option allows the user to fill form. It has a security feature that restricts access to the users who can fill the form.

Reports:

Selecting Reports displays the Business Activity Monitoring dashboard with real-time reporting on workflow execution.

Tools:
This option provides link to manage lists, forms, work items, and views. It also has the SOA assembly, form

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

controls, external connections, and ArchestrA Events links.

Settings:

The Settings group has options for Database Connections, Calendar, Queue Management, Users, Alternate Actor, Impersonation User, File Watcher, SOA Folder, Lookup Settings, Global Work Item Layout, Data Grid, and Reporting Service.

- The **Database Connections** option enables the user to create a new data source connection, edit and delete an existing data source connection, and manage security.
- The **Calendar** option enables setting up of multi-level organization and resource calendars.
- The **Queue Management** option enables setting up of queues.
- The **Alternate Actor** option enables assigning an alternate actor for acting on a task or an activity that was assigned to a primary actor. For more information about how to set an alternate actor, see Alternate Actors.
- The **Users** option enables user maintenance tasks such as adding new users to the Enterprise Console, editing and deleting existing users, and managing security.
- The **Impersonation User** option enables you to create new impersonation users which will have domain account users with full rights.
- The **File Watcher** option enables the user to continuously monitor one or more folders for incoming files and raises events like created, updated, renamed and deleted.
- The **SOA Folder** option enables user to register a folder, delete a folder, disable an enabled folder, enable a disabled folder, and manage security settings.
- The **Lookup Settings** option enables the user to create a new database.
- The **Global Work Item Layout** option enables the user to define and save a global or specific layout for the Work items list.
- The **Web Part Pages** option enables the user to access the three default home page templates and personalize web part settings in them. Note that when you click Web Part Pages > Page 3, you will always be redirected to Page.

Header:

The header of all the pages in the Enterprise Console displays the name of the logged on user, the repository, and the user provider.

Note:

- Do not add multiple queues, calendars, or BAMs in the home page as this might not render the controls correctly. Do not add multiple controls in a single web part. As a best practice, do not use Inbox, Queue, Calendar, and BAM controls in a smaller web-part zone of the home page as these controls might not render as expected.

- When you access Enterprise Console on Microsoft Edge and Google Chrome (Chromium based browsers), it lists the below errors related to Feature Policy:

```
Error with Feature-Policy header: Unrecognized feature: 'vibrate'.
Error with Feature-Policy header: Unrecognized feature: 'usermedia'.
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Workflow Management

Workflows are responsible for driving end-to-end business applications. Manage workflows of a AVEVA Work Tasks Repository with Workflow Lists.

What is a Workflow List

The Workflow List is a list of workflows and folders which contains workflows within it. The Workflow List provides a central interface that enables categorization of workflows.

Actions you can be perform for a Workflow list are:

- Create a folder
- Create and view workflow attributes
- Create new FileWatcher, List and BAM workflows
- Publish a workflow
- Associate a workflow with BAM, FileWatcher, and Enterprise Console Lists
- Organize workflow items by grouping them into folders
- Manage the version changes to a workflow
- Disable or delete a workflow
- Manage security settings

Group Workflow Items within a Folder

Folders enable you to categorize items in the Workflow List. For example, you can create folders and group the workflow items into separate folders using the Organize option.

To group Workflow Items with Folders

1. In the **Enterprise Console**, click menu icon, and then click **Workflows**. The Workflows page appears.
2. On the action bar, click **New**, and then click **Folder**. The **New folder** screen appears.
3. In the Title box, type a name for the folder, and in the Description box, type a description for the folder.
4. Click **Save & Close**. The folder is created.
5. Click **OK**. The new folder is added to the folders list.
6. Click **Organize** on the ribbon bar. The Organize Workflows window is displayed.

   Note: The Organize Workflows window arranges all the folders and workflows in a tree format. You can drag and drop the workflows into the required folders.
7. Drag and drop the required workflows into the folder and click **Close**.

   Note: You can also right-click the item (workflow or folder) in the tree menu and move it to the desired location using the Cut and Paste method.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create a Folder

To create a folder

1. On the action bar, click **New**, and then click **Folder**. The **New folder** screen appears.

   In the Title box, type a name for the folder, and in the Description box, type a description for the folder. Folder names cannot have special characters such as, **:"\\""'<>;|*/{}?**.
2. Click **Save & Close**. The folder is created.
3. Click **OK**. The new folder is added to the folders list.

Publish a Folder

You can publish the folder and all the workflows within that folder will be published.

To publish a folder

1. Select and right-click the folder to be published.
2. Click **Publish**. Alternatively, you can select the folder, and then click **Publish** in the ribbon.
   The selected folder and all workflows within the folder will be published.

Delete a Folder

You can delete a folder which is not published.

To delete a folder

1. Select and right-click the folder that you want to delete.
2. Click **Delete**. Alternatively, you can select the folder, and then click **Delete** in the ribbon. The selected folder will be deleted. A folder cannot be deleted if it contains workflows that are associated with Enterprise Console lists.

Enable or Disable a Folder

To disable a folder

1. Select and right-click the folder that you want to disable.
2. Click **Disable**. The selected folder will be disabled.
   The disabled folder appears grayed out in the list and cannot be accessed.

To enable a folder

1. Select and right-click the folder that you want to enable.
2. Click **Enable**. The selected folder will be enable.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Change the Owner of the Folder

To change the owner of a folder

1. Select and right-click the folder for which you want to change the owner.
2. Click **Change Owner**. Alternatively, you can select the folder, and then click **Change Owner** in the ribbon. The **Change List Item Owner** page appears.
3. Select the user from the **User Lookup** window.
4. Click **Save**. The user information is saved successfully.

Apply Security Settings to a Folder

To set the security settings of a folder

1. Select and right-click the folder for which you want to set the security settings.
2. Click **Security Settings**. Alternatively, you can select the folder, and then click **Security Settings** in the ribbon. The **Security Settings** page appears.
3. Set the security settings.

Navigate to Different Folders in Workflows List

You can navigate back to the parent folder by clicking the Navigate icon available in the ribbon.

Create a Workflow

To create a Workflow

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**.

   The **Workflows** page appears.
2. On the action bar, click **New**, and then click **Workflow**.

   The **New Workflow** dialog box appears.
3. In the **Title** box, type a name for the workflow, and in the **Description** box, type a description for the workflow.
4. Click **Save & Continue**.

   The workflow is created, and the **Workflow Options** wizard appears.
5. Select **Design the Workflow** now, which is the default option, and then click **Finish**.

Workflow Options

The following table shows the available workflow options:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Task	Action
Design the Workflow now	Select this option to open the Process Designer for the Workflow. For more details about the Process Designer, see About the Process Designer.
No task to be performed now.	Select this option to perform workflow actions later. Selecting this option takes you back to the Workflow list. You can return to the **Workflow Options** window by selecting Tasks from the shortcut menu of the workflow.

Create an Association for a Workflow

1. To create associations for a workflow, right-click on the workflow, and then select **Create Associations** from the context menu.
2. After you click **Create Associations**, the page appears to associate an event with workflow:

Associate Workflow with Enterprise Console Events

To associate the workflow with a new Enterprise Console list event

1. Select the workflow, and then click **Associate** from the ribbon.

   Alternatively, select and right-click the workflow, and then select **Create Associations**.
2. Click **Enterprise Console**.
3. Enter the following details:
4. **Type**: Select either **List** or **List** item.
   a. If **Type** is selected as List, the following needs to be done:
      a. **Event Details**: Select the event for which the workflow has to be associated.
      b. **List Name**: Enter the list name or select it from the List Explorer.
      c. **List Item Name**: Enter the list item name.
      d. **Effective From**: Select the date from which the association is effective.
      e. **Effective To**: Select the date till which the association is effective. Only the events occurring between this period will trigger the workflow.
      f. **Workflow Version**: Select the workflow version from the list
   b. If **Type** is selected as **List Item**, the following needs to be done:
      a. **Event Details**: Select the event for which the workflow has to be associated.
      b. **List Item Name**: Enter the list item name and click **List Explorer** icon, to access the List Explorer wherein you can select the required list: Select a List, for example Customer List i.e. the list to which you want to associate the workflow. Select the type of customer. The list item lookup will display the field which is set as the **Default Field** and the field and which has the options **Display in Grid** and **Display in Lookup** set as True.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

c. **Effective From**: Select the date from which the association is effective.

d. **Effective To**: Select the date till which the association is effective. Only the events occurring between this period will trigger the workflow.

e. **Workflow Version**: Select the workflow version from the list.

5. Click **Attach Event**, to associate the workflow with the new Enterprise Console list event.

   The confirmation message **Association saved successfully** appears.

BAM Events Associations

You use this feature to trigger a workflow using BAM Events.

To associate a workflow with a new BAM event, follow the steps given below

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**.

   The **Workflows** page appears.
2. Select an workflow.
3. On the action bar, click **Associate**.

   The **New Association** wizard appears.

   Alternatively, select and right-click the workflow to be associated. Click **Create Associations**.
4. On the action bar, click **BAM** to associate the BAM events.
5. In the **Workflow** drop-down list, select the workflow from which the BAM Events of Overdue and KPI are to be raised.
6. In the **Based** on drop-down list, select either **Overdue Analysis** or **KPI Analysis**.
   a. If you select **Overdue Analysis**, the **Select Overdue Slot** button appears. This is used to select the Overdue slot and the event associated with it.
   b. If you select **KPI Analysis**, then the **Select Slot** button appears. This is used to select the KPI name, attached KPI data points and the event associated with it.
7. Select the date range for **Event** execution using the **Effective From** and **Effective To** options.
8. Click **Attach Event** to trigger the Workflow when the BAM Report is being processed.

Associate Workflow with ArchestrA Events

Use the **Associate** option to associate a workflow with ArchestrA Events.

The ArchestrA Event association window appears.

To associate a workflow with an ArchestrA Event for a specific event

1. Select the workflow, and then click **Associate** from the ribbon.

   Alternatively, select and right-click the workflow, and then select **Create Associations**.
2. Click **ArchestrA Event** to associate the workflow with the ArchestrA Event list for a specific event.
3. Enter the following details:
   - **Event Provider**: Select the name of the event provider from which you want to poll for new events.

     The Event Provider field is mandatory, and the maximum length of the event provider name is 250

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

characters.

Based on the data available in the ArchestrA Event main list, the Event Provider list is populated. You can add other event providers to this main list, if required.

Example of Event Providers are InBatch Server or Avantis.

- **Event Type**: Select the Event Type of the Event Provider.

  It is the type of event which defines the properties the event will have.

  For InBatch Server, Batch is an event, and possible values for Batch events are such as, Batch, Equipment, UnitProcedure, Operation, Phase, MaterialInput, MaterialOutput, Custom, and BatchError.

  Based on the data available in the ArchestrA Event main list, the Event Type list is populated for the selected Event Provider. You can add other event types for the event provider, if required.
- **Source**: Type the Source property name of the event. It is the name of the event provider sending the event. For example, for a *Batch* event, *BatchMngr* is the source.

  Select the appropriate operator from the operator drop-down list.
- **Area**: Type the Area of the event. Areas provide a key organizational role in grouping alarm information and assigning it to users who use alarm and event clients to monitor their Areas of responsibility.

  The Area object plays a key role in alarm and event distribution. All objects belong to an Area. Areas can contain sub-Areas. Alarm and event clients are configured to subscribe to a set of Areas.
- **Priority**: Type the priority for each designated event.

  Priority, which is an integer value from 1 to 999 indicating the severity of the alarm which raises an event. An alarm priority of 1 is most urgent and 999 least urgent.
- **Workflow Version**: Select the version of the workflow to be associated with the ArchestrA Event. The workflow will be initiated only if the published version at the time of the occurrence of the event is the same as the version selected here. To trigger the currently published version of the workflow whenever the event occurs, select *Latest*. When a version is specified, the workflow alerts the latest published version of that version series. For example, if the specified version is 1.1 and the latest published minor version for 1.1 is 1.1.1.3, then 1.1.1.3 is alerted.
- **Properties**: Based on the **Event Type** that you enter, select values in the Properties section. For example, for a Batch event type, BatchServer is a property of the data type String. The Properties field is optional.

  Based on the data available in the ArchestrA Event main list, the properties of the Event Type list is populated. You can add other properties of the event type, if required.

4. Click **Attach Event** to associate the workflow with ArchestrA Event for that specific event.

   Click **Cancel** to abort.

   The confirmation message, *Successfully associated the workflow* appears.

Limitations

- The *Between* operator is not supported for **Priority** and **Properties**.
- The data type of the operators are not validated. Ensure to provide valid values to the property data type.
- Only one workflow with one event provider is supported. That is, different event providers for a workflow with different association is not supported.
- While creating a filter condition, duplicate properties are not supported.
- The *OR* condition is not supported while creating a filter condition.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Associate Workflow with Message Queue

Use the **Associate** option to associate a workflow with Message Queue.

To associate a workflow with Message Queue for a specific connection

1. Select the workflow, and then click **Associate** from the ribbon.
   Alternatively, select and right-click the workflow, and then select **Create Associations**.
2. Click **Message Queue** to associate the workflow with the Message Queue.
3. Select the name of the Message Queue connection from the **Connection** list.
   The following fields appear based on the **Binding Type** selected for the **Connection**:

Binding Type for Connection	Field Name	Description
WM Object	Message Type	Enter the message type that needs to be listened from Message Queue.
XML	XPath	Enter the XPath for the XML Message. For example, `/CATALOG/PLANT[1]/PRICE/text()`
	XPath Value	Enter the XPath Value for the XML Message. For example, `$2.44`
	XML Schema	Enter the XML Schema for the XML Message. For example, see XML Schema Example.
JSON	JSON Path	Enter the JSON Path for the JSON Message. For example, `topping[3].type`
	JSON Path Value	Enter the JSON Path Value for the JSON Message. For example, `Powdered Sugar`
	JSON Sample	Enter the sample for the JSON Message. For example, see JSON Sample Example.

4. Click **Save** to associate the workflow with Message Queue for that specific connection.

The Workflow is triggered based on only message type, irrespective of the queue or connection to which it is associated. Hence, the message type must be unique through out the repositories configured.

Example:

1. Create Message Queue Connection in **machine1** with queue name as **Q1** and **machine2** with queue name

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Q2.

2. Associate the workflow to the connection created in the above step and message type **M1**.
3. Publish the message in Message Queue queue **Q2** with message type **M1**.

   The Workflow will be triggered, even if it is not from the queue or machine configured in the Message Queue Connection.

Note:
- JSON of array type without a root node is not supported for Associating Workflow with Message Queue. To support this, the JSON should be embedded inside a root node.

For example:

The following JSON of array type without a root note is not supported:

```
[{
"Data1": false,
"Data2": 12,
"Data3": 54657.66
}, {
" Data1": true,
" Data2": 45,
" Data3": 4576
}]
```

```
The above JSON sample can be modified as below to be supported:
{
"DataParent":
[{
"Data1": false,
"Data2": 12,
"Data3": 54657.66
}, {
" Data1": true,
" Data2": 45,
" Data3": 4576
}]
}
```

- The associated Workflow may get executed even though the JSON or XML schema of the message received by the queue does not match the schema specified in the association. In such scenario, the Xml Variable may not be in the correct format.

- RMQEventData is the XML variable that gets created on associating to Message Queue Association.

- Associating multiple versions of same workflow to different Message Queue associations is not recommended.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

XML Schema Example

```
XPath: /CATALOG/PLANT[1]/PRICE/text()
XPath value: $2.44
Sample XML:
<CATALOG>
<PLANT>
<COMMON>Sphagnum Moss</COMMON>
<BOTANICAL>Sanguinaria canadensis</BOTANICAL>
<ZONE>4</ZONE>
<LIGHT>Mostly Shady</LIGHT>
<PRICE>$2.44</PRICE>
<AVAILABILITY>031599</AVAILABILITY>
</PLANT>
<PLANT>
<COMMON>Columbine</COMMON>
<BOTANICAL>Aquilegia canadensis</BOTANICAL>
<ZONE>3</ZONE>
<LIGHT>Mostly Shady</LIGHT>
<PRICE>$9.37</PRICE>
<AVAILABILITY>030699</AVAILABILITY>
</PLANT>
</CATALOG>
```

Enter below XML Schema in the XML Schema field to consume the above XML Sample.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema"> <xs:element name="CATALOG"> <xs:complexType>
<xs:sequence> <xs:element name="PLANT" maxOccurs="unbounded" minOccurs="0">
<xs:complexType> <xs:sequence> <xs:element type="xs:string" name="COMMON"/> <xs:element
type="xs:string" name="BOTANICAL"/> <xs:element type="xs:byte" name="ZONE"/> <xs:element
type="xs:string" name="LIGHT"/> <xs:element type="xs:string" name="PRICE"/> <xs:element
type="xs:short" name="AVAILABILITY"/> </xs:sequence> </xs:complexType> </xs:element>
</xs:sequence> </xs:complexType> </xs:element> </xs:schema>
```


JSON Sample Example

```
JSON Path: topping[3].type
JSON Path value: Powdered Sugar
JSON Sample :
{
"id": "0001",
"type": "donut",
"name": "Caramel Donut",
"ppu": 0.55,
"batters":
{
"batter":
[
{ "id": "1001", "type": "Regular" },
{ "id": "1002", "type": "Chocolate" },
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{ "id": "1003", "type": "Blueberry" },
{ "id": "1004", "type": "Devil's Food" }
]
},
"topping":
[
{ "id": "5001", "type": "None" },
{ "id": "5002", "type": "Glazed" },
{ "id": "5005", "type": "Sugar" },
{ "id": "5007", "type": "Powdered Sugar" },
{ "id": "5006", "type": "Chocolate with Sprinkles" },
{ "id": "5003", "type": "Chocolate" },
{ "id": "5004", "type": "Maple" }
]
}
```

Associate Workflow with Logic App Events

Use the **Associate** option to associate a workflow with a Logic App event.

To associate a workflow with a Logic App event

1. Select the workflow, and then click **Associate** from the ribbon.

   Alternatively, select and right-click the workflow, and then select **Create Associations**.
2. Click **Logic App Event** to associate the workflow with a Logic App event.
3. Enter the following details:
   - **Enterprise Console**: Select an Enterprise Console site. The list shows Enterprise Console sites to which repositories are mapped, and for which Azure Service Bus (with Shared Access Signatures) are configured.
   - **URI**: The URI is automatically generated based on the selected Enterprise Console.
   - **Authorization Token**: The authorization token is automatically generated based on the selected Enterprise Console.
   - **Logic App Alert Key**: The default value is the name of the workflow being associated. You can enter any other value, if required.
   - **User**: The default value is the username of the logged-on user. You can select any other user, if required.
   - **Workflow Version**: The default value is the latest version of the workflow. You can select any other version, if required.
   - **Logic App JSON Data**: Enter the JSON schema to be passed from the Logic App.
   - **Logic APP XML Schema**: After entering the JSON schema for **Logic App JSON Data**, click **Get XML Schema** to auto generate the XML schema.
4. Click **Save** to associate the workflow with the Logic App event.

   The association is created, the **LogicAppAssociationData** XML variable is created for the schema, and an information message appears.
5. Click **Close**.

   The information message is closed.

Note: While posting alert from the Logic app, ensure the following are part of the HTTP request:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **URI** as generated based on the selected Enterprise Console.
- **Authorization Token**, as generated based on the selected Enterprise Console, is in the header.
- **ApplicationName** and **LogicAppAlertKey** in JSON schema are in the body.

View all Associations of a Workflow

To view all the associations of a workflow

1. Select and right-click the workflow for which you want to view associations.
2. Click **View Associations**.
   The associations of the workflow appears.
3. Right-click an association to **View Properties**, **Delete**, **Disable**, or define **Security Settings** of the association.
   If there are no associations for a workflow, the following message appears:
   There are No Events associated for the selected Workflow.

Delete Workflow Associations

To delete workflow associations

1. Select and right-click the workflow for which you want to delete associations.
2. Click **View Associations**.
   The associations of the workflow appears.
3. Right-click the association to be deleted, and then click **Delete**.
   The association is deleted.
4. Click **OK**.

Note: When you delete a workflow, the associations of that workflow gets deleted.

Disable Workflow Associations

To disable a workflow association

1. Select and right-click the workflow for which you want to disable associations.
2. Click **View Associations**.
   The associations of the workflow appears.
3. Right-click the association to be disabled, and then click **Disable**.
   The association is disabled.
   The disabled association appears grayed out and cannot be accessed.
4. Click **OK**.

Note: If a workflow is disabled, the event associated with it will not get triggered. You will be able to trigger the workflow only when the association is enabled.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create different versions of a Workflow

To create different versions of a workflow

1. Select and right-click the workflow for which you want to create different versions.
2. Click **Save As**, and then select the required available option. The available option are as follows:
   a. **Next Version:** Adds 1 to the lowest level of the last saved version of the workflow. If the last saved version is 1.1.1, then selecting this option saves the workflow as 1.1.2.
   b. **Next Major Version:** Next major version increments the workflow version by 1. If the last saved version is 1.1.1, selecting this option assigns 2 as the version number.
   c. **Next Minor Version:** Minor version for the last saved version. If the last saved version is 1, the next minor version would be 1.1. If the last saved version is 1.1, the next minor version would be 1.1.1.
   d. **Specific Version:** View a list of the next version numbers under the above three options.
3. In the **Save List Item** dialog box, enter the required comments, and then click **Save**.
4. A message appears to confirms that the item is saved successfully with the chosen option. Click **OK**.

When a workflow is first saved, the version number is set as 1.0 by default.

If all versions of the workflow are in **draft** status, then the version number in **Workflows** list is updated with the last saved version number.

If at least one version of the workflow is published, then the version number in the workflows list is updated with the published version of the workflow.

View All the Versions of a Workflow

To view all the versions of a workflow

1. Select and right-click the workflow for which you want to view different versions.
2. Click **Show All Versions**. All versions of the selected workflow appears.

The following workflow details appears:

- **Version:** Version of the workflow.
- **Comments:** Comments provided while saving the version.
- **Status:** Status of the version.
- **Is Archived:** Whether the workflow is archived or not.
- **Created On:** Date and time when the version was created.

Note: The **Publish and Test Run** option always executes the latest published version of the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Modify a Workflow Design

You can modify the design of the workflow.

Select the workflow and click **Design** from the ribbon. Alternatively, select and right-click the workflow, and then select **Design Workflow**. The **Process Designer for the Workflow** appears.

If you are designing a new workflow, the **Start** activity will appear in the **Process Designer** by default.

For more information about the Process Designer, see About the Process Designer.

Edit a Workflow

Use the **Edit** or **Edit Workflow** option to edit the description of an existing workflow.

To edit a workflow

1. Select the workflow you want to edit and click **Edit** from the ribbon. Alternatively, select and right-click the workflow, and then select **Edit Workflow**. The **Edit Workflow** window appears.
2. Change the **Description** of the workflow.
3. Click **Save & Continue** to save the changes. The workflow is updated, and the **Workflow Options** wizard appears.
4. Select an appropriate option, and then click **Finish**.

Note: Consider a scenario where an application level Work Item Field, which is set for a workflow, is not available in the repository. When you open or edit the workflow in the Design Mode, then the Work Item Field is recreated in the repository.

Delete a Workflow

You can delete a workflow which is in draft status.

To delete a workflow

1. Select the workflow you want to delete and click **Delete** from the ribbon. Alternatively, select and right-click the workflow, and then select Delete. A confirmation message to delete the workflow appears.
2. Click **OK**. The workflow is deleted.

Note: A workflow cannot be deleted if it has any child item. Only the user who has rights to delete will be able to delete a workflow.

On deleting the workflow, all the following information related to the workflow gets deleted:

- Work items
- Tasks
- History
- BAM and KPIs
    - BAM configurations

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

  - KPIs and KPI Data points
  - Performance Analysis
- Event Associations
  - Enterprise console
  - All the other associations
- Alternate Actor rules
- Work item layouts
- Work item fields
- Work items and Work item transactions
- Executions and Execution details

If a particular version of the workflow is deleted, only associations related to that specific version of the workflow are deleted. If the associations are not related to that particular version, then the associations are not deleted.

Example Scenario:

If a workflow has five versions and the third version gets deleted, then the information related only to the third version gets deleted. All the executions, work items, work item transactions, and execution history pertaining to the third version are deleted.

When the last version of a workflow is deleted, then all the associations and other related information of the workflow gets deleted. The following list shows the related information that gets deleted when deleting the last version of a workflow:

- Alternate Actor rules
- Work item layouts
- Work item fields (specific to this workflow)
- Event Associations
  - Enterprise console
  - All the other associations
- BAM and KPI
  - BAM configurations
  - KPIs and KPI Data points

When the last workflow of a repository gets deleted, then the information related to that workflow gets deleted. The following list shows the related information that gets deleted when deleting the last workflow in a repository:

- Alternate Actor rules
- Work Item Layouts
- Work Item Fields (defined at application level)
- BAM
  - Performance Analysis

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Publish a Workflow

Publish a workflow to deploy and activate the workflow.

To publish a workflow

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**.

   The **Workflows** page appears.
2. Right-click on a workflow, and then click **Publish** (or click on a workflow, and then on the **Action** bar, click **Publish**).

   The workflow is published and a message bar appears.
3. Click **Close**.

   The message bar is closed.

To publish multiple workflows

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**.

   The **Workflows** page appears.
2. Select multiple workflows, and then on the **Action** bar, click **Publish**.

   The workflows are published and a message bar appears.
3. Click **Close**.

   The message bar is closed.

Unpublish a Workflow

Unpublish a workflow to deactivate the workflow. When a workflow is unpublished, existing instances of the workflow are completed, and no new instances of the workflow are created.

To unpublish a workflow

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**.

   The **Workflows** page appears.
2. Right-click on a workflow, and then click **Unpublish** (or click on a workflow, and then on the **Action** bar, click **Unpublish**).

   The workflow is unpublished and a message bar appears.
3. Click **Close**.

   The message bar is closed.

To unpublish multiple workflows

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**.

   The **Workflows** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Select multiple workflows, and then on the **Action** bar, click **Unpublish**.

   The workflows are unpublished and a message bar appears.

3. Click **Close**.

   The message bar is closed.

Save Workflow as Quickflow

You can save the published workflow as Quickflow for faster execution. The **Save as Quickflow** option is available only for the published workflows.

Note: The workflows which have the activities and properties that are supported by Quickflow can only be saved as Quickflow.

To Save the Workflow as a Quickflow

- Select and right-click the workflow you want to save as Quickflow, and then select **Save as Quickflow**.

  The selected workflow gets saved as Quickflow.

Import a Workflow

You can import a workflow definition that has been saved as an XML file. You can import only unpublished workflows.

To import a workflow XML file

1. Select and right-click the workflow that you want to import from the workflow list.
2. Click **Import Workflow**. The **Import** dialog box appears.
3. Select the workflow definition which you want to save as an XML file.
4. Click **Import**. A message workflow successfully imported appears.
5. Click **Close**.

Export a Workflow

You can export a workflow definition as an XML file. This makes the workflow portable and accessible from other applications.

To export a workflow as an XML file

1. Select and right-click the workflow that you want to export from the workflow list.
2. Click **Export Workflow**. The **Export** dialog box appears.
3. Select the workflow definition which you want to export.
4. Click **Export**. Save the XML file.

Lock a Workflow for Editing

You can **Lock** or **Unlock** a Workflow, Process Designer, Forms Designer, and List designer while editing.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

When the designer or the list item is locked, other users are not allowed to edit or modify it. A lock icon will appear beside the list item if the designer is locked.

The following image shows the representations of different types of lock supported by AVEVA Work Tasks:

	Title	Description	Owner	Version
	Leave Details	Employee Leave Details	admin	1
	adminform1		admin	1
	adminWF		admin	1
	RS	RS	admin	1
	200941	200941	admin	1

Lock Legends

A locked list item can have one of the following **Lock** icon beside it:

: Item is manually locked by user who is currently logged in.

: Item it is locked by other users.

: Item is automatically locked by user who is currently logged in.

Functionality of Lock and Unlock

- You can select and right-click the list item to be locked and unlocked, and then select **Lock** or **Unlock** from the menu.
- Every list item opened for editing will get locked automatically, even though the item is not locked manually.
- Name of the user who has locked the list item can be viewed by placing the cursor on the **Lock** icon.
- Only the user who has locked the list item or the users from the administrative security group can **Unlock** the list item.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Enable and Disable Workflow

To enable a workflow

1. Right-click a disabled workflow that you want to enable, and then select **Enable**. A message about successful enable appears.
2. Click **OK**.

To disable a workflow

1. Right-click a workflow that you want to disable and select **Disable**. A message about successful disabling of the workflow appears.
2. Click **OK**.

All disabled workflows appear grayed out in the list and cannot be accessed.

Access BAM Configuration

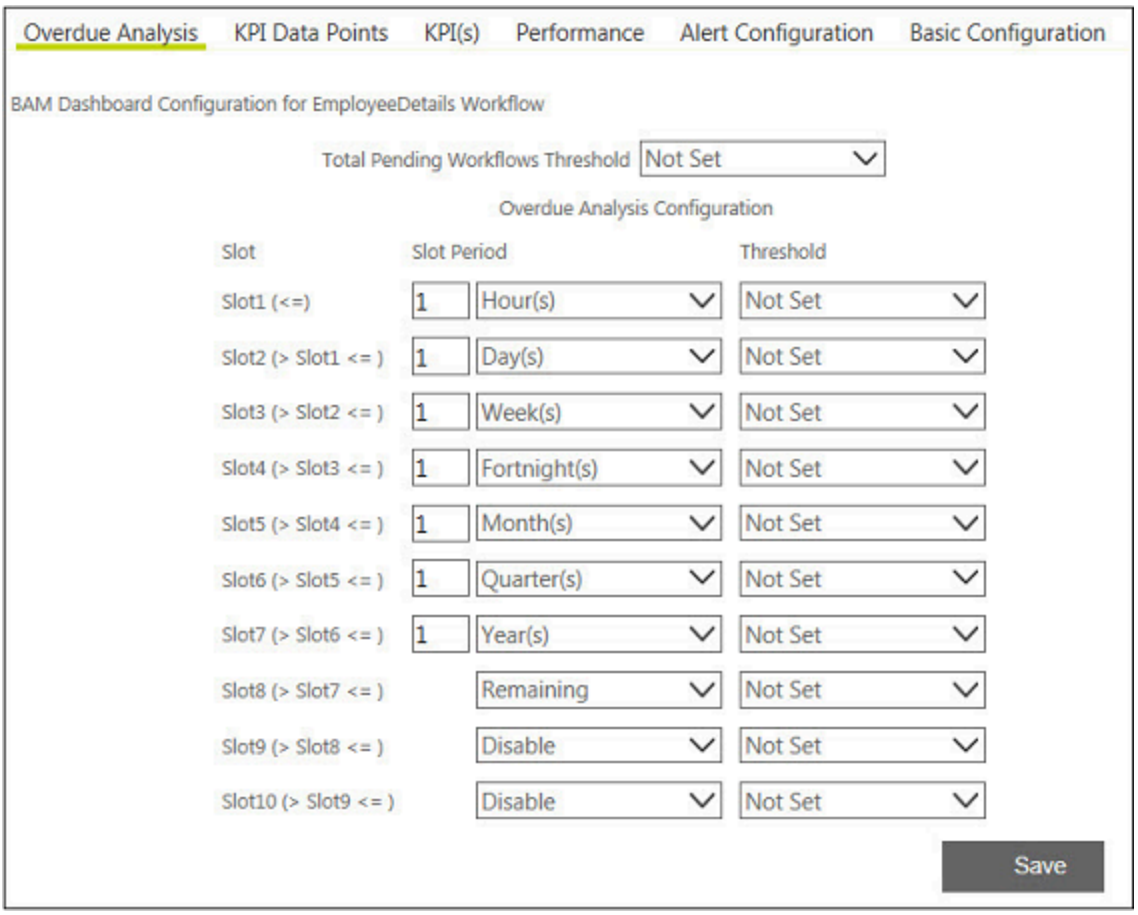
Use the BAM Configuration option to access the BAM Configuration of the workflow. The BAM Configuration option is available only if the logged in user has Process Designer level security rights.

To launch the BAM Configuration

1. Right-click a workflow, and then select **BAM Configuration**. The **BAM Configuration** window appears.
2. Select the required configurations, and then click **Save**. The modified configurations are saved.

   The following image shows the **BAM Configuration** window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Reference Topics:

Business Activity Monitoring Alert

BAM Configuration

Change Owner Information of a Workflow

You can change the owner of a workflow, if required. This feature is helpful if you want to to inherit workflows of a different user or you want to change the owner of a workflow. The **Change Owner** option is accessible only by the administrator.

To change the owner of a workflow

1. Right-click and select the workflow for which you want to change the owner information.
2. Click **Change Owner**. The **Changing Owner** window appears.
3. Select the new owner. Use the **User Lookup** window to select the user.
4. Click **Save**. A message confirming that the owner information is changed appears.
5. Click **OK**. The selected user will now be the owner of the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Access Workflow Tasks

To access Workflow Tasks

1. Right-click and select a workflow.
2. Click **Tasks**. The **Workflow Options** window appears.
3. Select the required **Workflow Tasks** option.

For more information about Workflow Tasks options, see Workflow Options.

Different Views Available in a Workflow List

This section shows the different views available in a Workflow List.

View a Workflow

You can view the design of the selected workflow in read-only mode by using the View Workflow.

To view a workflow

1. In the Enterprise Console, click the **Menu** button, and then click **Workflows**. The **Workflows** page appears.
2. Select and right-click a workflow, and then select **View Workflow** from the shortcut menu. The design of the selected workflow appears in read-only mode.

View the Execution of the Published Workflow

You can view a workflow when it is being published and test run. The workflow runs only if it is associated with any events.

To view the execution of the published workflow

- Select and right-click a workflow, and then click **Publish and Test Run**. The **Process execution view** appears.

View Child Items of a Workflow

Every workflow can have child items. The child items are either child workflows or folders.

To view the child items of a workflow

- Select and right click a workflow, and then click **Show ChildItem**. The child items of the selected workflow appears.

If the workflow has no child item, an empty list with the following message appears:

There are no rows in this view.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

View Audit History

You can view the history of a selected workflow to identify the action that was performed on the workflow and also the date and time when the action was performed.

To view Audit History

Select and right-click a workflow, and then click **View Audit History**. The **Audit History** window appears.

To view the audit history of a workflow, the logged-in user must have the **Can View Audit History** rights.

The Workflows list displays the latest version of the workflow in the grid. If all the versions are in Draft status, then the latest draft version of the workflow is listed or the latest published version of the workflow is listed in the grid.

Example:

Assume that the 'HR' workflow has up to 10 versions and all are in the 'Draft' state. The tenth version of 'HR' workflow gets displayed in the grid.

Consider that the 'Accounts Payable' workflow has the following versions 1, 1.1, and 1.2 and all are 'Published'. In such a case, the latest published version of the workflow that is, 1.2 is listed in the grid.

View the Properties of a Workflow

You can view the properties of a workflow using the **View Properties** option in the shortcut menu.

Column details in Workflow Properties window displays the following information:

- **Version:** Version of the workflow .
- **Status:** Status of the workflow
- **Is Archived:** States whether the workflow is archived or not

Child Items tab lists the child items of the selected workflow, details pertaining to the workflow which is added as a child item of the selected workflow is listed.

Grid details of Child Items tab displays the following information:

- **Title:** Title of the child workflow.
- **Description:** Description of the child workflow.
- **Version:** Latest version of the child workflow (if all the versions in draft status) or the status of the published version.
- **Status:** Status of the workflow.

View Exceptions

You can view all the exceptions and warnings of the corresponding workflow using the View Exceptions.

Select and right-click a workflow, and then click **View Exceptions**. The **Exception Details** window appears.

To enable exception persistence

1. To enable the persistence of exceptions and warnings in the database so that all these can be viewed using the “View Exceptions” option:

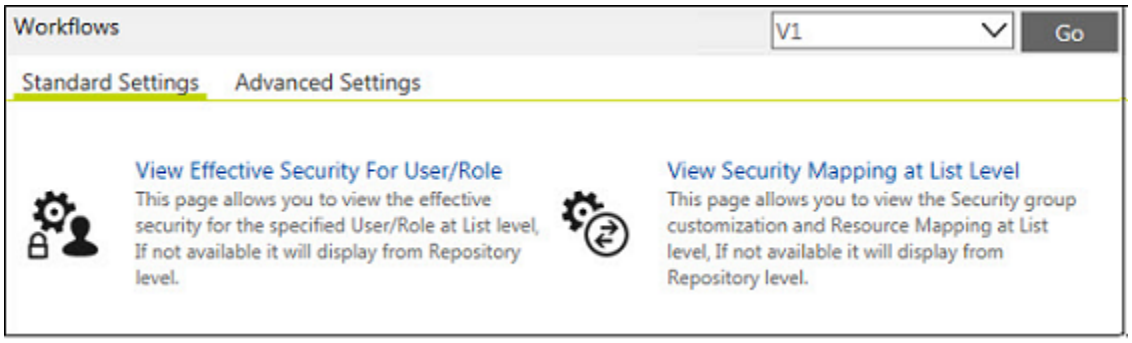
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Open the **Farm Configuration Wizard**.
3. Select **Service Settings**.
4. Click **Global Settings**, and then click **Log/Transaction** tab. The **Log/Transaction** tab window appears.
5. Select the appropriate values from drop-down to enable the persistence of the exceptions/warnings.

Set Security at Workflow Listitem Level

You define the security setting of a workflow at the listitem level using the Security Settings.

1. In the **Enterprise Console**, click menu icon, and then click **Workflows**. The **Workflows** page appears.
2. On the action bar, click **Settings**. The **Security Settings** page appears. The **Security Settings** page contains the **Standard Settings** and **Advanced Settings** tabs. You can configure the security settings for the selected workflow.



For detailed description about security settings, see Security Framework.

Calendars

AVEVA Work Tasks Calendar allows you to create multi-level and hierarchical calendars. You can define multiple child calendars that inherit from the parent calendars. For example, in an organization that spans across multiple geographical locations, you can define an organization-wide main calendar. You can define the work hours across the organization using the main calendar. You can create a separate child calendar for each work location to record the unique work hour requirements for that region. The child calendar inherits the settings of the main calendar and can be customized.

Additional levels of customized calendars, that inherit from the respective parent calendars, can be configured in this manner. The lowest level of calendars that can be defined by extending this process are called the Resource Calendars. See, About Resource Calendar.

For more information about configuring Resource Calendars, see Configure Resource Calendar by Administrator.

Working with Calendar Control

- Accessing the Calendar Control
- Adding a New Calendar
- Configuring the Calendar

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Configuring General Settings
- Configuring Day Settings
- Configuring Date Settings

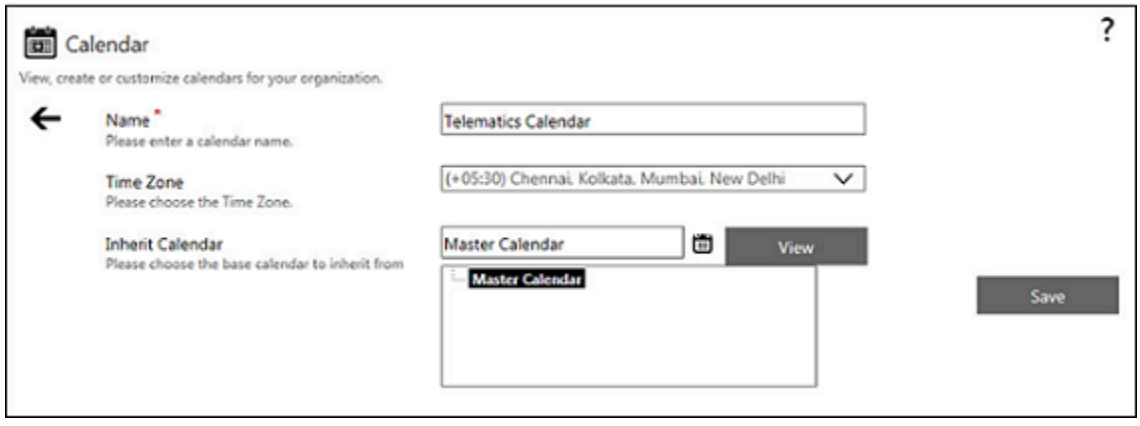
- Viewing the Parent Calendar

To access the Calendar Control

1. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Calendar**.
   The **Calendar Control** page appears.
2. On the **Calendar Control** page, you can view the list of all the available calendars.
   If no calendar has been configured, then only the **Add New Calendar** button is displayed.

To add a New Calendar

1. Click the **Add New Calendar** icon.
   The Calendar Control page is updated by the fields required to create a new calendar.
2. On the Calendar Control page, do the following:
   a. In the **Name** box, type a name for the new calendar.
   b. In the **Time Zone** list, select the time zone that you want to use for the new calendar.
   c. In the **Inherit Calendar** box, select the parent calendar from which you want the new calendar to inherit properties.
   d. Click the icon beside the **Inherit Calendar** box. A drop-down list appears with the names of all the parent calendars. Select one parent calendar from the list.
   e. In the **Provider** list, select the name of the **Calendar Provider** on which you want to create the new calendar.



Note: When a single provider is configured, then the Provider field is not available. Click save. The calendar Configuration section appears in the same page. Click View List to return to the calendar list view.

To configure the Calendar

After creating the new calendar you need to configure it with the required parameters. There are three grids in this section - **General**, **Days**, and **Date**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Click **Set Work Slot**. The **Set Work Slot** dialog box appears.
2. Update the work slot, and then click **Close**.

To configure General Settings

You can set the general work hours for the calendar in the General grid.

1. Click **Set Work Slot**. The **Set Work Slot** window appears.
2. In the **Set Work Slot** window, do the following:
   a. In the **Name** field, type a name for the work shift.
   b. In the **From** field, select the start time of the shift.
   c. In the **To** field, select the end time of the shift.

You can add more rows for setting shift details by pressing the <Tab> key or using the grid shortcut menu. After making the required work hour settings, click **Close**.

To configure Day Settings

You can set the work hours for specific days in the Day grid. For example, you can specify alternative work hours for Saturdays. You can also set Sundays or any other day of the week as the weekly day off.

1. Click **Set Work Slot**. The **Set Work Slot** window appears.
2. In the **Set Work Slot** window, do the following:
   a. In the **Day** list, select a day of the week. Click the **Get Details** button. The grid gets populated with existing shift details, if any. You can edit the existing shift details.
   b. In the **Consider as Holiday** option button, select **Yes** if you want to consider the selected day as holiday.
   c. In the **Name** column of the grid, type a name for the work shift.
   d. In the **From** column of the grid, specify a start time for the shift.
   e. In the **To** column of the grid, specify the end time of the shift.

You can add more rows for setting shift details by pressing the <Tab> key or using the grid shortcut menu. After making all the work hour settings, click **Close**.

To configure Date Settings

You can set the work hours for a specific date in the **Date** grid. For example, you can select a date and set it as a holiday.

1. Click the **Set Work Slot** grid. The **Set Work Slot** window appears.
2. In the **Set Work Slot** window, do the following:
   a. In the **Date** field, select a date. Click the **Get Details** button. The grid gets populated with existing shift details, if any. You can edit the existing shift details.
   b. In the **Consider as Holiday** option button, select **Yes** if you want to consider the selected day as holiday.
   c. In the **Name** column of the grid, type a name for the work shift.
   d. In the **From** column of the grid, specify a start time for the shift.
   e. In the **To** column of the grid, specify the end time of the shift.

You can add more rows for setting the shift details by pressing the <Tab> key or using the grid context menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

After configuring all the work hour settings, click **Close** to close the window.

Tip: These settings provides a great deal of flexibility in creating new calendars. If you inherit the settings from a parent calendar, you only need to change the parameters that need to be alerted. This saves the time and effort that would otherwise be required for setting all the parameters individually for a calendar.

To configure a Child Calendar

You can configure a child calendar that will inherit the settings of the specified parent calendar.

1. While creating the calendar, in the **Inherit Calendar** field, select the appropriate parent calendar. In this example, select **Main Calendar**.
2. Click **save**. Then configure the settings for this calendar in addition to the settings inherited from the **Main Calendar**.

To view the Parent Calendar

1. In the **Inherit Calendar** box, select the parent calendar.
2. Click the **View** button beside the **Inherit Calendar** box. The display of the view button is controlled by the **CanViewCalendar** property. The default value for this property is false. If it is set to **True**, the Calendar Control page appears as shown in the following image.

Inheriting from a Parent Calendar

You can create a new calendar by inheriting properties from an existing parent calendar. If it is not inherited from any parent calendar, the new calendar becomes the parent calendar. Taking the process of inheriting from a top-level calendar further, you can create a hierarchy of calendars.

For example, for an organization that operates in several countries you can have a separate calendar for each country. Under the parent calendar for a country, you can have a number of calendars for each of the region in the country. Under the calendar of each region, you can have a number of calendars for each department. Similarly, within each department calendar, you can have a resource calendar for each user in the department.

Sample Scenario

If a task is assigned to a user and there is no response after five working days (excluding holidays), a timeout reminder should be sent to the user. The following steps can be used to achieve this scenario:

1. Create a Calendar.
2. Add a General shift which covers 24 hours (Day Shift: 9 am to 6 pm; Night Shift: 6 pm to 9 am).
3. Set holidays as required.
4. Associate this Calendar to any human activity.

If you want to set timeout for a resource, you have to calculate the timeout based on the following:

- Shift timing of the resource
- Holidays/weekly off

The resource calendar can also be overridden based on the Calendar Property settings for the activity.

Consider the following example, where:

- Shift timing of the resource = 8 hours per day (working hours per shift)

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The resource works on all the days of a month (considering 5 working days in a week) = 20 days

Timeout value = (8 hours * 20 days) /24 hours = 6 days

Resource Calendar

Configuring a Resource Calendar

Configure a resource calendar by inheriting the settings of a specified top-level calendar. A resource calendar is the lowest level calendar that you can create and it cannot have a child-calendar.

See Configure Resource Calendar by Administrator for more details.

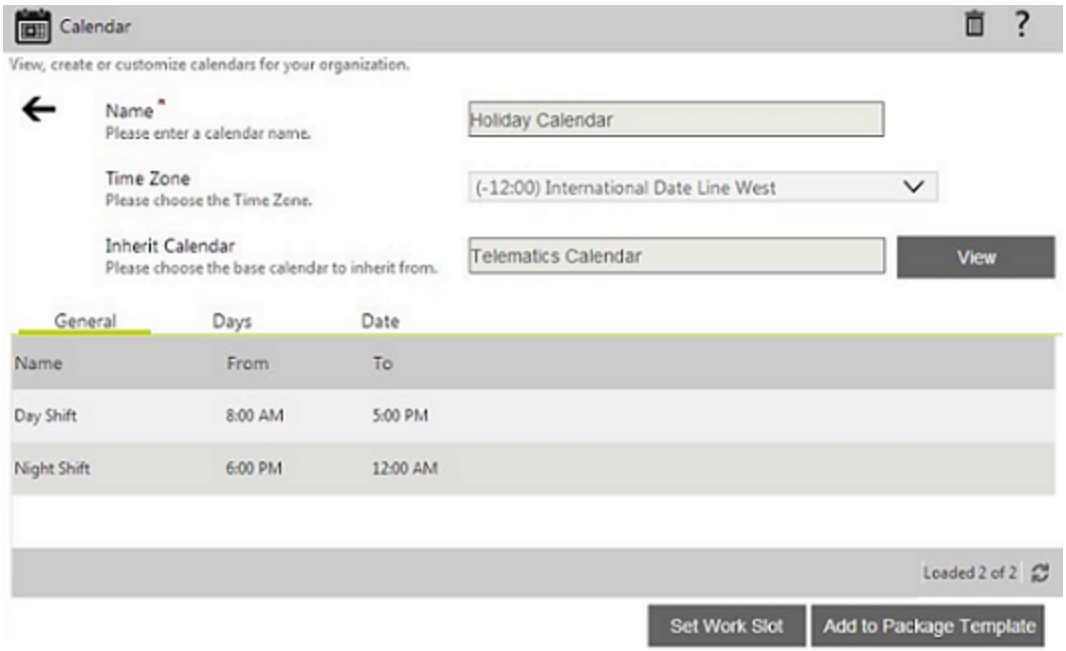
You must select the time zone for the resource in the **Time Zone** drop down. You also need to select the top-level calendar from which this resource calendar is to inherit properties in the Inherit **Calendar** field.

You can access the **Resource Calendar** feature from the Inbox.

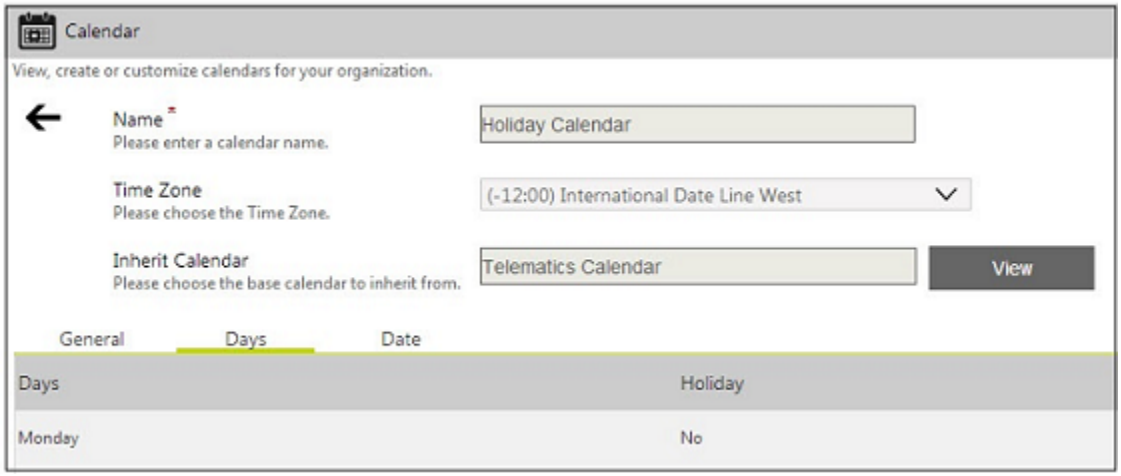
On clicking the icon, the Resource Calendar screen opens.

After configuring the settings, you can click the **Save** button to save the resource calendar. This displays the configuration window below the above one. You can configure the calendar using the same steps described for configuring the Calendar Control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



You can click on the **View List** button to return to the Calendar List view.



For more information about configuring Resource Calendars from the Work items list window. see Work Items List

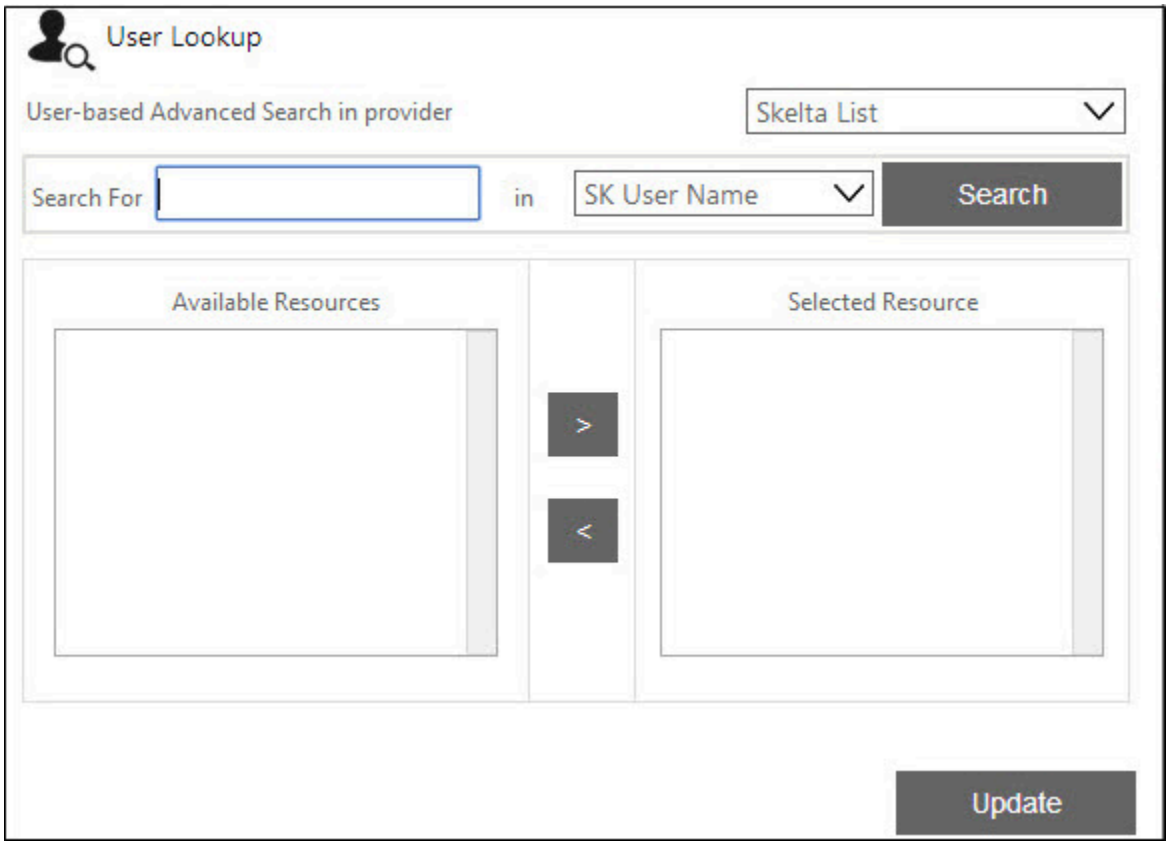
Configure Resource Calendar by Administrator

Users with administrative rights can configure the resource calendar for other users.

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **Calendar**.
   The **Calendar Control** page appears.
   One global calendar should be defined to set the Resource Calendar.
2. Click **Set Resource Calendar**.
3. Select the required provider from the Provider drop down.

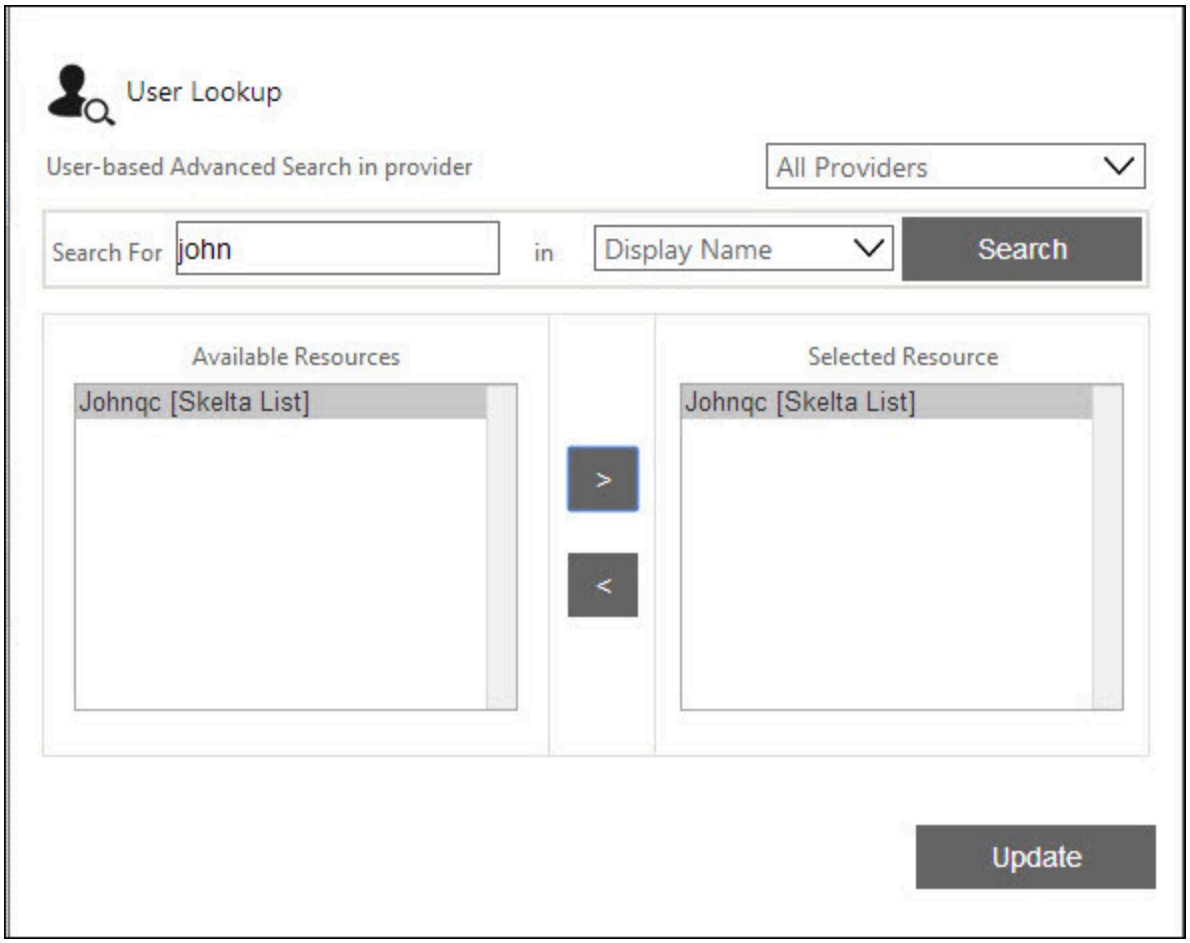
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Select the user for whom the Resource Calendar is to be set in the **User Lookup** pop-up.

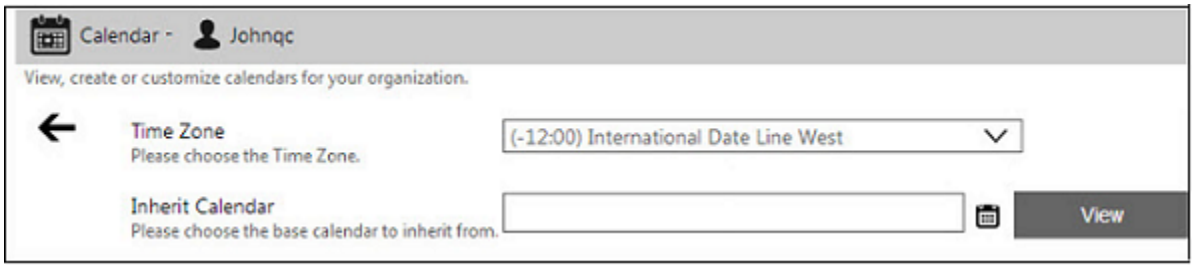


The following image shows user lookup with the selected resource:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

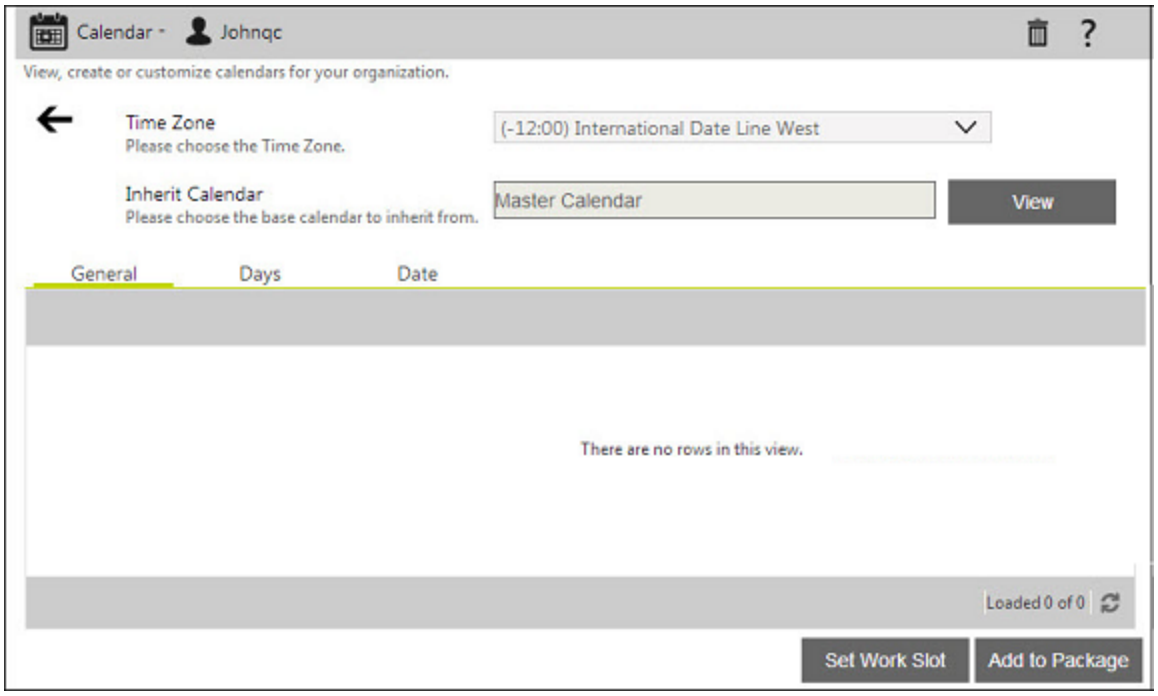


5. Calendar Control will be displayed for the selected user.

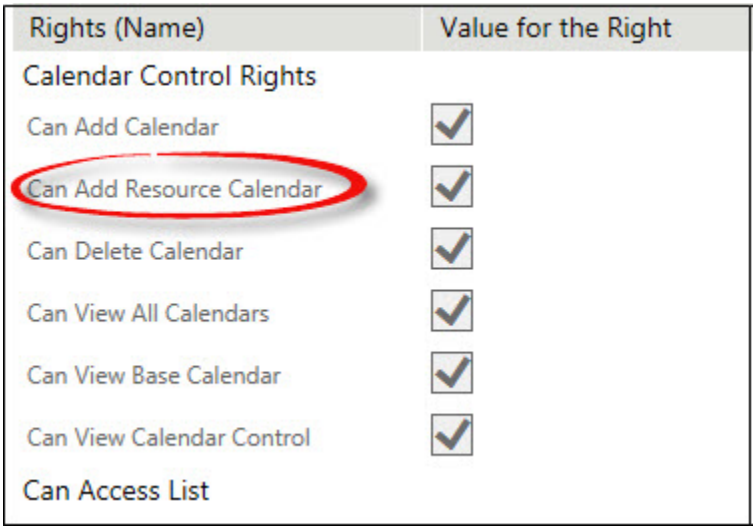


6. Save the calendar and then set the work slots.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Note: The right **Can Add Resource Calendar** is added to the **Calendar Control Rights** to set whether a user can configure Resource Calendars or not. By default, the administrator will have the **Can Add Resource Calendar** right checked.



Check Resource Availability

When the workflow process searches the calendar for work hour information, it begins at the lowest level calendar and works its way up to the top-level calendar.

- The lowest level is the resource calendar.
- If the required work hour information is not available in this calendar, the process looks in the calendar at the next higher level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- This could be the department calendar.
- If the information is not available even in this calendar, the search moves up to the next level, and so on.
- This process continues till the required information is found or till the top-level calendar is reached, whichever is earlier.

See *Finding the Resource Availability from the Resource Calendar* in the *API Scenarios Guide*.

Queue Management

You can use the Queue to assign tasks to groups of actors and to include queues comprising participants as part of their process. The queue feature is generally used by the **AVEVA Work Tasks** process administrators and owners. The queue feature provides a greater level of flexibility in the assignment of tasks to the participants.

Queue Features

Queue Management has the following features:

- Supports push-pull patterns
- Specify Role or User as Participants
- Access rights for Role or User
- Queue level escalations
- Allocate tasks to actors automatically or manually

You can perform the task allocation on the following basis:

- **Automatic:** The allocation is done automatically using defined algorithms. The assignment is done using the Round Robin, Random, or Least Used methods. This push pattern is the default setting.
- **Manual:** The task allocation is done manually. In this case, you can view and pick a task from the queue. Alternatively, the users can be allowed to use a button in the Work items list to get the next available task in the queue. In this case, the users cannot view the tasks in the queue but can click the button to get the next available task in the queue. This is the "pull" pattern as opposed to the "push" pattern where the queue assigns tasks.
- **Combination:** The task allocation is done using a combination of automatic and manual methods.

Significance of Queues

You can use Queues in workflows to set the recipient(s) of activities in the following manner:

- Assign tasks to resource(s) directly using the Assign Actor(s) property
- Assign tasks to Queues using the Assign Queue(s) property
- Assign tasks using both properties.

  Note: If both the properties are set, both the resource(s) and the queue(s) are assigned the task directly. The resource getting the task from the queue will have to act on it. In each case, the respective escalation or timeout setting will apply.

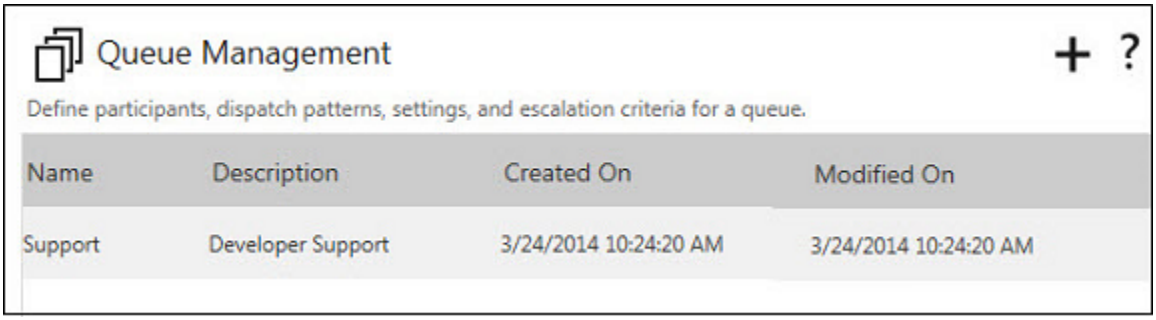
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Configure Queue Management

You use the **Queue Management** window to define the queue.

The Queue Control displays the following image when launched. It lists all the queues that have already been added in a grid.

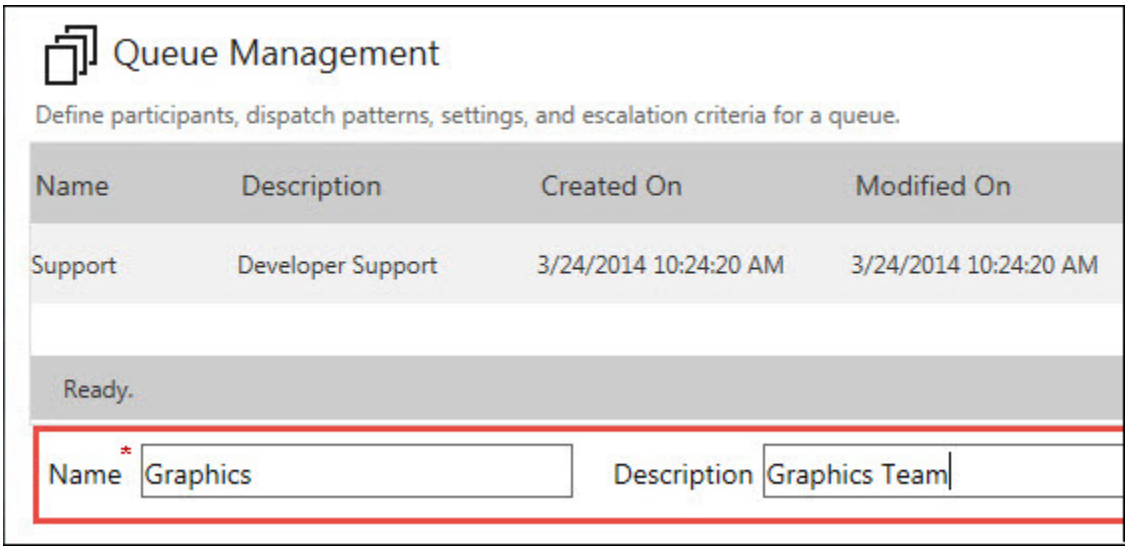
The following image shows the Queue Management window:



You can click the **New Queue** button to add a new queue. When you click the **New Queue** button in the previous window to add a new queue, it displays the **Name** and **Description** fields below the queue list, to enter a name and description for the new queue. You can enter the name and description and click on the Add button. This will add the new list and open the settings view of the Queue Management window.

Note: Ensure that the **Queue Name** that you enter does not exceed 250 characters, and the **Queue Description** does not exceed 150 characters.

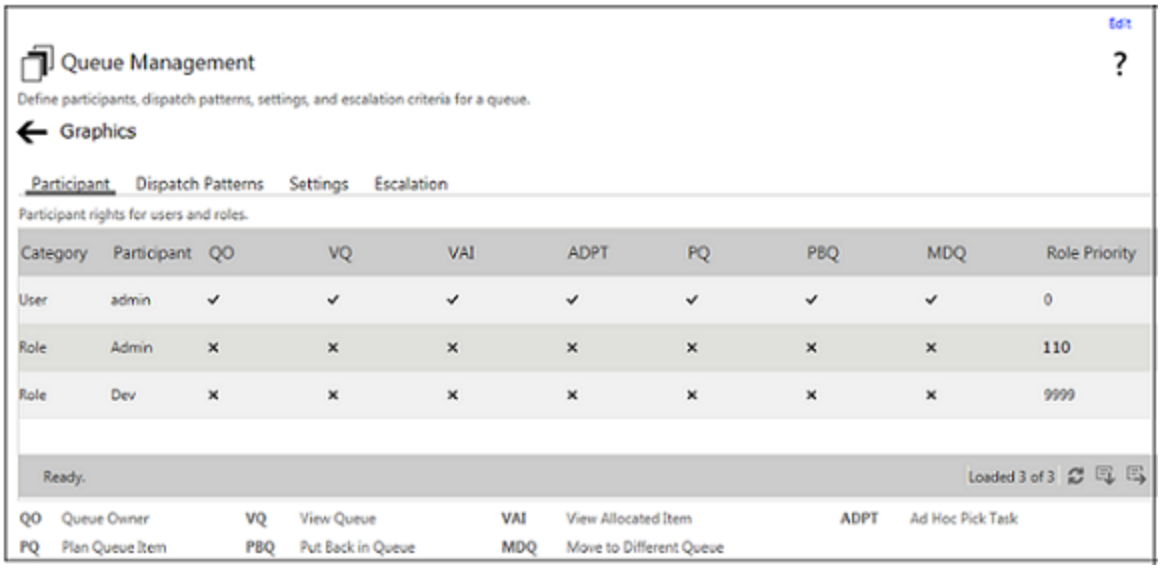
The following image shows Queue Management window with fields for adding new queue:



Queue Management Configuration View:

The following image shows the Configuration view of the Queue Management window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



You see this window when you click **Add** after specifying the name and description for a new queue. You can configure the different queue settings here for the new queue. This window has the following tabs, which opens the corresponding view.

- Participants Tab: Configure the settings related to the participants in the queue. This is the default tab.
- Dispatch Patterns Tab: Configure the settings for the dispatch of tasks from the queue.
- Settings Tab: Configure the queue settings.
- Escalation Tab: Configure the escalation settings for the queue.

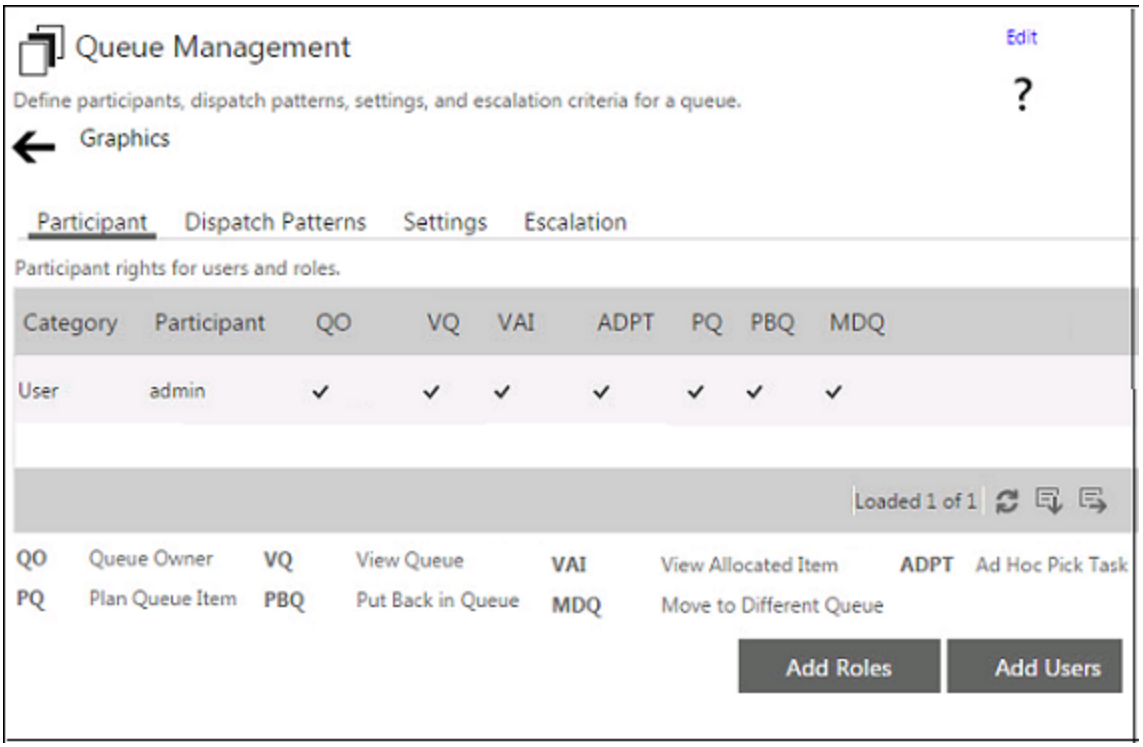
After you configure all the settings, click the **View List** button to return to the main Queue Management window. The new queue is added to the **Queue List**. Right-click on the row corresponding to the new queue, and select the **Edit** option from the context menu to open the configuration view to edit its settings, if required. Alternatively, double-click on the queue row to open this view.

Participants Tab

Configure the different settings for a new queue or edit the settings for an existing queue in this view of the Queue Management window. You can access these settings using the different tabs in the window. The first tab, **Participants**, is selected by default.

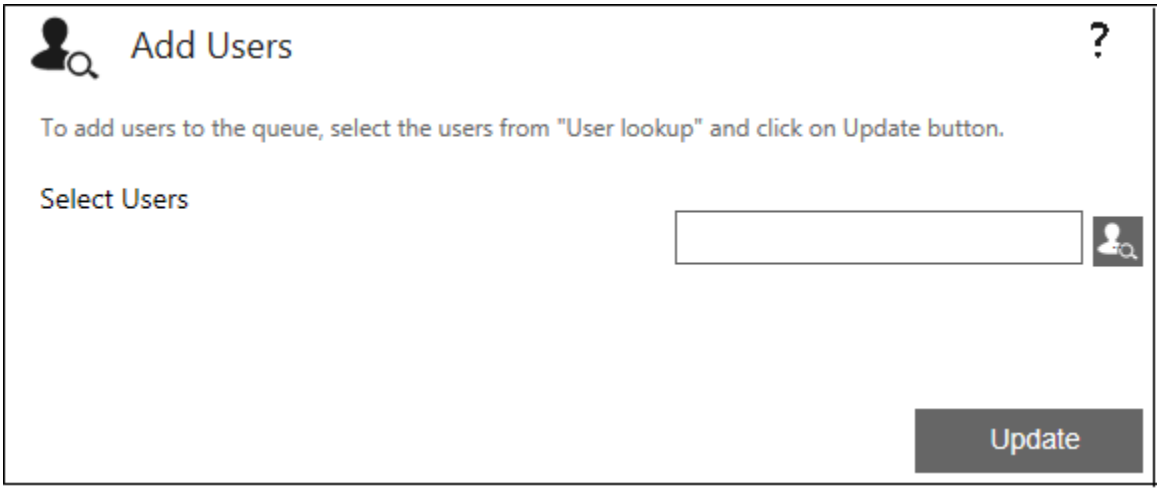
The **Participants** tab displays a list of all the existing participants along with their access rights. The following image shows **Participants** tab of the **Queue Management** window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

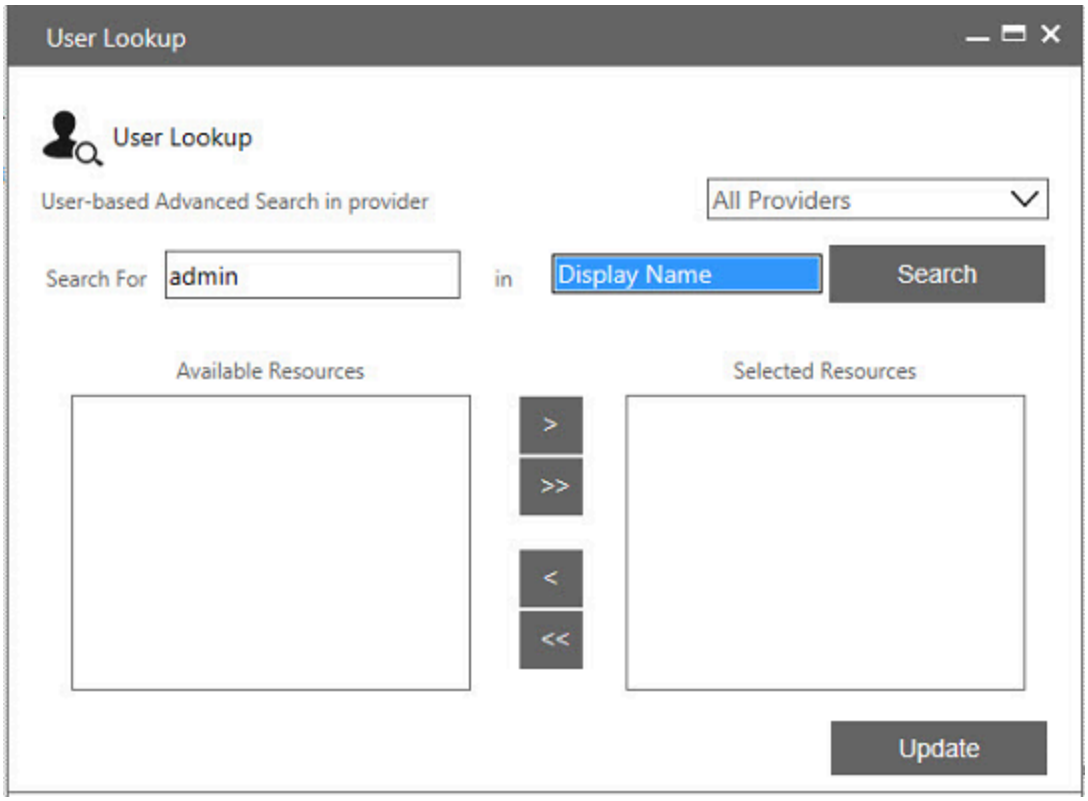


Adding Participants

Click **Add Users** to add a new participant to the queue. The **Add Users** window opens where you can add new users



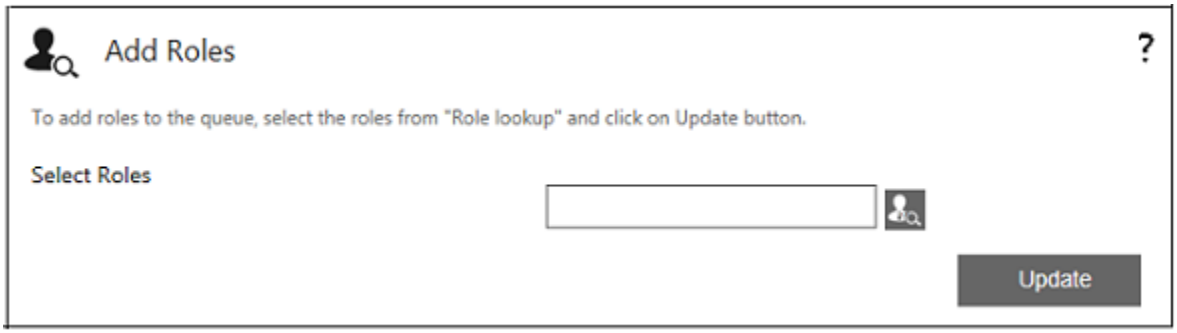
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



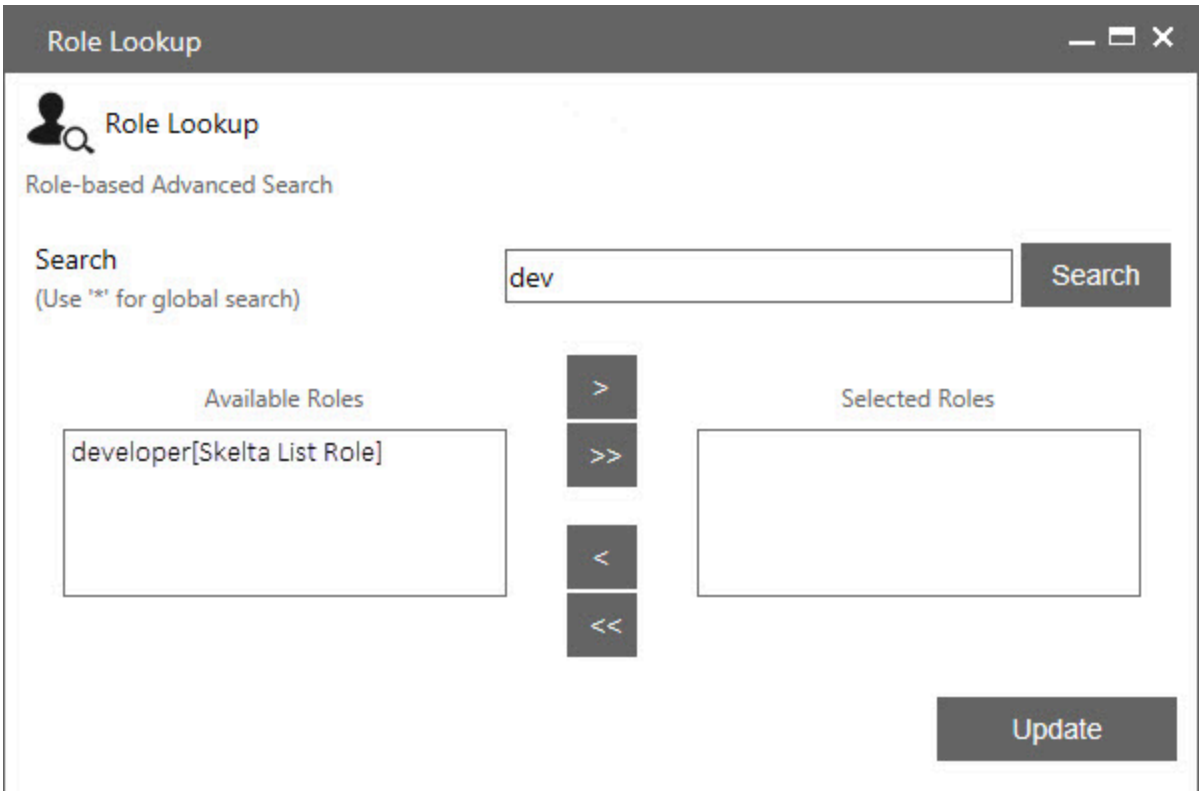
You can specify a search string to **Search for** resources from the desired provider. The resources matching the search parameter are extracted by the resource handler and displayed in the **Available Resources** list. You can select the resource names from this list and click the [>] button to move it into the **Selected Resources** list. You can click the [>>] button to move all the available resources into the selected resources list. After selecting the resources click **Update** to save the selection.

Adding Roles

Click the **Add Roles** button to add roles to the queue. You can add the new roles in the Add Roles window that is opened. The available roles extracted by the role handler are displayed in the **Available Roles** list. You can select the role names from this list and click the [>] button to move it into the **Selected Roles** list. You can click the [>>] button to move all the available roles into the selected roles list. After selecting the roles, click **Update** to save the selection.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

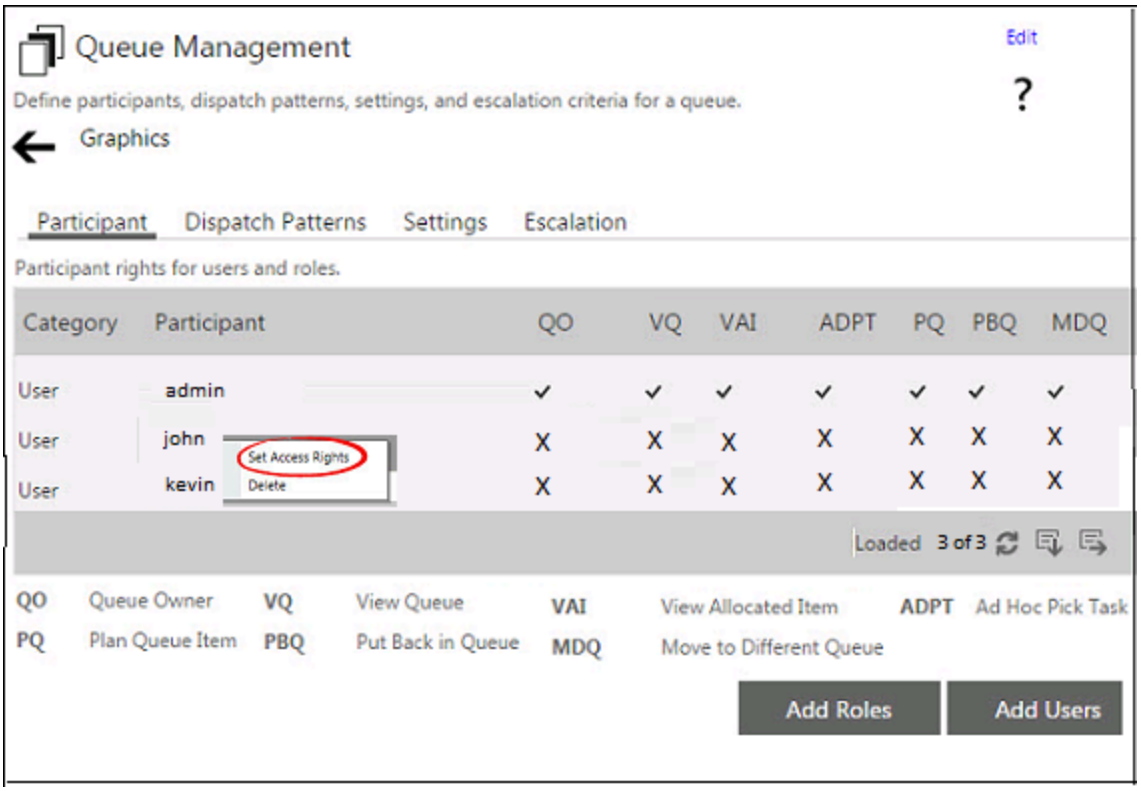


Note: The providers drop down will be available only if the repository has multiple providers.

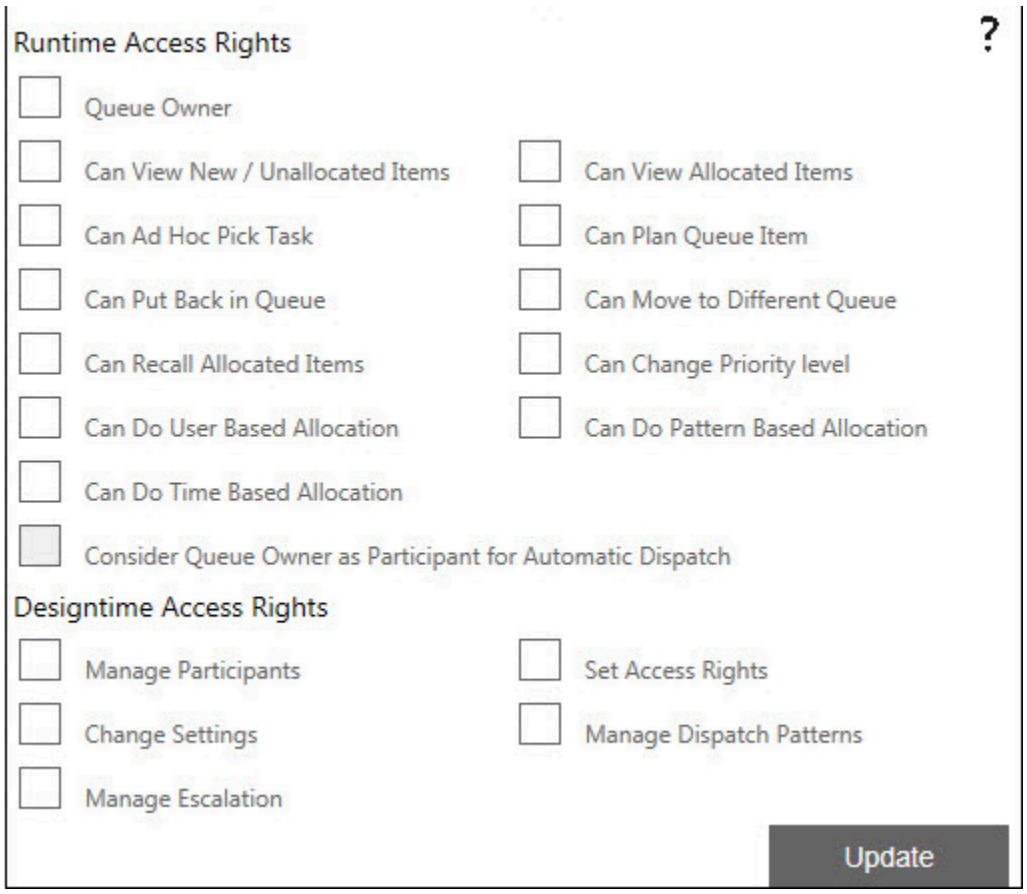
Setting Access Rights

You will need to set access rights for the various participants of a queue so that each person can perform only the permitted actions in the queue. You can right-click on any participant name or role name in the main Participants tab window, and then select **Set Access Rights** to open the Access Rights window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



After you select **Set Access Rights**, the **Runtime Access Rights** dialog box appears:



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

You can use the **Runtime Access Rights** window to set the access rights for the selected participants. You can set both Runtime and Design time Access Rights here.

Runtime Access Rights: The run-time access rights that can be set in this section allow administrators to set the access rights of participants during the execution of the queue.

- **Queue Owner** - You can set the participant as an owner of the queue.
- **Can View New/Unallocated Items** - Allow the participants to view new or unallocated items in the queue.
- **Can View Allocated Items** - You can allow the participant to view only the items that have been allocated.
- **Can Ad Hoc Pick Task** - You can allow the participant to pick up a task on an ad hoc basis
- **Can Plan Queue Item** - You can allow the participant plan to pick up a task
- **Can Put Back in Queue** - You can allow the participant to put back a task in the queue.
- **Can Move to Different Queue** - You can allow the participant to move the task to a different queue.
- **Can Recall Allocated Items** - You can allow the participant to recall a queue item that has been allocated.
- **Can Change Priority level** - You can allow the participant to change the priority level of the allocated queue item.
- **Can Do User Based Allocation** - You can allow the participant to allocate tasks based on users
- **Can Do Pattern Based Allocation** - You can allow the participant to allocate tasks based on patterns
- **Can Do Time Based Allocation** - You can allow the participant to allocate tasks based on time
- **Consider Queue Owner as Participant for Automatic Dispatch** - You can set the queue owner to be treated as a regular participant for the purpose of automatic dispatch.

Design time Access Rights: The design time access rights that can be set in this section allow administrators to set the access rights of participants during the configuration of the queue.

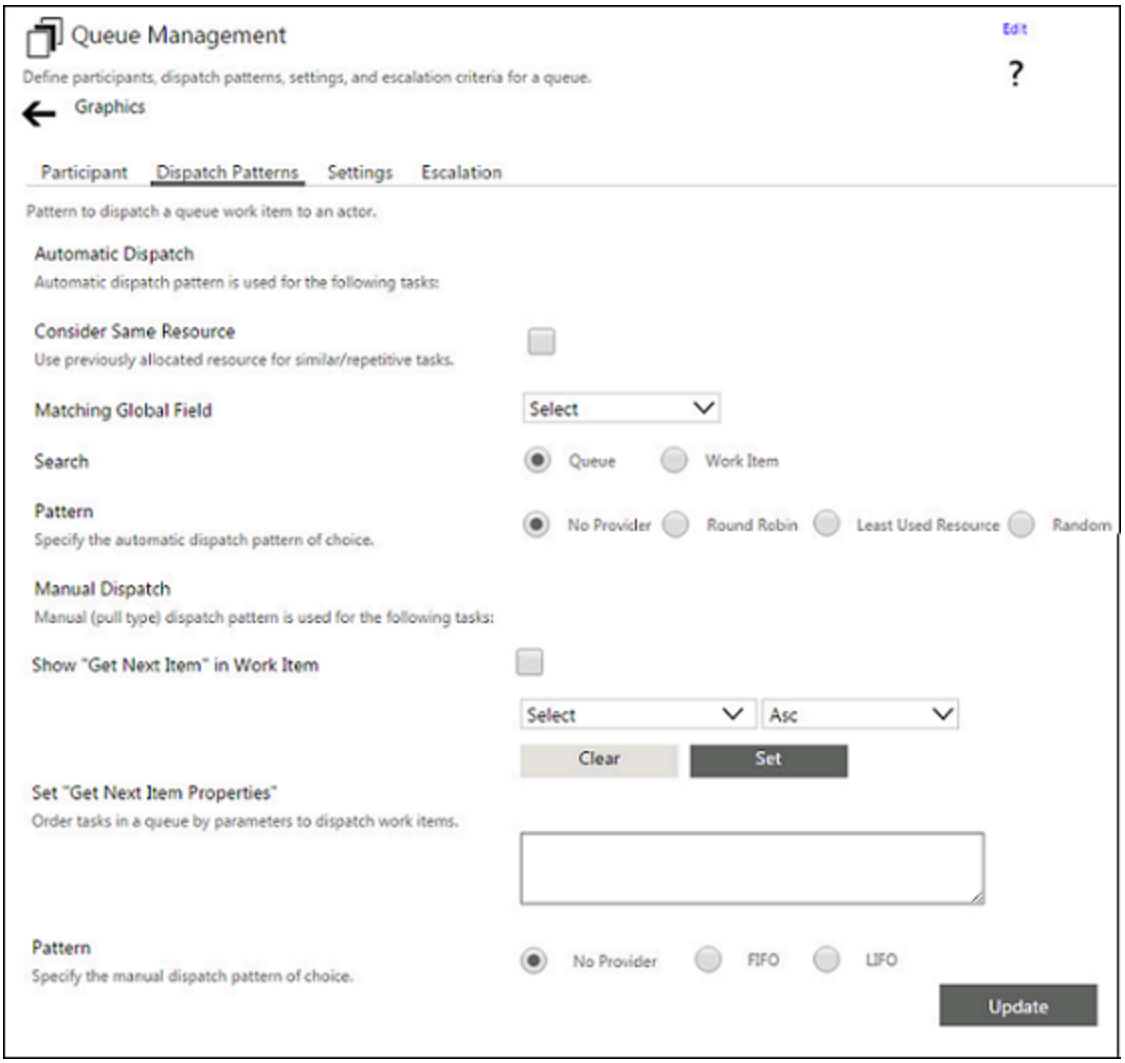
- **Manage Participants** - You can allow the participant to manage other participants
- **Set Access Rights** - You can allow the participant to set access rights for other participants.
- **Change Settings** - You can allow the participant to change the queue settings.
- **Manage Dispatch Patterns** - You can allow the participant to manage dispatch patterns.
- **Manage Escalation** - You can allow the participant to manage escalation.

Dispatch Patterns Tab

Click the **Dispatch Patterns** tab to open the dispatch patterns setting view of the Queue Management window. You can either configure automatic dispatch settings or manual dispatch settings here.

The following image shows the **Dispatch Patterns** tab of the **Queue Management** window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Automatic Dispatch

You can configure the following settings for automatic (or push type) dispatch of tasks:

- **Consider Same (Previous Allocated) Resource For Similar/Repetitive Tasks** - You can check this option to allocate a task, which is similar to a previously allocated task, to the same user to whom it was allocated in the first case. This can also be used to consider the same user for tasks that are repeated.
- You must specify the global field to be used to identify if a task is similar to an earlier one or is repeated by selecting from the **Matching Global Field** drop down.
- Specify the location to search for the matching task using the Search In radio buttons.
  - **Queue** - If you select this option, a matching task will be searched for in the queues.
  - **Work item** - If you select this option, a match will be searched for in the Work items list.
- Specify the automatic dispatch pattern next by selecting one of the following radio buttons in the **Specify Dispatch Pattern** option:
  - **No Provider** - This option specifies that no provider is available for dispatch. This option cannot be selected for both Automatic and Manual dispatch. The Dispatch Pattern has to be set for at least one

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

mode.

- **Least Used Resource** - This option ensures that the queue control identifies the participant who has the lowest work load to assign the task to.
- **Round Robin** - This option enables round robin task allocation where the participants will be allocated tasks by rotation.
- **Random** - This option sets the queue to allocate the tasks in a random fashion.

Manual Dispatch

You can configure the following settings for manual (or pull type) dispatch of tasks:

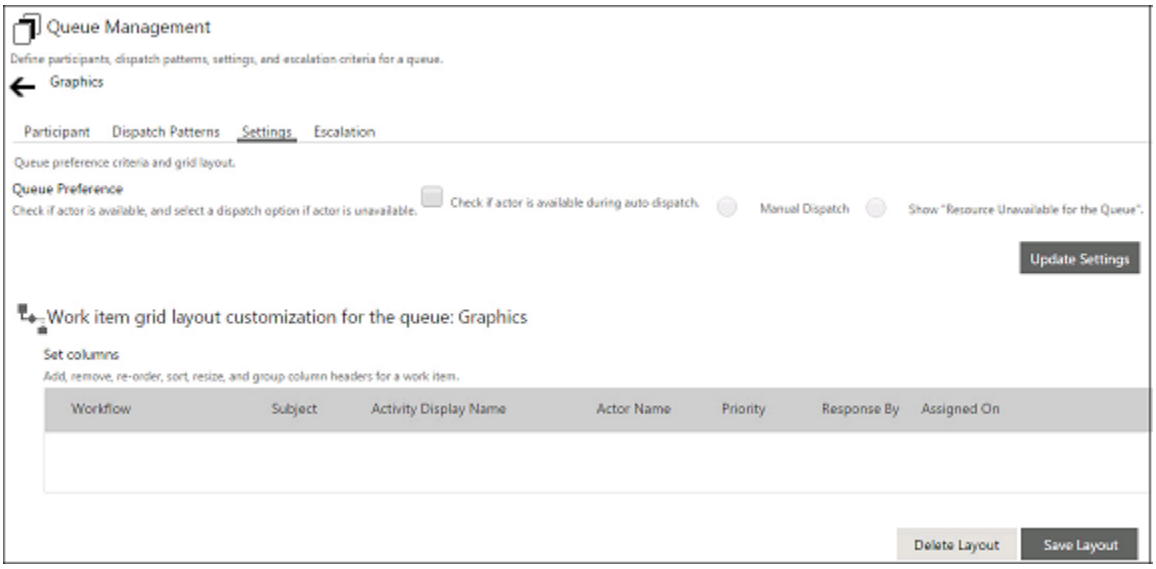
- **Show "Get Next Item" in Work item** - You can check this option to enable the display of the Get Next Item option in the Work items list. Participants can click this option after completing the allocated task to get the next task in the queue. You must set the following properties for the Get Next Item option:
- **Do OrderBy** - You must specify the order in which tasks in the queue must be sorted by selecting the parameters from the **Do OrderBy** drop-down and clicking Set. The parameters are separated by commas and can be applied in ascending or descending order by selecting **Asc** or **Desc** from the drop-down provided.
- Specify the manual dispatch pattern by selecting one of the following radio buttons in the Specify Dispatch Pattern option:
  - **No Provider** - This option cannot be selected for both Automatic and Manual dispatch. Dispatch pattern has to be set for at least one mode.
  - **LIFO** - This option sets the queue to allow users to access the tasks on a Last In First Out basis.
  - **FIFO** - This option sets the queue to allow users to access the tasks on a First In First Out basis.

After setting the Dispatch Patterns, click **Update Dispatch Patterns** to save the settings.

Settings Tab

Click the **Settings** tab to open the corresponding view of the Queue Management window.

The following image shows the **Settings** tab of the **Queue Management** window:



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

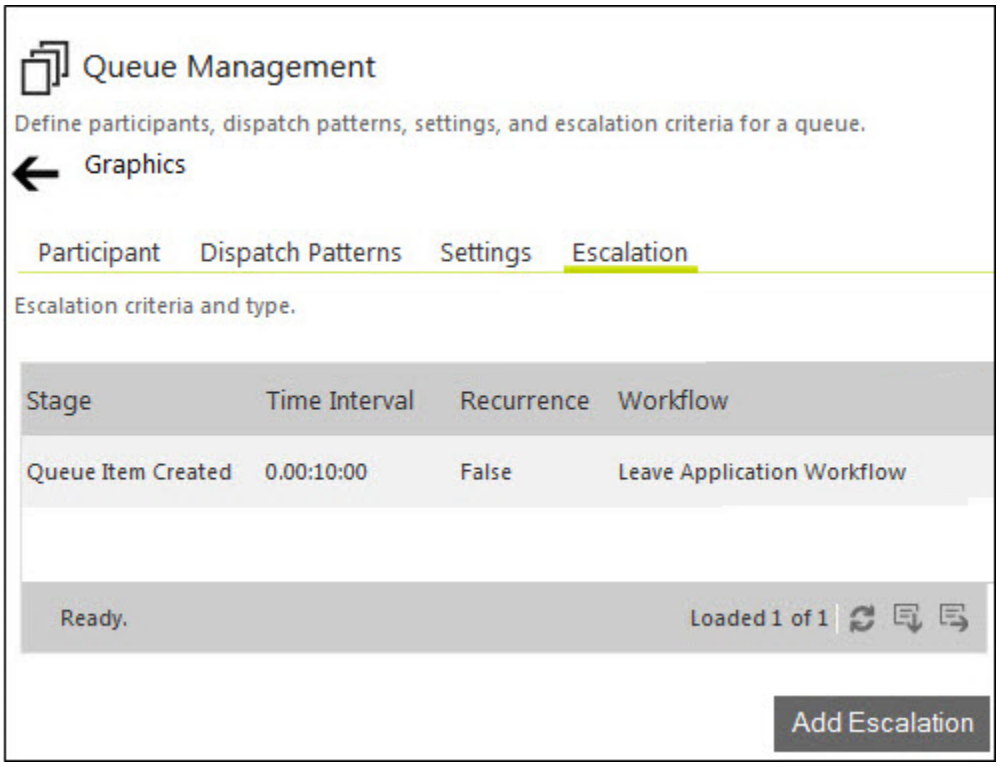
This view has the **Check if actor is available during auto dispatch**. You can select this check box to set the queue control to first check for the availability of actors before an auto dispatch of tasks is done. If you enable this option, you will need to select one of the two follow-on actions to be performed if the actor is unavailable. You can enable manual dispatch of the task by selecting the **Manual Dispatch** button. Alternatively, you can have the control generate the action output "Resource Unavailable For Queue" by selecting the second button **Show "Resource Unavailable for the Queue".** Click **Update Settings** to save the new settings.

The WorkList Customization for Queues section enables you to customize the view of work items displayed in the queue folder. For more information about worklist customization, see Define Layouts for Queues.

Escalation Tab

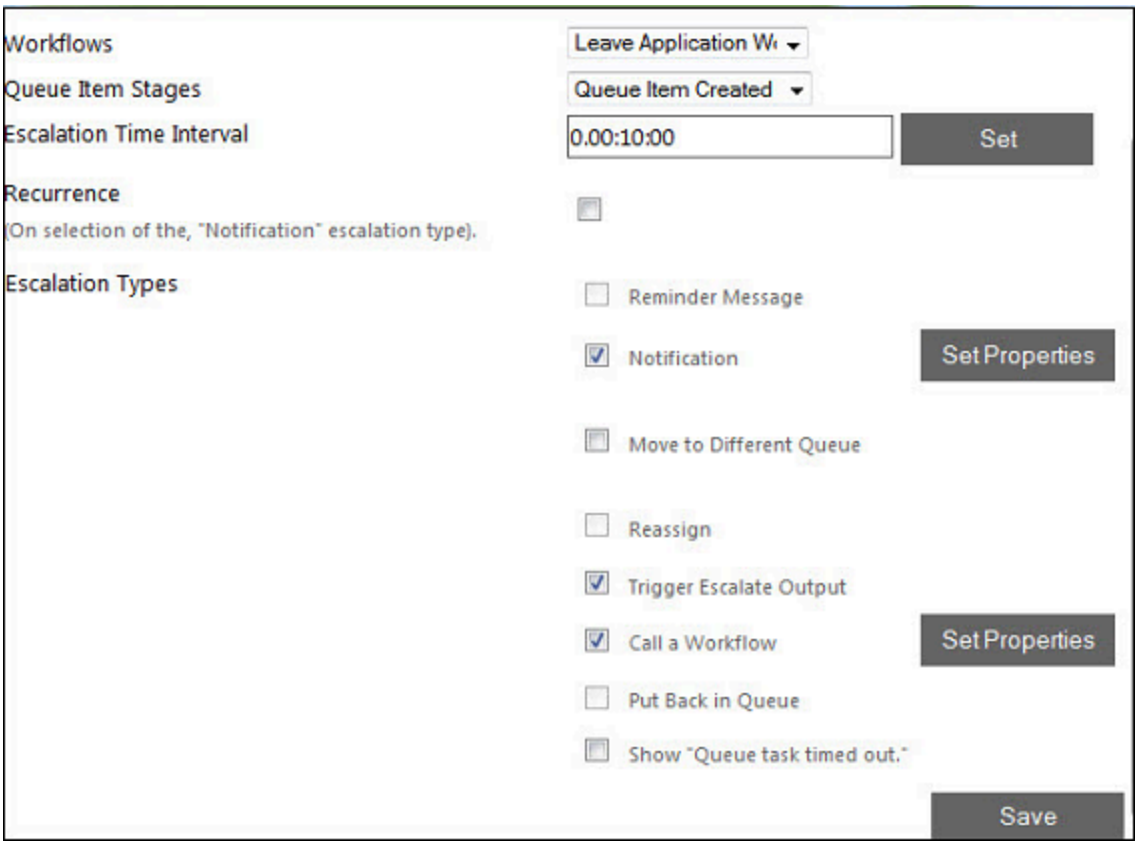
Click the **Escalation** tab to open the window for setting escalation properties.

The following image shows the **Escalation** tab of the **Queue Management** window:



This window displays escalations that have already been set in a grid. The details displayed include the escalation stage, time interval, whether the escalation should recur or not, and the escalation type. You can click on the **Add Escalation** button to add a new escalation. This will open the window shown below, where you can specify the escalation settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



You can specify the name of the Workflow corresponding to the queue. Next specify the stage in the queue at which the escalation should be triggered. You can select these two options from the respective drop down. Now, enter the Escalation Time Interval in the corresponding field. Select the Recurrence check box to set the escalation to recur.

After specifying the escalation settings, you need to specify and configure the escalation type using the controls under Escalation Types. The following check boxes corresponding to the different escalation types are available for selection:

- Reminder Message
- Notification
- Move To Different Queue
- Reassign
- Trigger Escalate Output
- Call a Workflow
- Put Back In Queue
- Trigger action output "TimeOut for Queue Task"

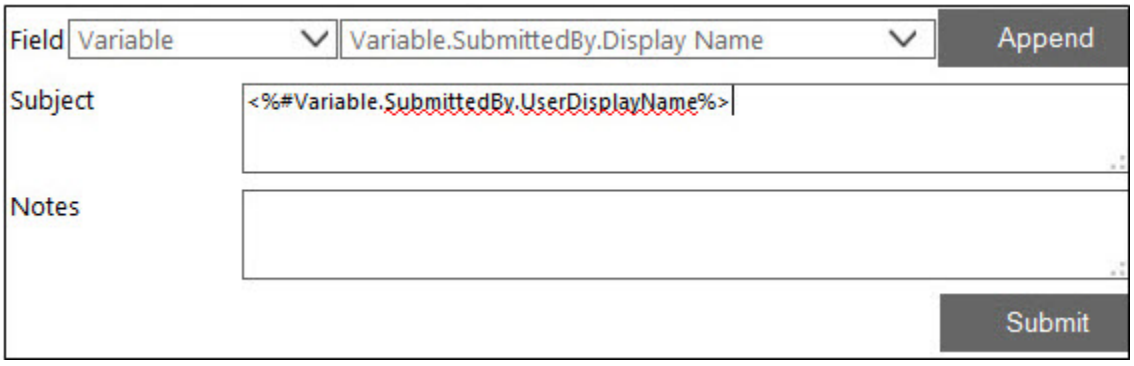
The options that can be accessed depend on the Queue stage selected earlier. For example, if the Queue stage selected is “Queue Item Created” then the options available are Notification, Move To Different Queue, Trigger Escalate Output, and Call a Workflow.

Reminder Message: To set the Reminder Message, click on the corresponding check box to select it and then click on the **Set Properties** button. This opens the window shown below.

Notification: To set the Notification escalation type, click on the corresponding check box to select it. Then click

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

on the **Set Properties** button to set the properties. This opens the following window:



Use Queue Control to Create and Manage Queues

You can add and define queue from the Queue Management page.

To add and manage a new queue

1. In the Enterprise Console, click menu icon, click **Settings**, and then click **Queue Management**. The **Queue Management** page appears.
2. Click **New Queue** to add a new queue.
3. In the **Name** box, type the unique name for the queue.
   Ensure that the **Queue Name** does not contain more than 250 characters, and does not contain the following special characters: [#~&+\"\'<>|@!$%*()=/{}?]\
4. In the **Description** box, specify the description for the queue.
   Ensure that the **Queue Description** does not contain more than 150 characters.
5. Click **Add**. The Queue is added and a new page appears that allows you to manage participants by adding the Users or Roles for the queue and setting their access rights. You can allow the users to view work items, pick up one, or pick the next available task without being able to view it, or put back a task in the queue.

You need to specify the Queue settings, set dispatch rules, and set escalations using the appropriate tabs in this window. For more information, refer to Configure Queue Management.

Add Participants to the Queue

You can add Users or Roles as participants in the queue. You can also set the priority for a role.

To add users to the queue

1. In the **Participants** tab, click **Add Users**. The **Add Users** page appears.
2. Click the **User Lookup** icon. The **User Lookup** page appears.
3. In the **Search for** box, specify the name or part of the name. A list of user(s) is displayed in the **Available Resources** box.
4. From the **Available Resources** list, select the required user name and then click the [>] icon to add the name to the **Selected Users** list.
5. Repeat the steps 3 and 4 to add more users.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Click **Update**.

To add roles to the queue

1. In the **Participants** tab, click **Add Roles**. The **Add Roles** page appears.
2. Click the **Role Lookup** icon. The **Role Lookup** page appears.
3. In the **Search** box, specify the role that you want to add. A list of role(s) is displayed in the **Available Roles** box.
4. From the **Available Roles** list, select the required role and then click the [>] icon to add the role to the **Selected Roles** list.
5. Repeat the steps 3 and 4 to add more roles.
6. Click **Update**.

Note: AVEVA Work Tasks requires a database setting to synchronize the users who are participants of a queue by virtue of their role, if the role changes in the resource database. For more information, *see* the *Synchronizing Roles of Queue Participants topic* in the *Developer Guide*.

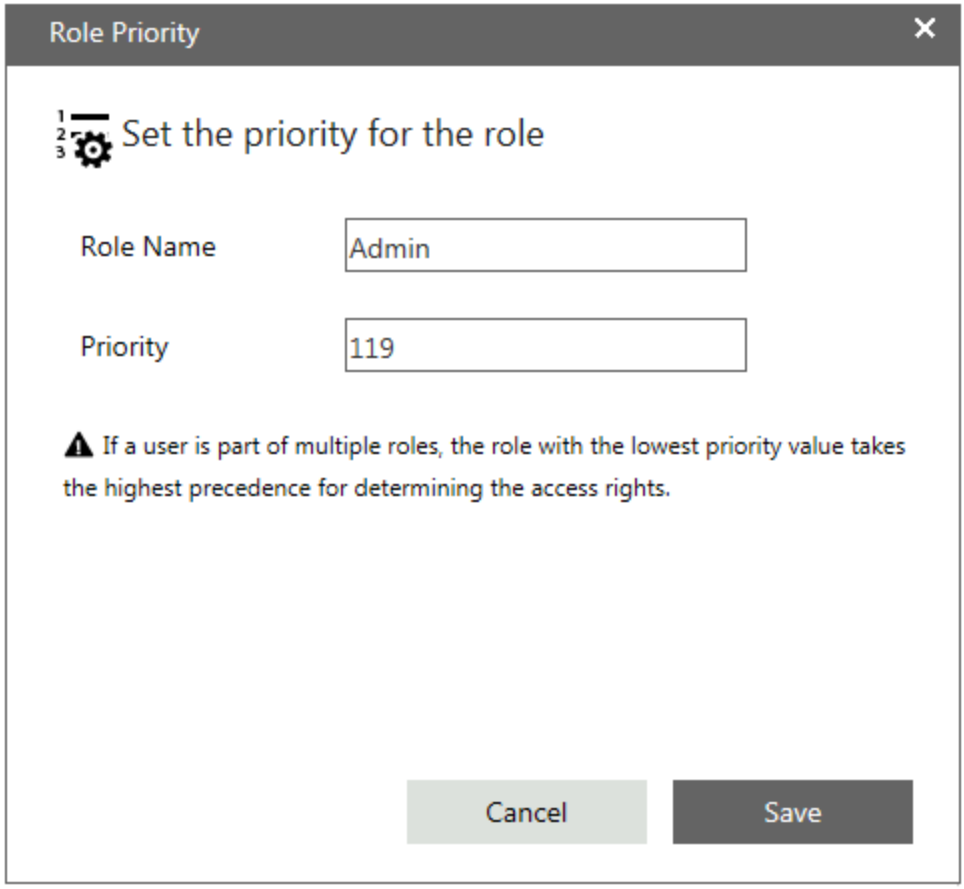
Tip: You can add all the users (or all the roles) at once by clicking the **[>>]** icon in the respective window.

To set the priority for roles

When you associate a user to a role, the user gets the access rights configured for the role. If the user is associated to multiple roles, you can define the priority for a role. The role priority determines which role takes precedence. The user gets the access rights of the role with the lowest priority value.

1. In the **Participants** tab, select the role for which you want to set the priority.
2. Right-click the role and select **Set Priority** from the context menu. The **Role Priority** dialog box appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



In the **Role Name** field, the name of the role for which you want to set the priority is displayed.

3. In the **Priority** field, type the value to indicate the priority of the role.

   You can specify the value from "1" to "9999". The default priority value for a role is "9999" and for a user is "0".

   Set the value to a lower number if you want the role to have a higher priority.

   Note:
   - It is recommended to specify the priority value in multiples of 10.
   - If a user is part of multiple roles, the role with the lowest priority value takes the highest precedence for determining the access rights.
   - If a user is added as a direct participant to the queue, then the access rights defined for the user take precedence, even if the user is associated to a different role.
   However, if the user is not added as a direct participant to the queue, the access rights for the user are based on the priority value of the roles.

4. Click **Save**. The priority is set for the role.

Configure Dispatch Patterns

You can configure the Dispatch Patterns for the queue from the Queue Management interface. You can set rules for both automatic Dispatch (push type) and manual Dispatch (pull type).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To specify the automatic dispatch pattern

1. Open the queue for which you want to specify the automatic dispatch pattern.

   The **Queue Management [Queue Name]** page appears.
2. Click the **Dispatch Patterns** tab.
3. In the **Automatic Dispatch** section, select the **Consider Same (Previous Allocated) Resource For Similar/ Repetitive Tasks** check box.

   This will assign the task that is a repetition or resembles an earlier task to the same resource who had handled the similar task earlier.
4. In the **Matching Global Field** list, select the global field.

   If you select the same resource, you will need to specify the global field that you want to use to identify if a task is similar to an earlier one or is repeated.
5. In the **Search In** option, specify whether the queue should search for a matching task by selecting one from **Queue** or from **Work item**.
   a. **Queue**: Click **Queue** if you want to search for the matching task in queues.
   b. **Work item**: Click if you want to search for the matching task in work items.**Work item**
6. In the **Specify Dispatch Pattern** option, click any of the following to assign tasks to the participants:
   a. **No Provider**: Click **No Provider** if you do not want any provider for automatic dispatch. If you select this, you must select a manual dispatch pattern other than No Provider.
   b. **Least Used Resource:** Click **Least Used Resource** if you want to assign the task to the participant with the lowest number of tasks assigned from the queue.
   c. **Random:** Click **Random** if you want to assign the task in a random order.
   d. **Round Robin**: Select **Round Robin** if you want to assign the task on a round robin basis to each participant.

To specify the manual dispatch pattern

1. Open the queue for which you want to specify the manual dispatch pattern.

   The **Queue Management [Queue Name]** page appears.
2. Click the **Dispatch Patterns** tab.
3. In the **Manual Dispatch** section, select the **Show Get Next Item in Work Item** check box.

   This option allows you to display the **Get Next Item** button and drop-down in the Work items list of participants. The participants can use this button to get the next item in the queue if the corresponding access rights have been set for them.
4. In the **Set "Get Next Item Properties"** option, select the item and specify the order in which tasks in the queue must be sorted by selecting from the **Do OrderBy** list.
5. Click **Set**.
6. In the **Specify Dispatch Pattern** option, click any of the following to assign tasks to the participants:
   a. **No Provider:** Click **No Provider** if you do not want any provider for manual dispatch. You can select this only when you have not specified the automatic dispatch pattern as **No Provider**.
   b. **LIFO:** Click **LIFO** if you want to assign the task to the participant on a Last In First Out basis.
   c. **FIFO**: Click **FIFO** if you want to assign the task to the participant on a First In First Out basis.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

7. After configuring the dispatch patterns, click **Update Dispatch Patterns** to save the automatic and manual dispatch patterns.

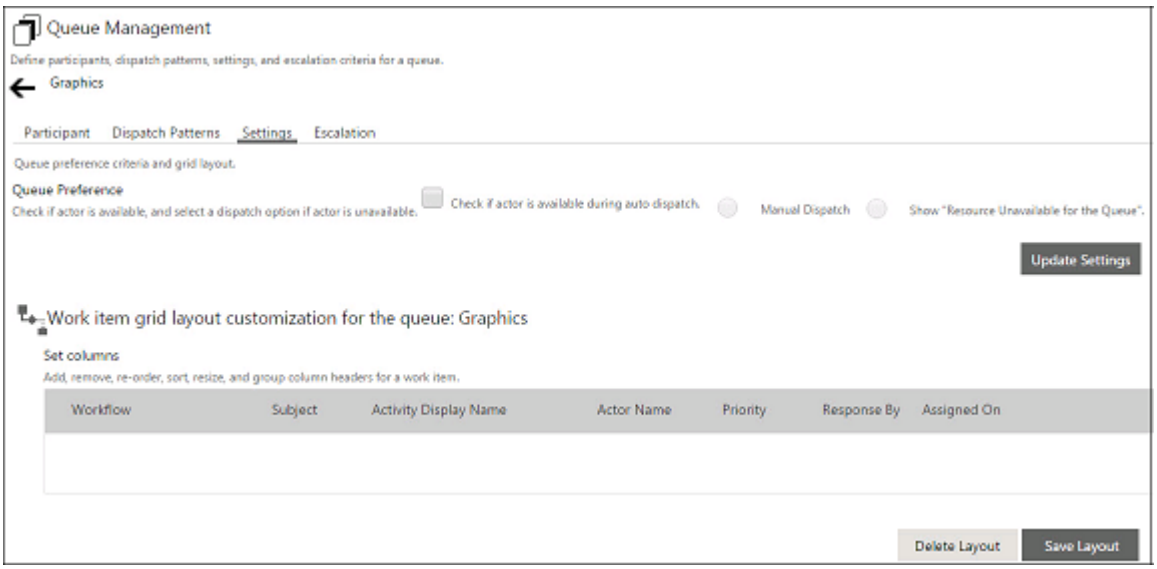
Set the Queue Preference

You can set the queue preference criteria and customize the grid layout for the queue work items.

To set the queue preference

1. Open the queue for which you want to specify the automatic dispatch pattern.

   The Q**ueue Management [Queue Name]** page appears.



2. Click the **Settings** tab.
3. In the **Queue Preference** section, select the **Check actor availability during auto dispatch** i**n case actor could not be identified** check box.

   You can select this check box to set the queue control to first check for the availability of actors before an auto dispatch of tasks is done. If you select this option, you must select any of the following follow-on actions to be performed if the is unavailable.

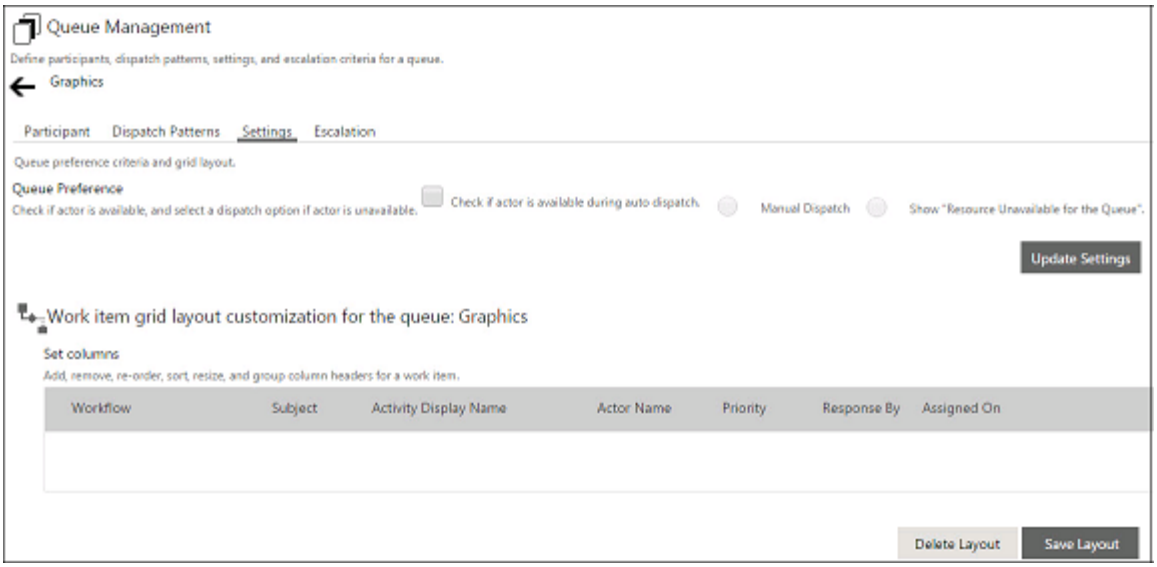
   a. Do a **Manual Dispatch**: Click **Do a Manual Dispatch** if you want to do manual dispatch of the task
   b. **Fire action output "Resource Unavailable For Queue":** Click **Work item** if you want the control to generate the action output "Resource Unavailable For Queue".
4. Click the **Update Settings**.

To customize work item for the queue

1. Open the queue for which you want to specify the automatic dispatch pattern.

   The **Queue Management [Queue Name]** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



2. Click the **Settings** tab.
3. The WorkList Customization for Queues section enables you to customize the view of work items displayed in the queue folder.

   For more information on worklist customization, see Define Layouts for Queues.

Set Escalations for Queues

You can set the Escalation for the queue from the **Escalation** tab of the queue management interface.

To set the escalation

1. In the **Escalation** tab, click **Add** to open the Escalation settings window.
2. In this window, specify the name of the workflow corresponding to the queue. Select the required workflow from the list for the **Workflows** field.
3. Next, specify the stage in the queue at which the escalation should be triggered. Select this from the **Queue Items Stages** list. Choose either **Queue Item Created** or **Queue Item Picked**.
4. Enter the **Escalation Time Interval** in the corresponding field.
5. Select the **Recurrence** check box to set the escalation to recur.
6. Select the required escalation type check box, under the **Escalation Type** section.
   a. If you select **Reminder Message**, it will generate a reminder message to be sent to the participant assigned a task from the queue
   b. If you select **Notification**, it will send a notification to the participant that the assigned task has not been acted upon
   c. If you select **Move To Different Queue**, it will move the task that has not been acted on to a different queue
   d. If you select **Reassign**, it will reassign the task that has not been acted on to a different participant
   e. If you select **Trigger Escalate Output**, it will trigger the "Escalate" output for the task that has not been acted on
   f. If you select **Call a Workflow**, it will execute the specified Workflow when the task has to be escalated

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

g. If you select **Put Back In Queue**, it will put the task that is not acted on back in the queue.

The following steps provide information to set the different Escalation Types.

Setting Reminder Message or Notification Properties

1. Select the **Reminder Message** or **Notification** escalation type check box.
   The **Set Properties** button appears next to it.
2. Click **Set Properties**.
   The **Escalation Properties** window appears.
3. Specify the subject of the Reminder Message or Notification either by entering a discrete value in the **Subject** box or by selecting an option from the **Field** list and then click **Append**.
4. Specify the message of the Reminder Message or Notification in the **Notes** box.
5. Click **Submit** to save the Reminder Message or Notification.

Setting Reassign properties

1. Select the **Reassign** check box.
   The **Set Properties** button appears next to it.
2. Click **Set Properties**.
   The **Escalation Properties** window appears.
3. Select the participant to whom you want to reassign the task.
4. Click **Submit.**

To set Call a Workflow properties

1. Select the **Call a Workflow** check box.
   The **Set Properties** button appears next to it.
2. Click **Set Properties**.
   The **Escalation Properties** window appears.
3. Specify the following:
   a. In the **Application Name** box, type the application name.
   b. In the **Workflow Name** box, specify the workflow name.
   c. In the **Version** box, specify the version details.
4. Specify the contents to be passed to the called Workflow by selecting the value type (variable, content, or value) from the first list for **Field** and selecting the corresponding variable or content collection from the second drop-down list.
5. Click **Append** to add the selection to the **Contents** text area.
6. Click **Submit**.

Use Queues in Workflows

In the Activity Properties pane of all activities, which require human intervention, there is an option called Assign

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Queue(s). You can use this property to specify a queue to which the corresponding work item will be assigned.

To assign a queue for an activity

1. In the **Process Designer for Workflow**, right-click the resource activity, and then select **Activity Properties**.
2. In the **Activity Properties** pane, click the button for the **Assign Queue(s)** property.
3. In the window that opens, select the queue(s) from the Available Queues list box.

   You can select multiple queues by selecting with the <Ctrl> key pressed.
4. After selecting the required queue(s) click the [>] button to move it to the Selected Queues list box. You can click the **[>>]** button to select all the queue(s) and move them to the **Selected Queues** list box.
5. You can remove any queue from the selected list. Selecting the queue and then click the **[<]** button. You can remove all the queues by clicking the **[<<]** button.
6. After adding the required queue(s), click the **Assign** button to assign the activity to the selection.
7. Click the **Close** button to close the window without saving the selection.

Difference between direct task assignment and task assignment through queue:

A task can get assigned to multiple resources directly whereas only one resource will get the task from a queue.

Access Queues in the Work Items List

The work items corresponding to activities assigned to queues will be delivered in the Work items list of all the actors who are participants in the respective queues. The Work items list will show separate folders for each queue that the logged in user belongs to. The tasks assigned directly to the user are listed for view in the Inbox.

To access work items assigned to the queues

1. For each queue, you can open a list of all tasks available for selection by clicking on the corresponding folder, provided you have the rights to view and select tasks. The task selected from the queue will be added to the Inbox.
2. If you have the rights to pick tasks from queues, you can select the queue from the drop-down list at the top and click on the **Next Item** button to retrieve the next available task from the queue. This task will get listed in the Inbox. The task is assigned on the basis of the dispatch provider algorithm selected during configuration. AVEVA Work Tasks has two dispatch providers, First-In-First-Out (FIFO) and Last-In-First-Out (LIFO). Users can create their own dispatch providers and configure them for use. The Next Item button is displayed even for participants who have been assigned no access rights.

Database Connections

This section provides information on how to create a database connection within AVEVA Work Tasks.

Create a Database Connection

You can create a new database connection to connect to either a repository datasource or an external datasource.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To create a Database Connection for a repository datasource

1. Log on to the Enterprise Console.
2. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Database Connections**.
   The Connection List page appears.
3. Click **New**.
   The **New Connection** dialog box appears.
4. In the **Title** box, type the name of the database connection.
5. In the **Description** box, type any additional details for the database connection.
6. Select the **Connect to repository datasource** check box to connect to the current repository database.
   A database connection for the repository datasource is successfully created and is displayed in Connection List page.

To create a Database Connection to external datasource

1. Log on to the Enterprise Console.
2. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Database Connections**.
   The Connection List page appears.
3. Click **New**.
   The **New Connection** dialog box appears.
4. In the **Title** box, type the name of the database connection.
5. In the **Description** box, type any additional details for the database connection.
6. Clear the **Connect to repository datasource** check box to connect to an external database.
7. Perform the steps mentioned in the following tables.

Field	Description
Database type	Select SQL Server or Oracle. If the database type is Oracle, ensure to install Oracle Client on your machine.
Connection type	Select either Connection string or Connection provider.

If Connection type is Connection string

Field	Description
Server name	Perform one of the following: • Enter the name of the datasource server. • Click the **Refresh** icon to get a list of the available datasource servers and then select the required

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	server. If the database type is Oracle, ensure that you enter the Oracle Service name, which is used to connect to the Oracle database server.
Authentication type	Provide information on the process used to validate a user. Both Windows, and Microsoft SQL Server authentication process is supported. • For Microsoft SQL Server, the authentication type can be either Microsoft SQL Server or Windows.
User Id	Enter the Microsoft SQL Server user ID for Microsoft SQL Server type of authentication. For Windows Authentication, **User Id** is disabled.
Password	Enter the Microsoft SQL Server password for Microsoft SQL Server type of authentication. For Windows Authentication, **Password** is disabled.
Advanced settings	(Optional). You can enter any additional connection information, such as connection timeout. For example, Connection Timeout = 20 While creating a data source for a repository, use the following custom settings to set properties for connection pool: `encrypt=true; enableltm=true; connectionretrywaittime=30; maxconnectionretries=10; sqlcommandtimeout=300;` where, • **encrypt**:Ensures secure communication between AVEVA Work Tasks Server and SQL Server when a certificate is installed on the SQL Server for SSL encryption. • **enableltm**: Internally light weight transaction mode is implemented by reusing the connection which is already open within the same transaction. With this implementation, a connection which is used from a transaction is returned back to connection pool only when the transaction is completed. Since the same

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Field	Description
	connection is used within the transaction, the transaction distribution is reduced to a great extent. With this feature the transaction is controlled from being escalated to a distributed transaction which is very expensive. The above property can be used to turn off this feature. The default value is set as *True*. • **connectionretrywaittime**: While opening a new connection, if the database server is not available, the default behavior is to retry the connection. This setting can be used to set the wait time to retry for a new connection. The default value is set as 30 Seconds. The value should be set in seconds. • **maxconnectionretries**: While opening a new connection, if the data base server is not available, the default behavior is set to retry for a new connection. This setting can be used to set the retry count. The default value is set infinite. • **sqlcommandtimeout:** Sets the wait time before terminating the attempt to execute an SQL command and generating an error. The default value is set as 300. **Note:** These settings are case sensitive and should be mentioned only in lower case.
Database	Perform one of the following: • Enter the name of the database. • Click the **Refresh** icon to get a list of the available databases and then select the required database.

If Connection Type is Connection Provider

Field	Description
Assembly path	Enter the path of the connection dll. In a client-server environment, the dll must be present on the machine that hosts the Enterprise Console site. All requests of install and uninstall are sent to the host site.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Work Items List

The Work items list provides enhanced usability features. Workflow actors can view and respond to work items assigned in a user-interface that is similar to Microsoft Office Outlook. Work items are delivered to the actor's Inbox with a detailed view available on selection of the item. The name of the logged-in user is displayed at the top right corner of the window.

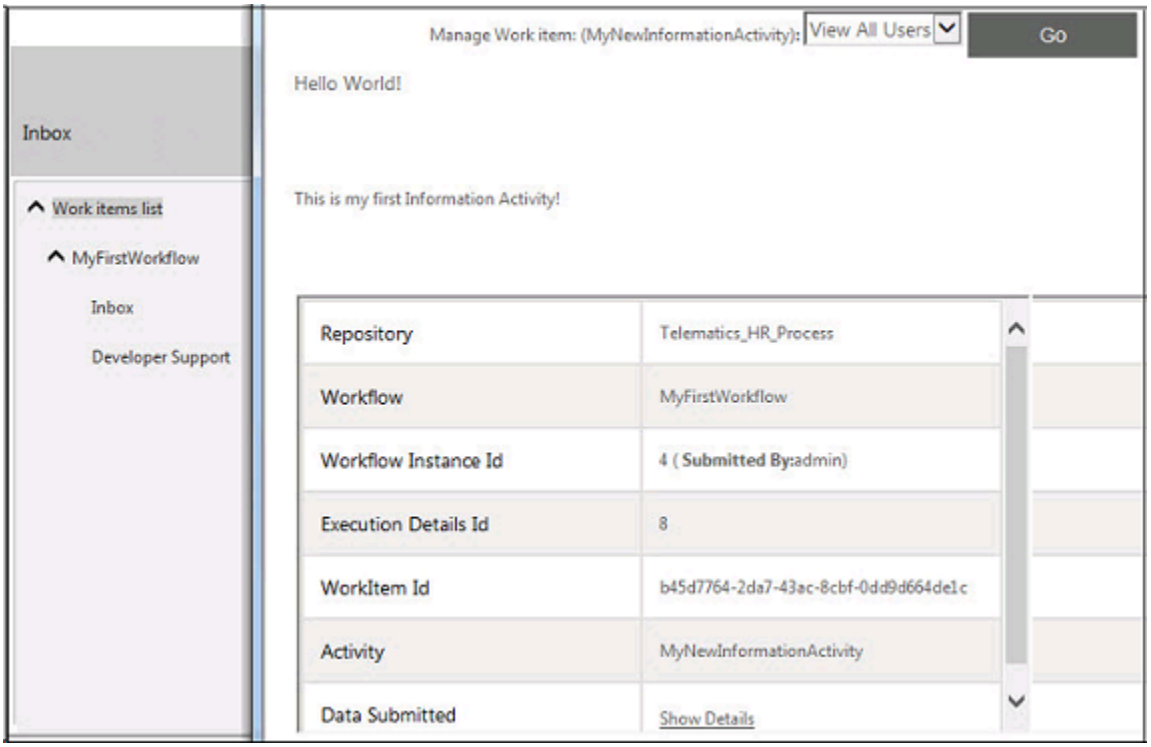
The detail view of the Work items list has the following user interface elements:

- Worklist Folders
- Work items list
- Work item details
- Toolbar

The Work items list also provides properties that enable customization and configuration of elements of the user interface. For details on customizing the view, see Customizing Work Items List Views.

The Work items list allows participants in queues to manage items in their queues. A separate Manager View forms a part of the Work items list. It allows actors who are also managers to administer activities assigned to their team members.

Work items list - Detail View



Note: When you click a work item to edit, the editor appears in a modal window. You have to close the modal window to access the parent window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

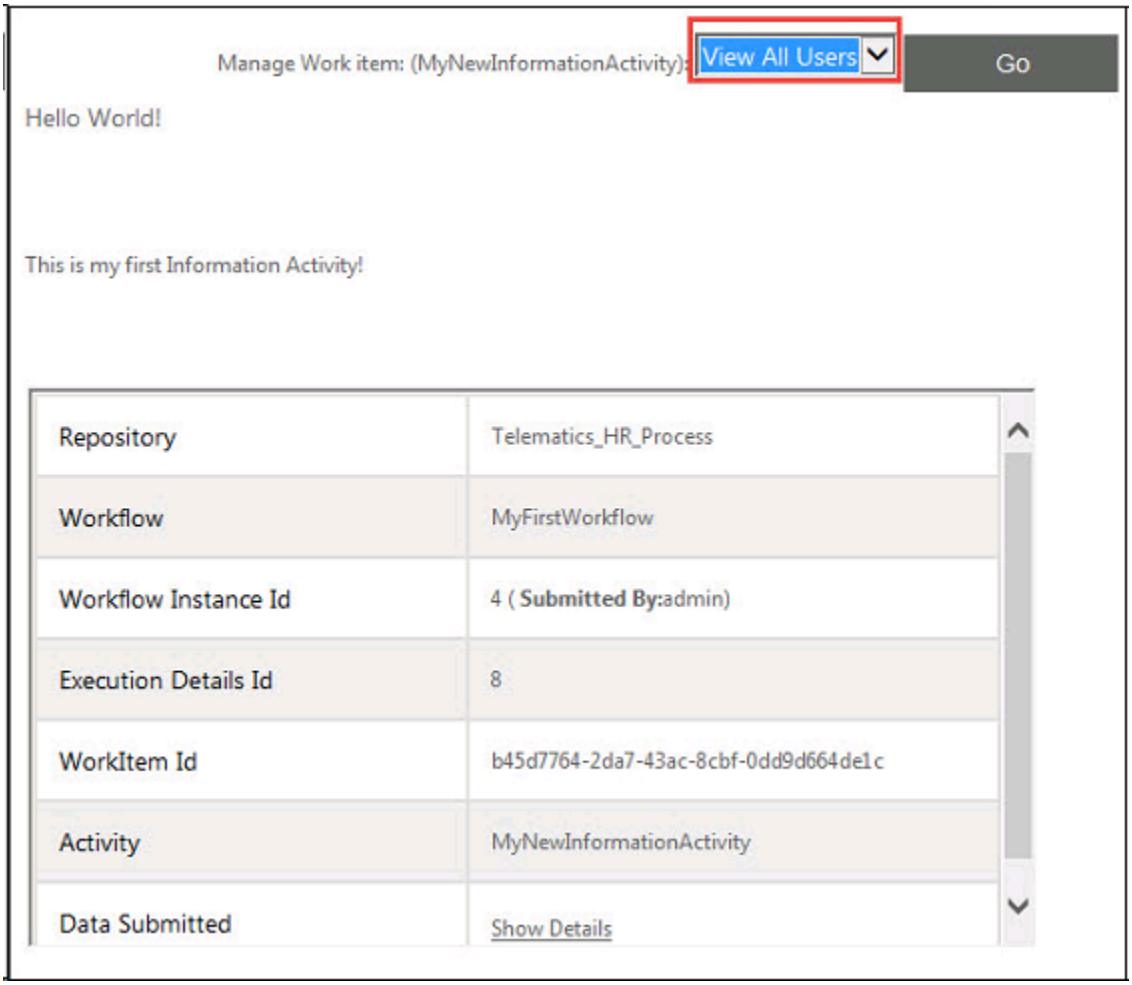
Manage Work Item - View All Users

The **View All Users** screen from the Work items list captures the comments given by all users and also groups them under Action Taken categories.

Consider a leave approval scenario where the approval task is assigned to three users, namely John, Ben, and Ken.

Suppose that the approval task has been acted upon by John, Ben has taken ownership of the work item, and Ken's status is awaiting.

After Ben takes ownership of the work item, the **View All Users** option in the **Manage Work item** drop-down is seen.



Work Items List User Interface Elements

The Work items list provides enhanced usability features. Workflow actors can view and respond to work items assigned in a user-interface that is similar to Microsoft Outlook. Work items are delivered to the actor's Inbox with a detailed view available on selection of the item. The name of the logged-in user is displayed at the top right corner of the window.

The detail view of the Work items list has the following user interface elements:

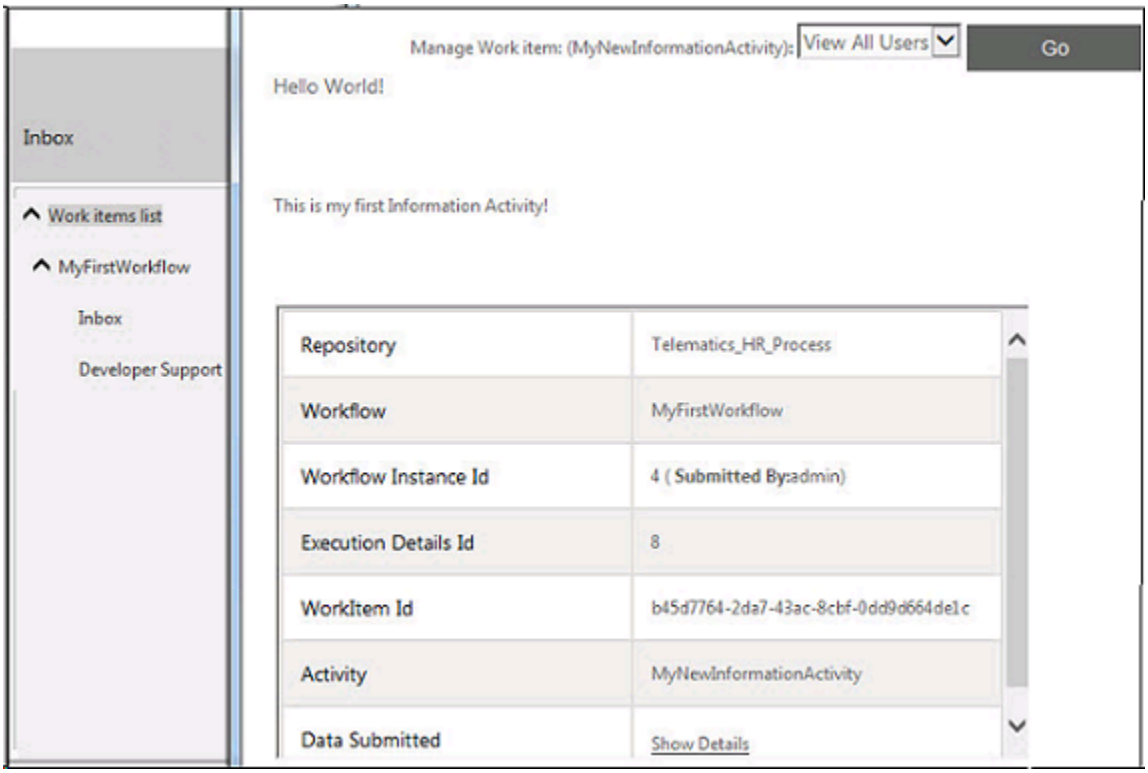
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Work Items List Folders
- Work Items List View
- Work Item Details
- Work Items List Toolbar

The Work items list also provides properties that enable customization and configuration of elements of the user interface. For details on customizing the view, see Customize Work Items List.

The Work items list allows participants in queues to manage items in their queues. A separate Work Items List Manager View forms a part of the Work items list. It allows actors who are also managers to administer activities assigned to their team members.

Work items list - Detail View



Note: When you click a work item to edit, the editor appears in a modal window. You have to close the modal window to access the parent window.

Work Items List Folders

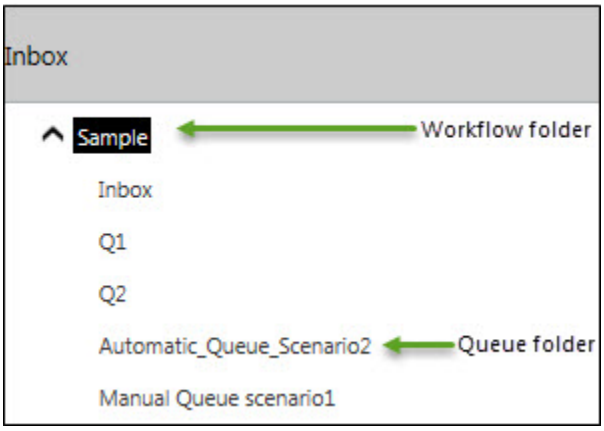
The worklist folders list all folders that contain work items. The **Work items list** folder is the top-level folder. In the default view, the Work items list folder contains sub-folders for each workflow from which tasks have been assigned.

The workflow folder has an **Inbox** folder and a folder for each queue in the workflow that the actor is a participant of.

The default layout of items in the Inbox can be customized and saved as a personalized folder. All items in the Inbox will also be available in the personalized folder with the specified layout. See Customizing Inbox View for

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

more information.



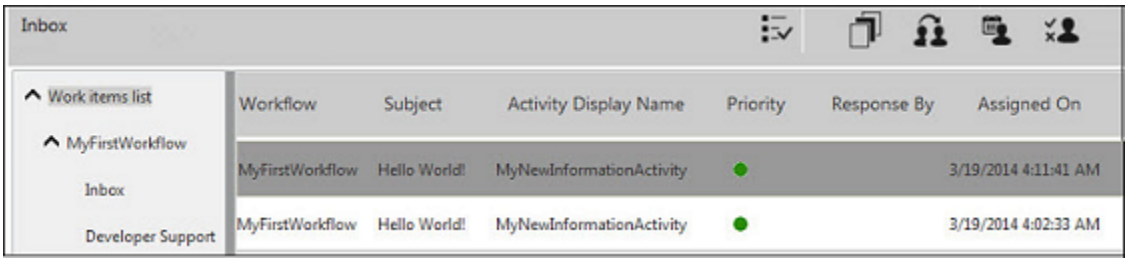
Work Items List View

The Inbox lists all open work items assigned to the user that awaiting action or response. The columns displayed in the default view are:

- Workflow
- Subject
- Activity Display Name
- Priority
- Response By
- Assigned On
- Approver1

The **Subject** column lists work items in bold type. When the user takes ownership of the item, its display changes from bold to normal. Double-clicking on an item displays its details in Work Item Details.

Default columns in the Work items list



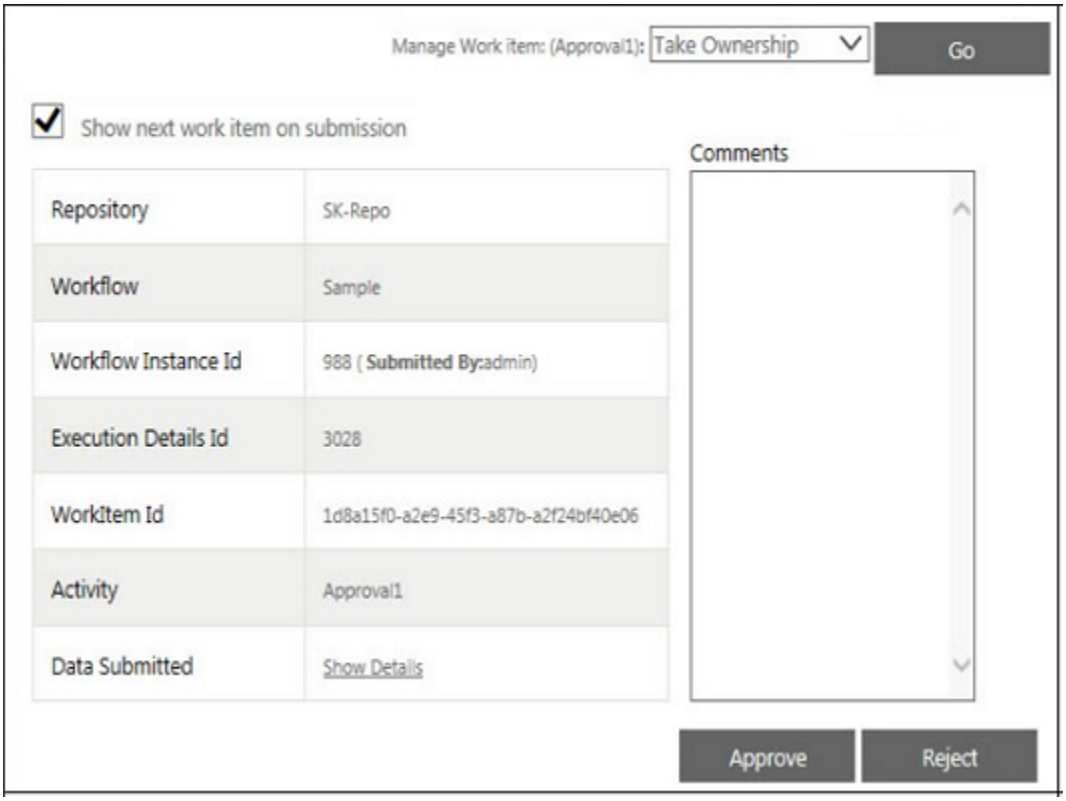
See also: Customize Work Items List Views

Work Item Details

The following image shows the Work Item Details for a selected work item in a Work items list. You can view and respond to work items using the **Work Item Detail View** modal window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Work Item Details



The **Work Item Detail View** dialog box displays the following items:

Manage Work Item: The Manage Work item drop-down has options to handle the work item. Select Take Ownership to take ownership of the item. Alternatively, you can select the **Hold** option to put it on hold. Click **Go** to submit the response. By setting the **AllowAutoownershipOnRowClick** property, you can allow users to take ownership on selection of the work item. See Preferences for Work Item Management for more information.

After you take ownership of a work item, you can have the option to view all user responses through the **View All Users option**. The Work items list property of **Hide Response** should be set as **No** to view all users responses. See *Manage Work Item - View All Users* for more information.

Custom Document View: The Custom Document View is at the left of the detailed view. It displays information on Execution Details.

Work Item Action Controls: The Work Item action controls are located at the right of the detailed view. The controls and the content depend on the activity type. The figure given below displays controls for the Approval work item with a field for comments. The controls and information displayed for an activity also depend on the properties of the activity as set in the process. For example, if the property Hide Responses for a group approval activity is set to False, users will also be presented with a link to view comments from other approvers.

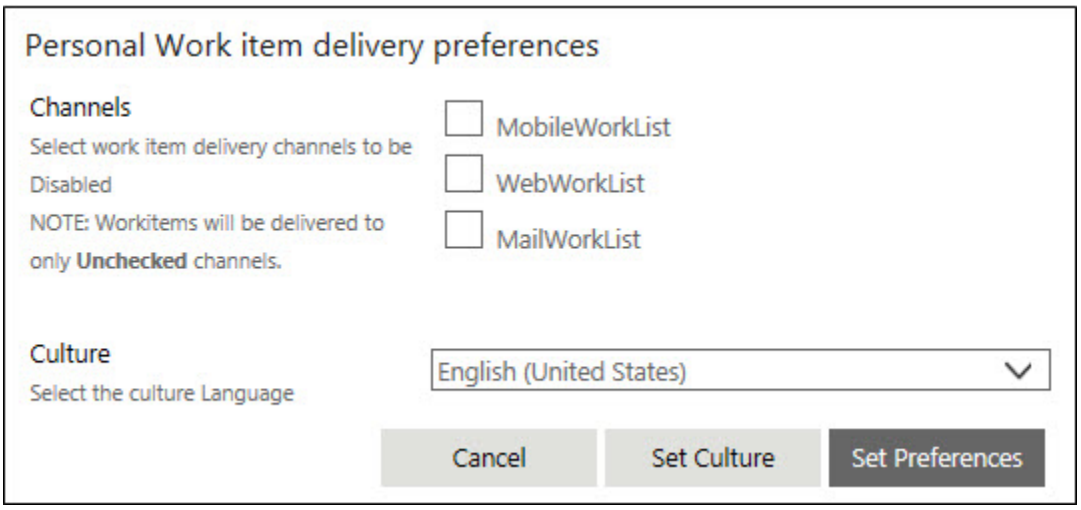
Work Items List Toolbar

The Work items list toolbar has options that enable users to manage how work items are delivered, when they are delivered and to assign alternate actors in their absence. An additional option for queue enables queue owners to access the settings to their queues.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

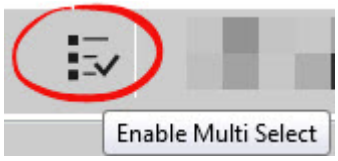
- **Show Queue:** This option opens the appropriate Queue Control window for the folder from which it is accessed. To open the main Configure Queue Management window with the list of all queues, select Inbox and click the Show queue icon. To open the Participants Tab window for a queue, select the queue folder and click the icon.
- **Show alternate actor:** This option opens the **Alternate Actors** setting window, which can also be accessed from the Process Designer. See Alternate Actors for more details.
- **Show resource calendar:** This option opens the **Resource Calendar** window for the active resource. You can set the resource calendar preferences here. See Resource Calendar for more details.
- **Set preferences:** This option allows actors to set preferences for the delivery of work items. It opens the Personal Work item delivery preferences window, where you can disable the delivery channels that you do not wish to receive the work items on.
- To disable a channel, select the check box corresponding to it. For example, if you are traveling and may not have access to the web, select the **WebWorkList** option to stop work items from being delivered in the Work items list. Click **Set Preferences** to save.

This option also allows users to specify the language settings for the user interface. In **Select the culture language**, select the language. Click **Set Culture** to set the language preference.



Note: The **Culture** drop-down displays a list of languages. However, for the preference to take effect, the language strings for the corresponding language must be available in AVEVA Work Tasks folders. AVEVA Work Tasks does not provide default support for all languages in the drop-down.

Enable Multi Select - The Enable Multi Select button provides users with options of acting on items as a batch. An additional option is made available when the items are grouped.



The Multi Select option is not supported for **Resource Set** activity and **Information** activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Work Items List Manager View

The **Manager View** folder displays a list of all work items of all users reporting to the logged in user. This view enables managers to quickly access the status of the work items assigned to reporters. Work item details are not displayed in this view.

Manager View Options

The following options are available from the right-click menu for an item:

- **Reassign Work Item** - Enables work items to be reassigned to a different user. Reassigning the work item will remove it from the current user's work item list and transfer the item to the user to whom it has been reassigned.
  Once the work item is reassigned to a different user, the pop-up window will be closed automatically and the user will not be able to act upon the work item by reopening it.
- **Delegate Work Item** - Enables work items to be delegated to a different actor. When a work item is delegated, it continues to appear in the list of the original user.

Queue Multi-Select and Reassign Queue Work Items

Queue Multi Select

The queue multi-select feature allows you to select multiple queue items. You can pick the queue items from the work item Inbox and assign them to actors. You can plan the queue items so that they are scheduled for the time specified and assigned to actors once the time elapses. You can also recall already allocated items so that they get back to the respective queues.

Showing the multi select capability for the Queue item tasks in Work items list control is controlled by the Work item list Control Property - **EnableMultiSelectByDefaultForQueue**. Based on this property, the set user can control the visibility of the multi-select check boxes for the queue view.

If the property **EnableMultiSelectByDefaultForQueue** is set to False for the Work items list control, then the queue multi-select option is controlled by the **DisplayGroupUI** property of the Work items list control. On click of the **Enable Multi Select** icon, check boxes will be enabled for selecting multiple queue items.

The following image shows the Queue multi-select option in Work items list control:

Note: The options for performing queue operations are displayed in the **Inbox,** based on the runtime access rights defined for a role in Queue Management. For more information about setting role priority, see Set Role Priority.

Reassigning Queue Work Items

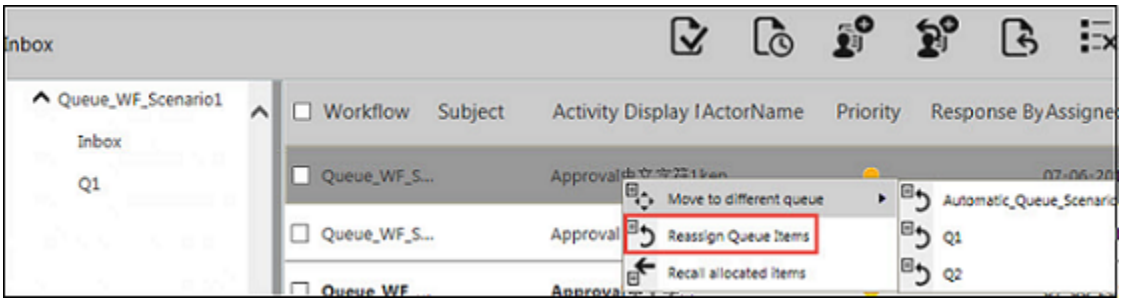
This option enables you to reassign queue work items through the context menu. Also, the scenario of reassigning multiple queue work items is achieved by clicking Reassign Queue Items in the multi-select mode.

Reassigning of queue work items functionality is achieved through the context menu of the queue work items

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

list.

The following image shows reassigning queue work items:



Multiple reassigning of queue work items is achieved by clicking the **Reassign Queue Item** button with the multi select mode enabled.

Access Queue Items

The work items list shows a separate folder for every queue that the user is a participant in. When a work item from the queue is assigned to the actor, it is moved to the Inbox. The folder displays all items that are assigned to the queue. However, the display of all queue items is subject to the rights assigned to the user for the queue.

When the queue is configured for manual dispatch and the actor has access rights to pick the next item from the queue, the actor can view the **Next Item** button at the top of the Work items list along with the corresponding drop down. After completing the current assigned task, the actor can select the queue from the drop down and click on the button to pick the next item for action.

For more information about setting rights for queue participants, see Participants Tab.

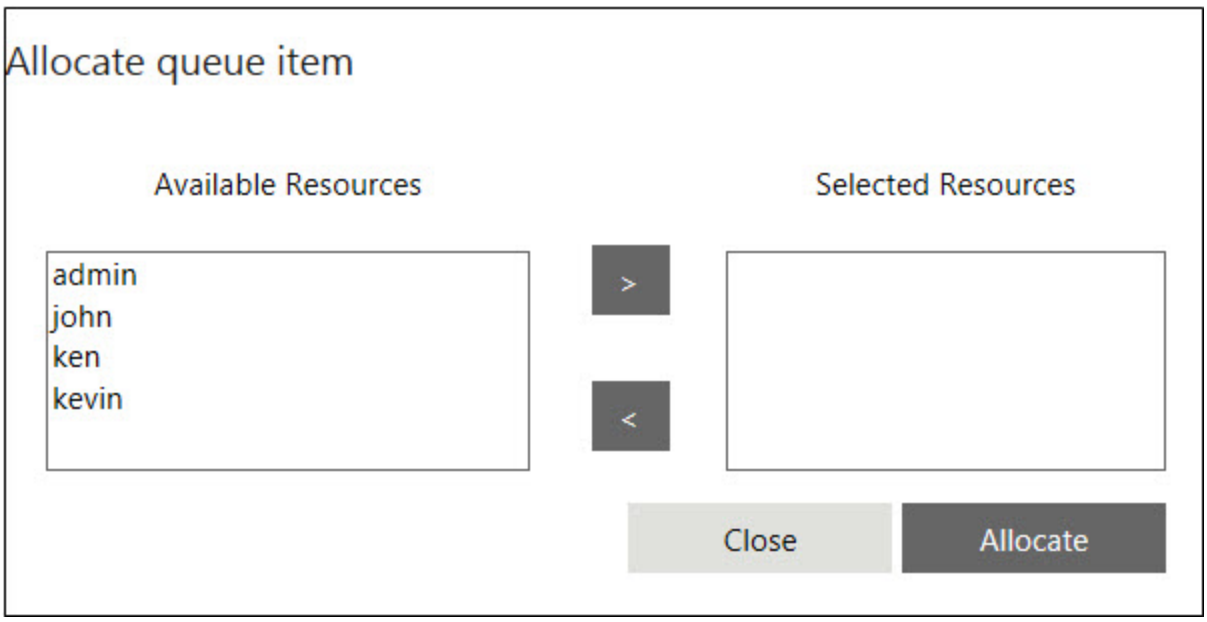
Queue Item Options

Depending on the queue configuration and the participants, the context-sensitive menu for the queue item lists the following options:

Allocate queue item

Allocates the item to a participant. This option brings up the **Allocate Queue Item** window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



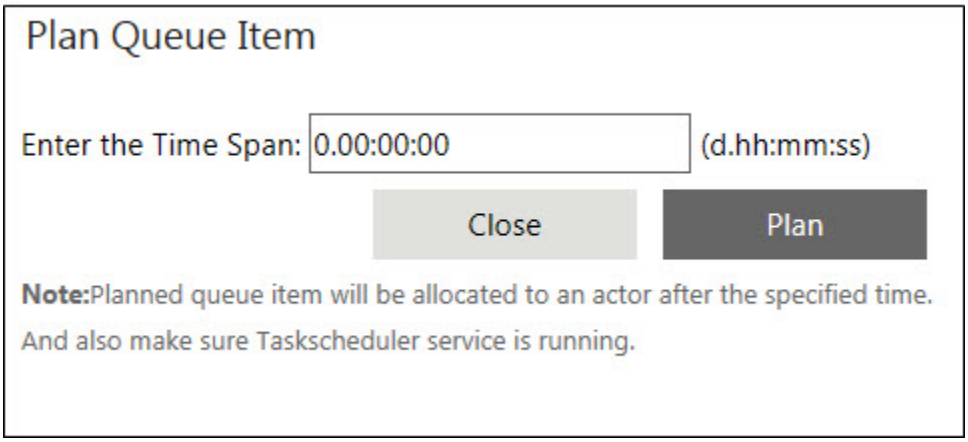
Move to different queue

Moves the item to a different queue. This option will drop another menu with a list of available queues.

Set Priority: Set the priority of the queue item by selecting the required priority level (High, Medium, or Low) from the secondary menu.

Pick: Allows selection of the participant who will act on the queue item. The actor is picked based on the dispatch pattern that has been set.

Plan: Allows users to plan to get the queue item at a specified time. Selecting this option opens the **Plan Queue Item** window to enter the lead time for picking up the item.



Cancel Planned Queue Work Item

- Allows users to cancel the planned queue work item and move the work item back to the queue.
- This option is displayed in the context menu only for the queue work items that are already planned.
- If the user wants to cancel a planned queue work item, the user needs to right-click the planned work item and select **Cancel Planned Item** from the context menu. The work item will be moved back to the queue.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Customize Work Items List

AVEVA Work Tasks allows the following types of customization:

- Customize Work Items List Views
- Customization of Layouts
- Preferences for Work Item Management

Customize Work Items List Layouts

The field layouts in Work items list can be defined at the following levels:

- Global - Work items list layouts can be specified for the entire application programmatically.
- Define Layouts at Workflow Level - The layout for a Workflow can be specified in the Process Designer while designing the Workflow.
- Define Layouts for Queues - The layout for work items assigned to the queue can be specified in the Queue Management window.
- User - Users can further personalize the work item list views and save layouts.

Global layouts are applied only to those workflows and queues that do not have Workflow/queue specific layouts defined and can be saved with a specific name.

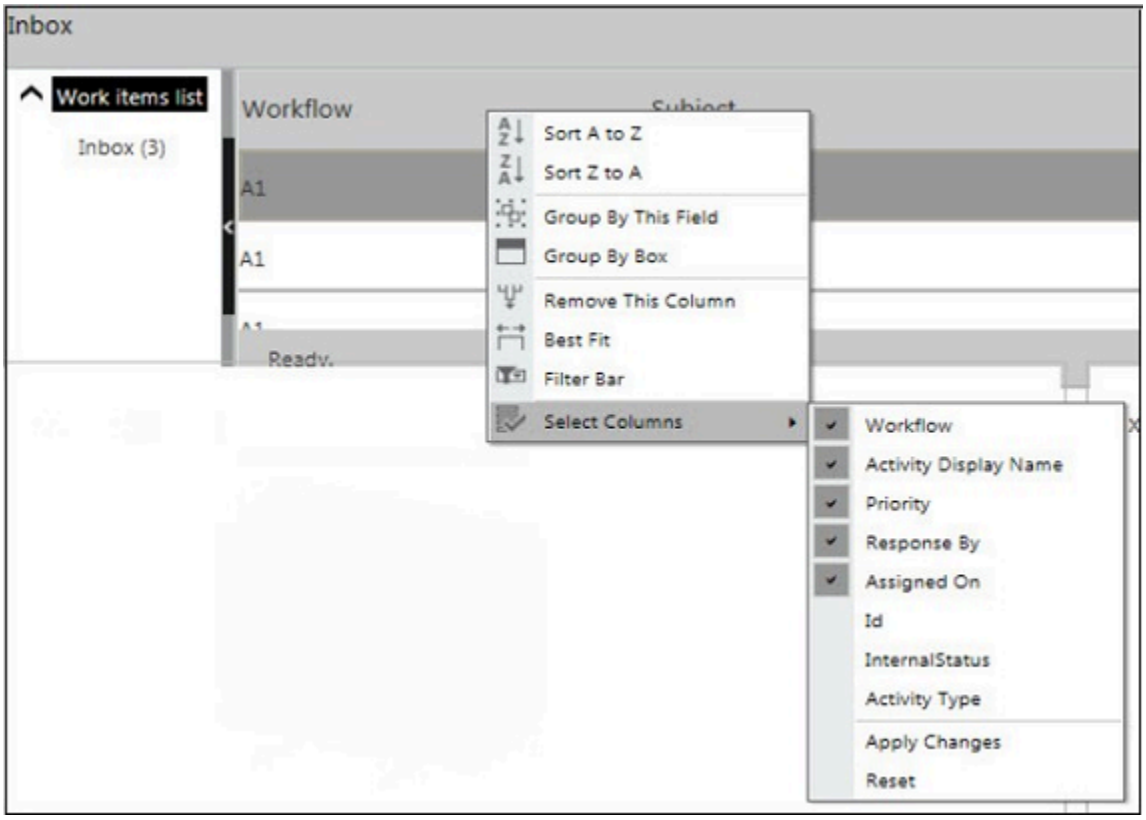
The Work items list also enables filtering of data (see Filtering Data in Work Items List) within a list.

Work Items List Layout Customization

The Work items list view provides list management features for sorting, grouping, column selection, filtering and rearranging. Right-clicking on any column in the column header bar brings up a menu with the available options.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

List Management Options



A description of the available options follows:

- **Sort A to Z** - Sorts the list on the selected column in ascending order
- **Sort Z to A** - Sorts the list on the selected column in descending order
- **Group By This Field** - Groups the list by the selected column
- **Group By Box** - Displays the Group By box at the top. You can drag and drop a column header in this box to group the list by that field. This is an alternative way to group the list.
- **Remove This Column** - Removes the selected column. The removed column can be restored by reselecting the column from the Select Columns option.
- **Best Fit** - Changes the column width to best fit
- **Filter Bar** - Toggles the filter bar from on and off
- **Select Columns** - Provides a selection list with all available column headers. Columns displayed in the view are checked. Click Apply Changes for the new settings to take effect. Click Reset to revert to the previous settings. You can also rearrange the columns by dragging and dropping them to a new position.

Define Layouts at Workflow Level

The layout for the Work items list can be defined for a Workflow in the **Process Designer** by selecting the Work item list Layout option from the **Settings** menu. Selecting this option brings up the **WorkList Layout Customization** window. The Worklist Layout can be customized for the following:

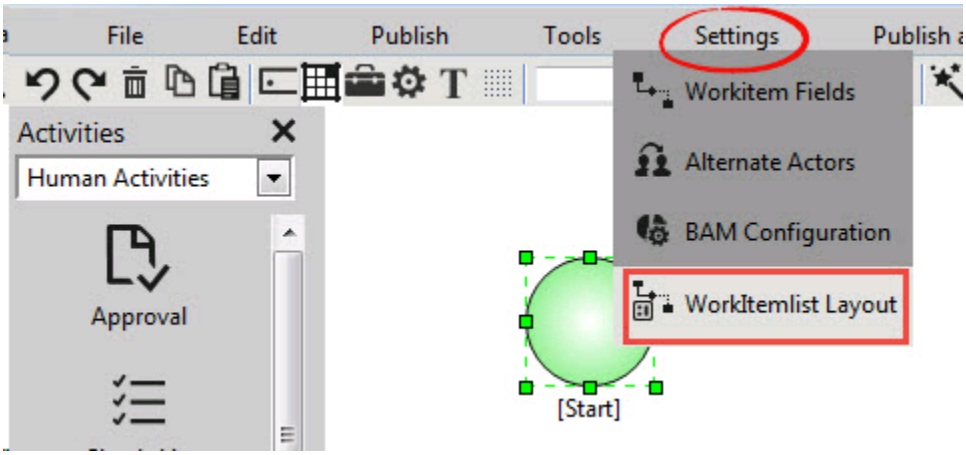
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Column positions
- Column selection
- Grouping a column
- Sorting a column

If you set the **ShowInboxViewonly** property to true, Workflow layouts will not be applied.

The default view of Work items list in the Enterprise Console cannot be customized. The above customization applies only to work item lists created by embedding the Work Item List Control in an ASPX page. For more information about using AVEVA Work Tasks Controls as a part of a project, see *Using Controls in Applications* in the *Developer Guide*.

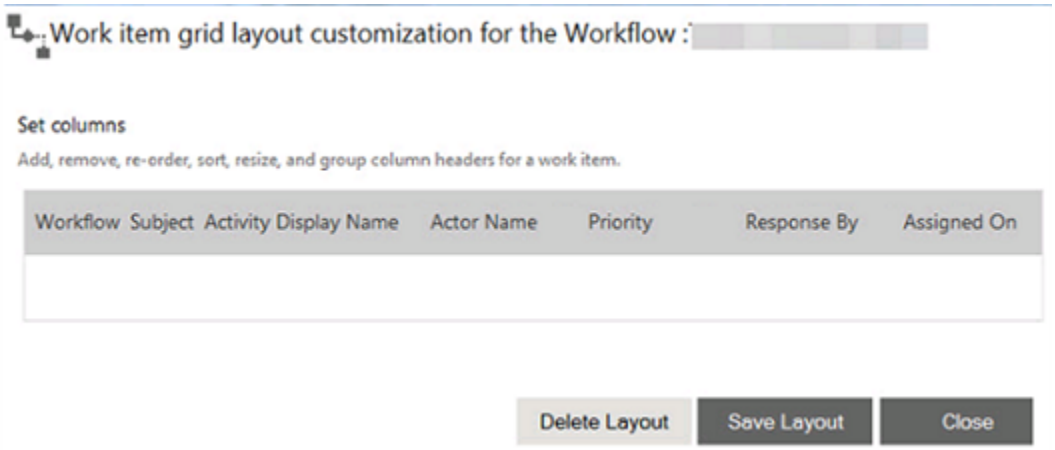
Work items list Layout Option in Process Designer



WorkList Layout Customization

The WorkList Layout Customization window allows you to customize the layout of Work items lists assigned from the Workflow. For more information on customization options, refer to Work items list Customization Options.

WorkList Layout Customization for Workflow Window



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Define Global Layouts

AVEVA Work Tasks enables Work items list layouts to be defined for the entire application. If the property GlobalLayout is set to true, the Work item list layout can be defined at the global level.

To add a Work items list layout using a panel

1. Drag and drop a panel.
2. In the underlying code, create a new instance of control and add the instance to the panel as in the following example:

```
Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject("Telematics");//Repository name
object loggedinUserId = "2d52b2c0-4cf8-4af6-acad-a0c9908eeab5";
Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext(loggedinUserId,
application, "skeltalist");
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl WIL = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
WIL.ApplicationName = uContext.Repository.ApplicationName;
WIL.UserIdString = uContext.LoggedInUserId;
WIL.ID = "WorkItemListControl";
WIL.DisplayGroupUI = true;
WIL.Height = Unit.Percentage(100);
WIL.ReadingPaneInPopupWindow = true;
Panel1.Controls.Add(WIL);//add control to the panel
```

3. To load all the AVEVA Work Tasks controls separately on aspx page, set the user context first.For example:

```
Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext(loggedinUserId,
application, "skeltalist")
In the underlying code, create a new instance of control and add the instance to the
panel in the OnInit as shown in the following example:
Skelta.Core.ApplicationObject application = new
Skelta.Core.ApplicationObject("Telematics");//Repository name
object loggedinUserId = "2d52b2c0-4cf8-4af6-acad-a0c9908eeab5";
Skelta.Entity.UserContext uContext = new Skelta.Entity.UserContext(loggedinUserId,
application, "skeltalist");
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl WIL = new
Skelta.HWS.WorkListChannel.Web.WorkItemListControl.WorkItemControl();
WIL.ApplicationName = uContext.Repository.ApplicationName;
WIL.UserIdString = uContext.LoggedInUserId;
WIL.ID = "WorkItemListControl";
WIL.DisplayGroupUI = true;
WIL.Height = Unit.Percentage(100);
WIL.ReadingPaneInPopupWindow = true;
Panel1.Controls.Add(WIL);//add control to the panel
```

Note: The default view of the Work items list in the **Enterprise Console** can be customized. For more information about using **Controls** as a part of a project, see *Using Controls in Applications* in the *Developer Guide*.

Define Layouts for Queues

The layout for the Work items list for items assigned to a queue can be defined in the **Settings** tab of the **Queue Management** window. The Worklist Layout can be customized for the following:

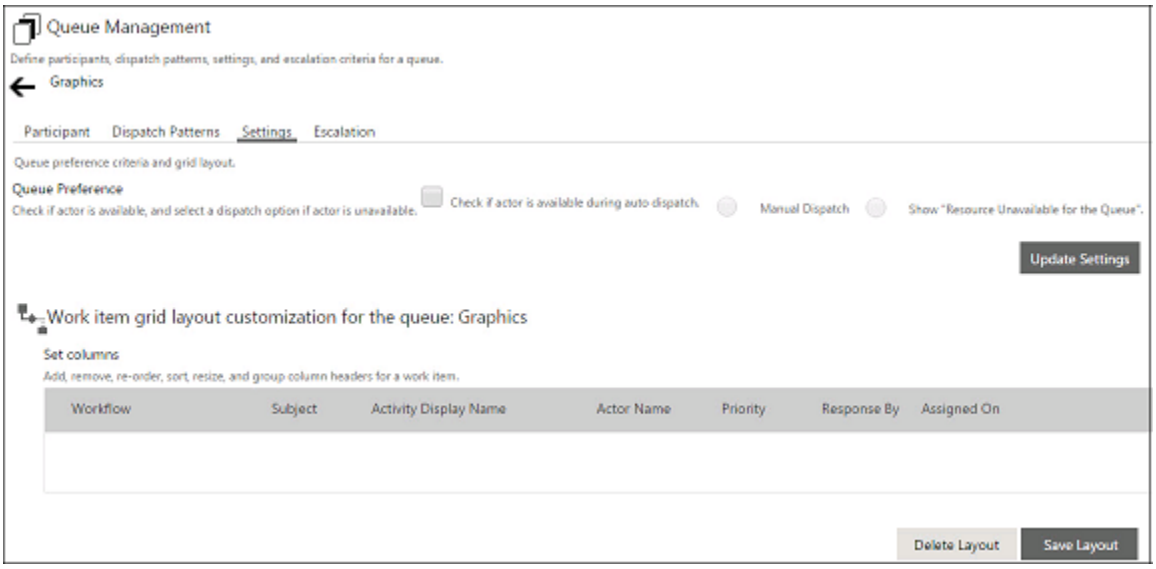
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Column positions
- Column selection
- Grouping a column
- Sorting a column

For more information on customization options, see Work Items List Layout Customization.

Note: The default view of the Work items list in the Enterprise Console cannot be customized. The above customization applies only to Work items lists created by embedding the Work Item List Control in an ASPX page. For more information about using Controls as a part of a project, see *Using Controls in Applications* in the *Developer Guide*.

WorkList Layout Customization for the Queue



Define Work Items List Layouts for a User

Individual users can perform the following activities to customize Work items list layouts:

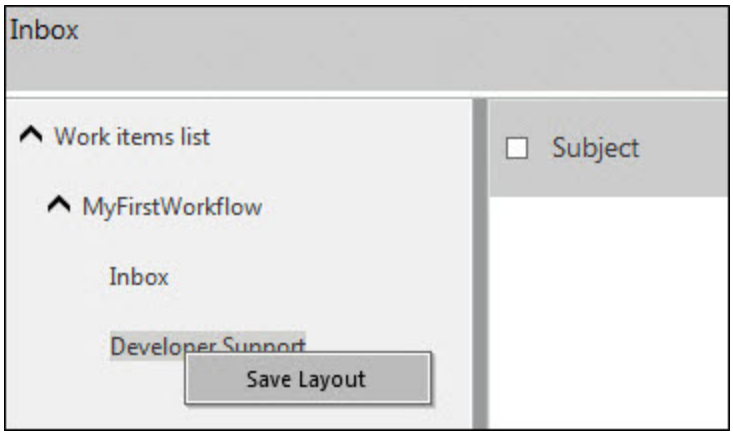
- Customize layout (For more information about customization options, see Work Items List Layout Customization.)
- Save a customized layout
- Delete a customized layout

To save Customized Layouts

Customized views of a Work items list can be saved under a personalized name. To save the custom settings:

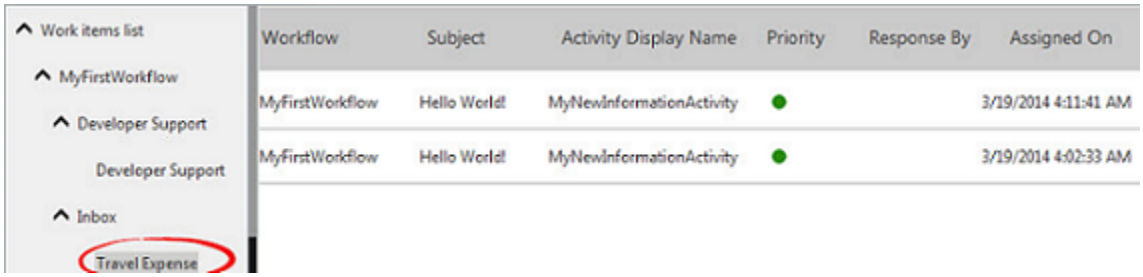
1. Right-click the folder name that contains the customized view. For example, if you have customized the Work items list for Inbox, right-click **Inbox**.
2. Select **Save Layout**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Enter the name of the new layout at the **Enter the view name** prompt. The name of the saved layout appears in the folder list under the folder.

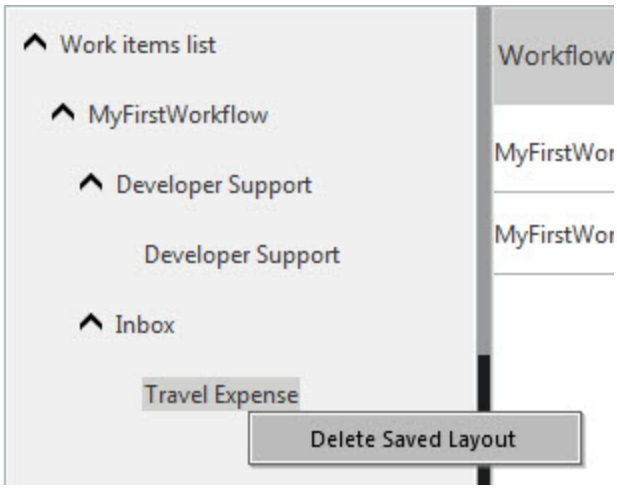
4. The new folder displays the same list of items as the default folder that was customized. The work items will also be displayed in the default folder. For example, in the following figure, Inbox has been customized and the layout saved as **Travel Expense**.



To delete Customized Layouts

- Right-click the folder name for the layout, and then select **Delete saved Layout**. The layout is deleted.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Customize Work Items List Views

The Work items list view and interface can be customized by setting appropriate properties for the Work items list control. The properties that enable customized views are as follows:

- **DisplayLeftTreeMenuAtApplicationLevel**: This property enables all work items to be grouped by the parent application instead of the default grouping by Workflow.
- **WorkflowNames**: This property enables you to list only work items assigned by the specified workflows.
- **ShowQueueNames**: This property enables you to display work item folders for only the specified queues.
- **ShowInboxViewOnly**: This property enables you to display only the Work items list for the Inbox folder without the view of the folder structure.
- **ShowManagerViewOnly**: This property enables you to display only the Work items list for the Manager View.

The default view of the Work items list in the Enterprise Console cannot be customized. The above customization applies only to work item lists created by embedding the Work items list control in an ASPX page.

Display Work Items List by Application

The Work items list property **DisplayLeftTreeMenuAtApplicationLevel** enables all work items to be grouped by the parent application. If this property is set to false, then the default view of work items grouped by workflow applies.

Work items list Grouped by Application

DisplayLeftTreeMenuAtApplicationLevel = **True**

Display Work Items List Grouped by Workflows

The Work items list property **DisplayLeftTreeMenuAtApplicationLevel** enables all work items to be grouped by the parent application. If this property is set to false, then the default view of work items grouped by workflow applies.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Work items list grouped by Workflows

DisplayLeftTreeMenuAtApplicationLevel = **False**

Display Work Items List for Specified Workflows

The property WorkflowNames enables you to display work items assigned by specified workflows. For example, if work items are assigned by workflows for Invoice Processing and Expense claims, you can create two controls for Work items list with the first set to display only items assigned from Invoice Processing and the other set to display only Expense work items.

Note: This property is effective only when the DisplayLeftTreeMenuAtApplicationLevel property is set to False.

Display Work Items List for Specified Queues

The property **ShowQueueNames** enables you to display only specified queue folders. For example, if work items are assigned to Finance queue and Purchase Department queue, you can create two controls for Work items list with the first set to display only the Finance queue folder and the other set to display only the Purchase Department queue folder.

The display of queue folder is subject to the participants right to View Queue Items for the queue. For more information on participant rights, see Participants Tab.

Display Inbox View

The **ShowInboxViewOnly** property enables you to display only the Work items list for the Inbox folder without the view of the folder structure. If this property is set to true, users will not be able to save customized layouts or view work items in queue folders.

Display Manager's View

The **ShowManagerViewOnly** property enables you to display only the Work items list for the **Manager View** folder without the view of the folder structure. If this property is set to true, users will not be able to save customized layouts, view work items in queue folders or view other Inbox items. This property is useful to create a separate control for the manager's view.

Preferences for Work Item Management

When the volume of work items is high, it is essential to minimize the processing time as much as possible. **AVEVA Work Tasks** provides properties at the control level to set preferences for work item management. These properties enable you to expedite processing by defining how the work items can be managed and acted upon.

The two properties for work item management are as follows:

- Auto-ownership on Row Click: This property enables you to allow users to take ownership of a work item by clicking on the item in the Work items list. If this property is set to true, the option Take Ownership in the Manage Work Items field in **Work Item Details** will not be available.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Enable Multi-Select Option in Work Item List: This property provides users with an option to select multiple work items for processing in a batch. For example, selecting multiple work items for approval.

  Note: The default view of the Work items list in the Enterprise Console cannot be customized. The above customization applies only to the work items lists are created by embedding the Work Item List Control in an ASPX page. For more information about using Controls as a part of a project, see *Using Controls in Applications* in the *Developer Guide*.

Auto-ownership on Row Click

The property **AllowAutoOwnershipOnRowClick** enables you to allow users to take the ownership of a work item by clicking on the item in the Work items list. If this property is set to true, the option **Take Ownership** in the **Manage Work Items** field in Work Item Details will not be available.

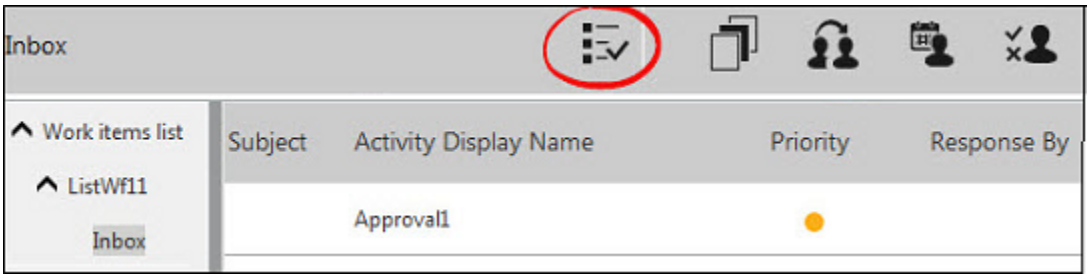
Setting this property to true is useful in scenarios that require strict adherence to performance indicators, which measure the time when a work item is assigned to the time when a user takes ownership of it.

Enable Multi-Select Option in Work Item List

When the **DisplayGroupUI** property is enabled, an additional button **Enable Multi Select** is displayed in the Work items list. This property is a powerful feature that expedites processing of similar items in a batch.

The **Enable Multi Select** button provides users with options of acting on items as a batch. An additional option is made available when the items are grouped.

Enable Multi Select Option in Work items list



Note: The Multi Select option is not supported for **Resource Set** activity and **Information** activity.

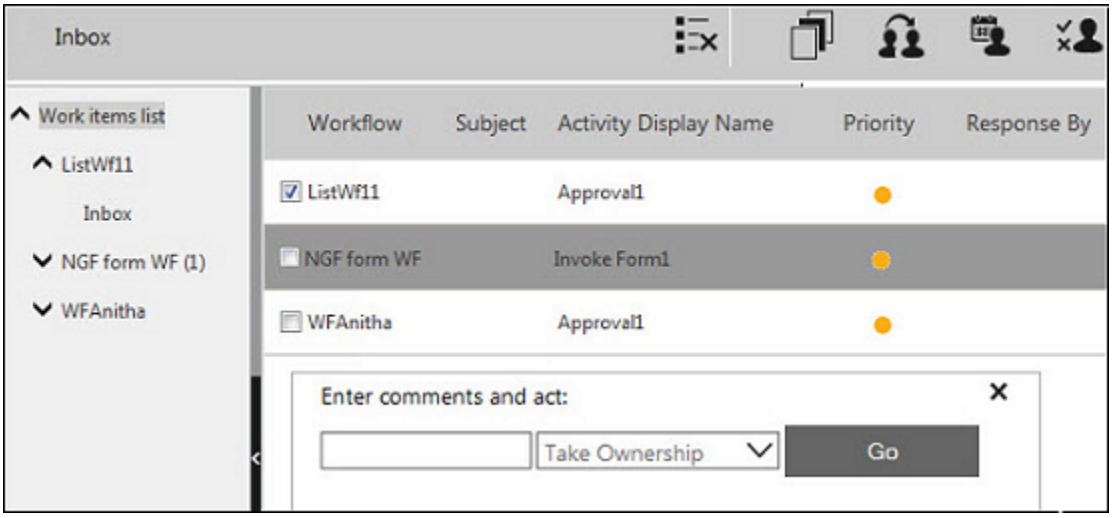
Work Item List in Multi Select Mode

On clicking the **Enable Multi Select** button, the Work items list interface is displayed in the multi-select mode with check boxes for each item. The user can enter comments that are applied to all the selections. The options for acting upon the selection depend on the type and stage of processing that the selected items are in.

Example 1: A user has multiple new work items that have the subject Approval of Vendor Invoices. The user can enable multi-select, then select all the items that they want to take the ownership of. They can then select the **Take Ownership** option in the drop-down to take the ownership of the selected items in a single click.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Work Item List with Multi-Select Enabled



Selecting Similar Work Items from the Work Items List

You can select all the similar work items in the **Work items list** based on the activity type of the selected work item. For example, if you select the work item with **Approval** activity, then all the other work items with the Approval activity will be selected.

To select similar work items from the Work items list

1. Select the work item and then right-click to view the context menu.
2. Click **Select all similar items** option in the context menu.

   Based on the activity type of the selected work item, all the similar items in the Work items list get selected.

To deselect the work items, click **UnSelect all similar items** option in the context menu.

To select similar work items from the grouped Work items list

1. Right-click on the column and then click **Group By This Field** to group the work items based on the selected column.
2. Select the work item from the grouped list and then right-click to view the context menu.
3. Click **Select all within group** option in the context menu.

   Based on the activity type of the selected work item, all the similar work items listed in the group get selected. If the group has items with different activity, an error message is displayed.

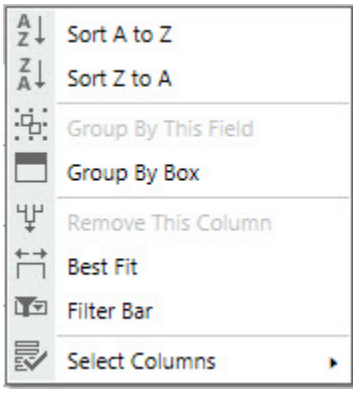
To deselect the work items within the group, click **UnSelect all within group** option in the context menu.

Filter Data in Work Items List

The **Filter Bar** in the Work items list is displayed when the Filter Bar option is selected from the right-click menu for the columns. This option allows users to filter the list for specific data.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

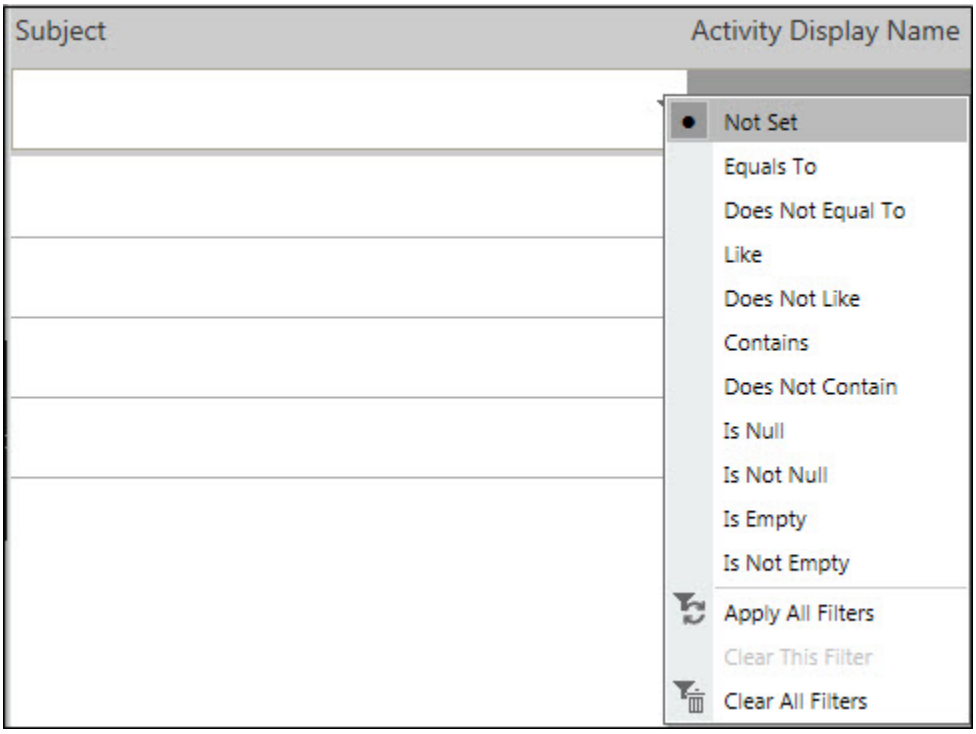
Toggle Filter Bar Option



To set filters

1. Enter the filter criteria for the column in the filter bar.
2. Click the filter icon to select the appropriate condition from the filter menu.

Filter Menu



When you have finished setting all the required conditions for the various columns, click the filter icon and select **Apply All Filters**.

To clear a particular filter

1. Click the filter icon on the column whose filter needs to be cleared.
2. Select **Clear This Filter**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To clear all filters

1. Click the filter icon on any column.
2. Select **Clear All Filters**.

Manage Work Items

Manage Work Items in AVEVA Work Tasks by the following ways:

- Forward: Forward work items among different security groups.
- Abort: Abort or kill work items, as long as the user has permission to abort the work item.
- Remove: Remove a workflow item without aborting the workflow instance.

Forward Work Items

In AVEVA Work Tasks, you can view the work items in two different modes.

- **Normal View**: In the normal view, you can see all the assigned work items.
- **Manager View**: In the manager view, you can see all the assigned work items in addition to all the work items assigned to the users reporting to the reportees.

In both these views, you have the provision to forward the work items.

Apart from **Inbox**, the following users should be able to view and forward the work items from the **Manage Work items** in the **Enterprise Console**:

- Owner of the workflows
- Users who are not owners of the workflows, but part of the default Administrator security group.

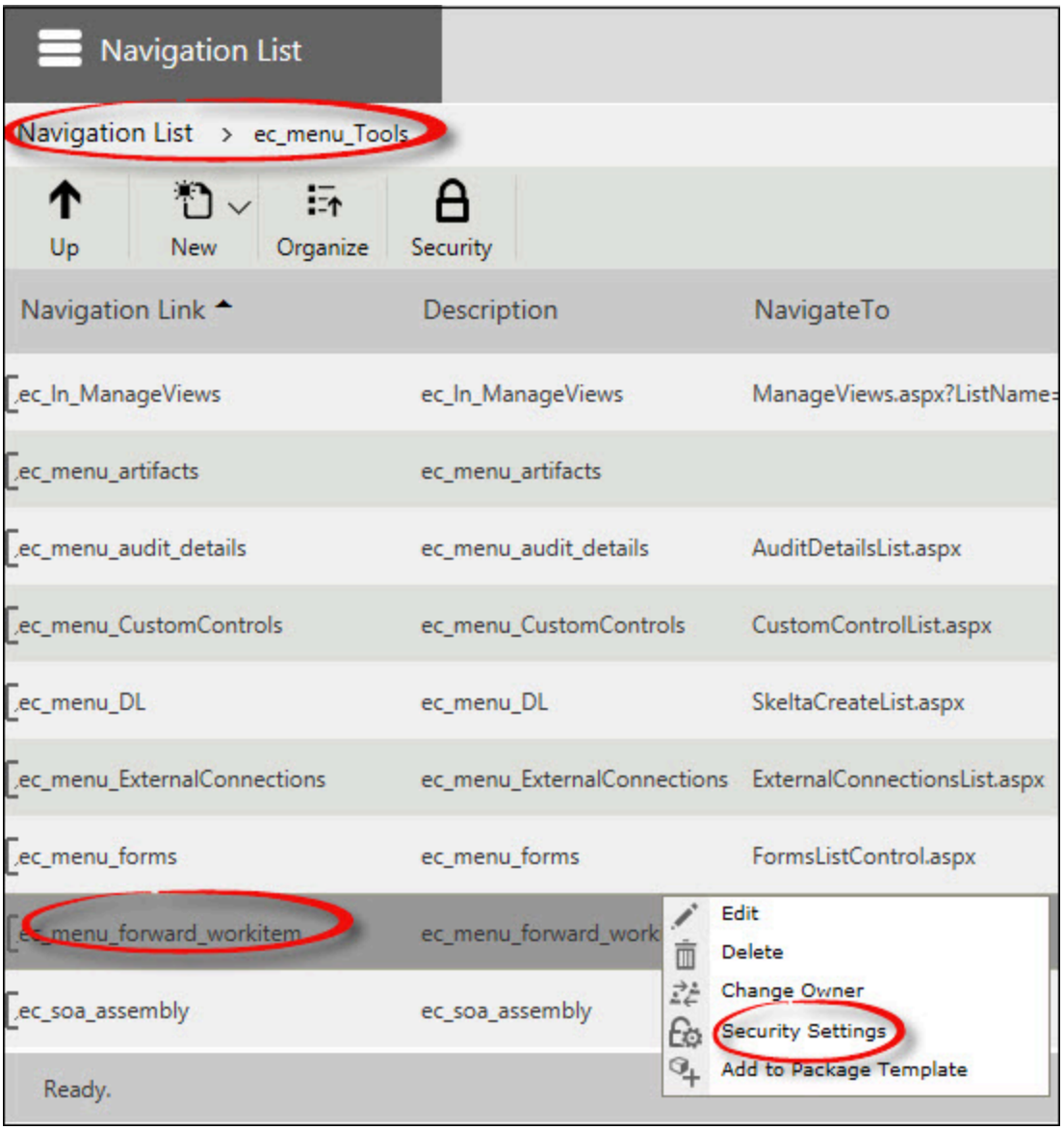
To open Manage Work Items

1. In the **Enterprise Console**, click the **Menu** icon, click **Tools**, and then click **Manage Work items**.
2. By default repository configuration, only the users who belong to the Administrator security group will be able to see this menu item.

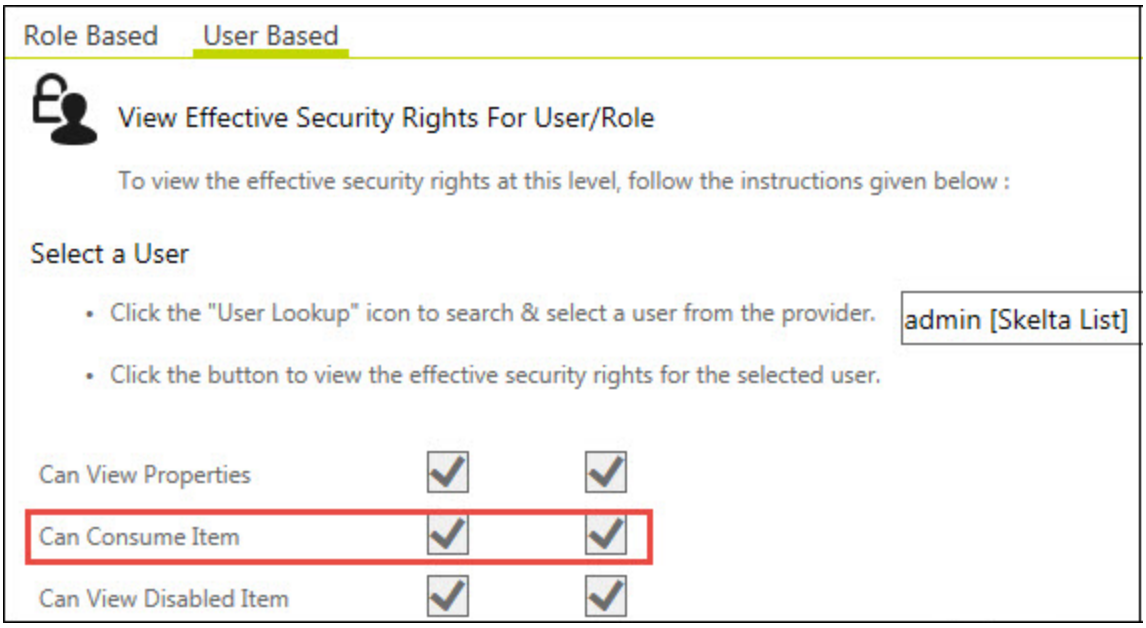
To View 'Manage Work items' in the Enterprise Console

To view the **Manage Work items** option in the **Enterprise Console** menu, you must have the **Can Consume Item** right for **Manage Work item** list item (**ec_menu_forward_work item**) in the **Manage Navigation List** of repository settings.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



For details about the Manage Navigation List, see Manage Navigation List.

Forward Work Items

To forward work items, you must have the **Can Forward Work Item** right. For example, if the user belongs to the security group **Administrator**, the **Can Forward Work Item** right should be enabled for the Administrator security group.

To forward work items to other users

1. In the **Enterprise Console**, click the **Menu** icon, click **Tools**, and then click **Manage Work items**. The **Manage Work items** page appears.
   In the **Manage Work items** page, you can see the work items of all the workflows owned by you.
2. Select the required work item(s) to forward to a different user.
3. Click **Forward**. Alternatively, right-click the work item and then click **Forward**. The **Select a user and forward the work item(s)** dialog box appears.
4. Click the **User Lookup** icon to select a user. The **User Lookup** dialog box appears. Select only one user at a time to forward work item(s).
5. Search and enter the user name.
6. Click **Update**.
7. Click **Forward**. The selected work item gets assigned to the selected user and the information is updated in the grid.

Kill Work Items

To kill or abort work items, the user should have the **Can Kill Work Item** right. For example, if the user belongs to the security group **Administrator**, the **Can Kill Work Item** right should be enabled for the Administrator security group.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Abort Work Items

You can abort or kill work items in the Manage Work items window. To abort work items, the logged in user should have the permission to abort the work items. For this, the **Can Kill Work Item** right should be checked.

To abort a work item

1. In the **Enterprise Console**, click the **Menu** icon, click **Tools**, and then click **Manage Work items**. The **Manage Work items** page appears.
   All work items appear in a grid showing the information about work items assigned to different users within the application.
   The **Manage Work items** is visible only to users who have the right to forward work items.
2. Select the work item(s) that you want to abort. You can select multiple work items at a time by selecting the check box in the grid.
3. Click **Abort**. Alternatively, right-click the work item, and then select Abort. A confirmation message to abort the work item appears.
4. Click **Yes** to confirm the abort, or **No** to cancel the abort. The message about successful abort appears.
5. Click **OK**.

Remove Work Items

You can remove the work item of a workflow instance without aborting the workflow instance. To remove work items, the logged in user must have the **Can Kill Work Item** security rights.

The **Manage Work items** option is visible only to users who have the right to remove or forward work items.

The work item does not get removed under the following conditions:

- If the work item to be removed is the only work item of the activity
- If the ownership of the work item is taken by the logged in user

To remove a work item

1. In the **Enterprise Console**, click the **Menu** icon, click **Tools**, and then click **Manage Work items**. The **Manage Work items** page appears.
2. Select the workflow instance that you want to remove. You can select multiple work items at a time by selecting the check box in the grid.
3. Click **Remove**. Alternatively, right-click the work item, and then select **Remove**.
4. Click **Yes** to remove the work item.
5. Click **OK**.

Alternate Actors

An actor is a user who is responsible for acting on a task or an activity. An alternate actor is another user who can be given the required rights to act on the task or activity on behalf of the actor.

You can set an alternate actor in the **Alternate Actor** menu item in the **Settings** tab of the **Enterprise Console**.

To set an alternate actor, refer to the section Alternate Actors.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Alternate Actors

An actor is a user who is responsible for acting on a task or an activity. An alternate actor is another user who can be given the required rights to act on the task or activity on behalf of the actor.

You can set an alternate actor for an user in one of the following ways:

- Use the **Alternate Actor** menu item in the **Settings** tab of the **Enterprise Console**. This **Alternate Actor** menu item is visible by default for *Administrator* and *Contributor* security groups. For Custom or customized Security Group, set the permissions **Can Design Workflow** and **Can Set Alternate Actor** for that group.
- Assign alternate actors to a workflow. This alternate actor will have rights to approve or reject an activity in the absence of the original actor. For more information about how to set an alternate actor in the Process Designer, see Steps to set an Alternate Actor in the Process Designer.
- Set an alternate actor in the **Inbox**.

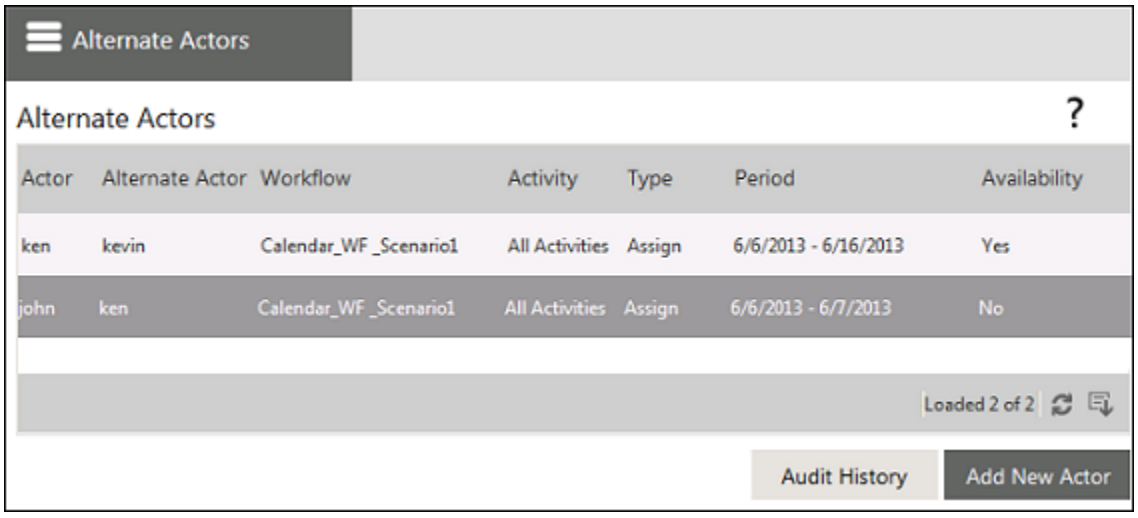
To set an Alternate Actor in the Enterprise Console

1. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Alternate Actors**.

   The Alternate Actors page appears.

   This page lists the alternate actors set for the particular repository.

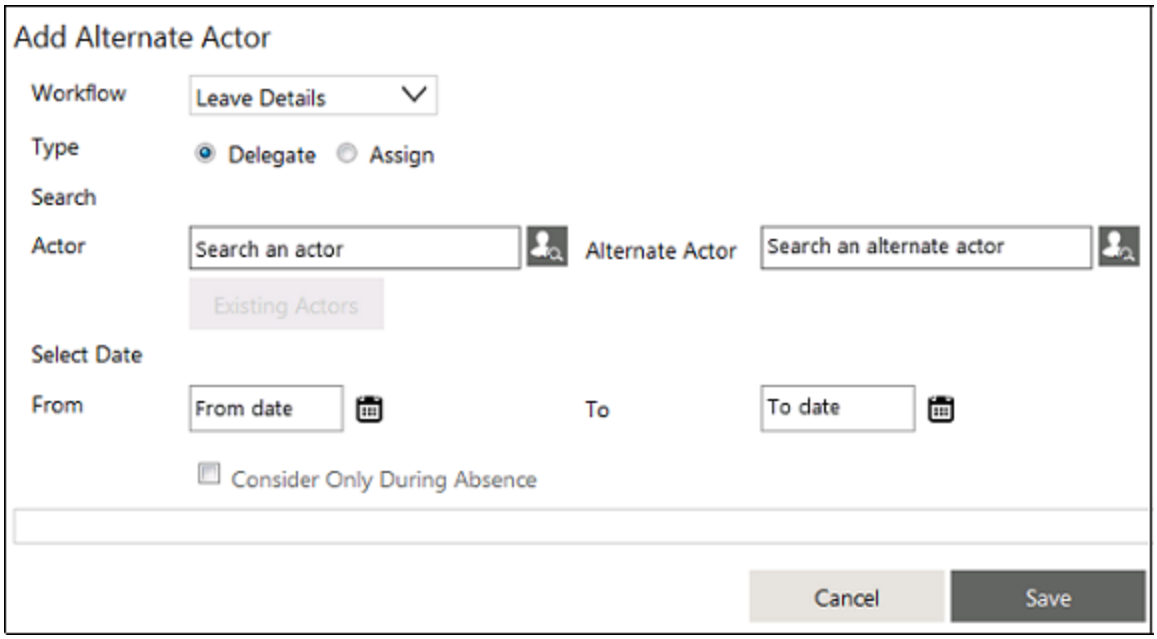
   The following image shows the Alternate Actor page:



2. Click **Add New Actor** to add a new alternate actor.

   The Add Alternate Actor dialog box appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Enter the required inputs for each work item field:
   a. **Workflow**: Specify the workflow for which the alternate actors are being added.
   b. **Activities**: Specify the activity for which the alternate actors are being added. The field will be displayed to the actor with this label name in the Work items list.
   c. **Type**: Specify the type of activity transfer from the original actor to the alternate actor.
      a. **Delegate**: Delegate the task to the alternate actor.
      b. **Assign**: Shift the task assignment to the alternate actor.

      Only one type of setting (**Delegate** or **Assign**) is allowed for an Actor, Alternate Actor, and Workflow combination.

      In an upgraded environment where **Delegate** and **Assign** is already set for an actor, **Delegate** takes precedence over **Assign**.

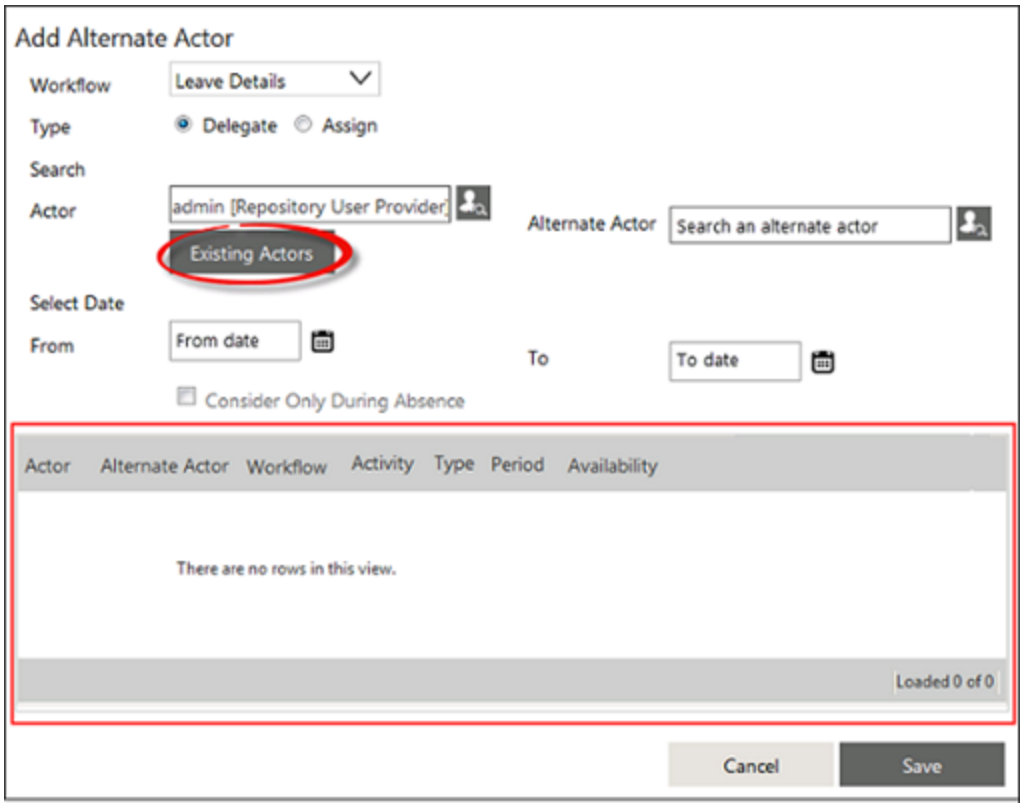
      Note: At runtime, only a single level of delegation or assignment occurs at any given point of time. For example, consider an actor *Jim.* An alternate actor, *John* is configured with delegation type as *Delegate*. Again, an alternate actor, *Kevin* is configured considering *John* as an actor with delegate type as *Assign*. Then, the task goes to *John* from *Jim*.
   d. **Search**: Specify the actor and the alternate actor.
      i. Click the **User Lookup** button to select the **Actor** or the **Alternate Actor.**

      The User Lookup dialog box appears.
      ii. Select the required users, and click **Update**.

      The same user cannot be set for both Actor and Alternate Actor.

      Click **Existing Actors** to view the list of all the existing alternate actors of the selected actor. The list of existing actors and the associated details get populated in the grid at the bottom section of the dialog box as shown in the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



e. **Select Date**: Specify the period for which the alternate actor is being assigned to the original actor.

   i. **From**: Click the calendar to select the start date.

   ii. **To**: Click the calendar to select the end date.

   iii. **Consider Only During Absence**: Select to specify that the alternate actor is to be considered only during the absence of the original actor.

   Note that the **From** date cannot be older than the current date.

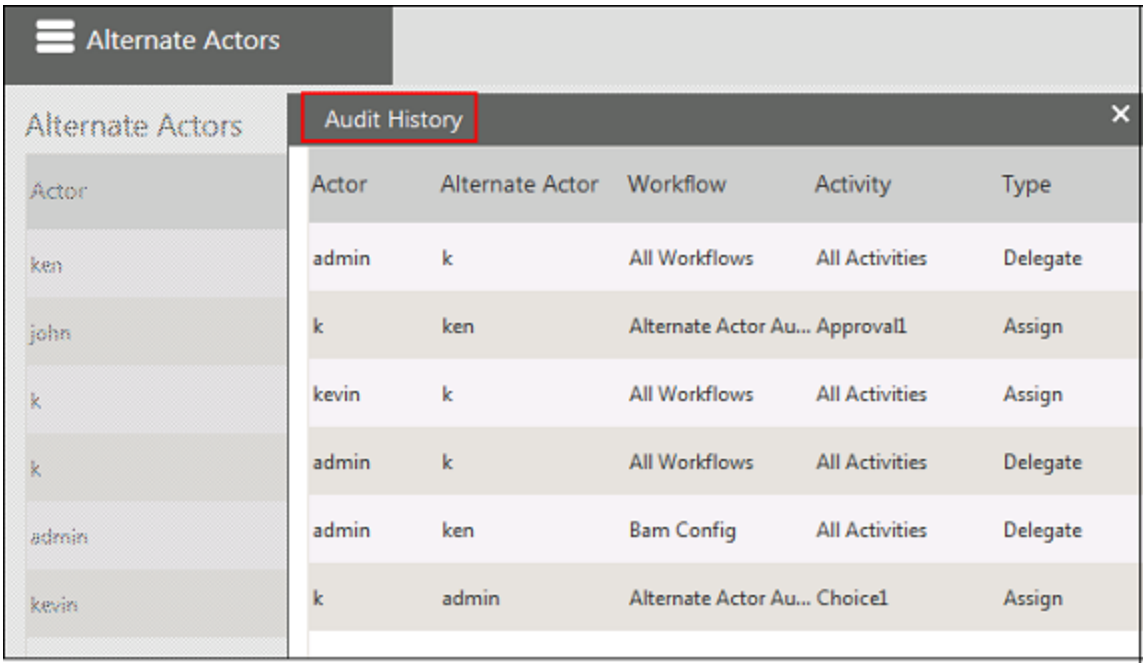
4. Click **Save** to add the new alternate actor.

   The **Alternate Actors** window opens showing the newly added alternate actor.

Audit History

You can view the complete change history for an actor, when the alternate actor settings were added or deleted.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



- Click **Audit History**.

  The Audit History dialog box appears.

  The Audit History dialog box contains the complete change history when an alternate actor was added or deleted. You cannot edit the audit history.

By default, the audit history is enabled. To disable the audit history, do the following:

1. Go to **SKRepository** table in the **Farm Database.**
2. In the **Properties Column**, set the property key **isDeletedListItemsTraceEnabled** to *False*.

   After upgrading to the required version of AVEVA Work Tasks, if the **Audit History** button is not visible, then set the property key **isDeletedListItemsTraceEnabled** to *True*.

   For more information about enabling and disabling the audit history, see Enable or Disable Deleted List Items feature for a Repository.

See Also: To set the security rights for showing an alternate actor, see Secure Work Item Control - Show Alternate Actor.

Configuration Scenarios

Consider the following:

Actor = Jim

Alternate Actors = Richard and John

The following configuration scenarios for alternate actor are possible:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scenario 1: For all workflows and all activities of all workflows with delegation type as *Delegate*

In this scenario, either one of the following combinations are possible.

When **Option 1** is set for an alternate actor, **Option 2(a)** and **Option 2(b)** cannot be set. When **Option 1** is not set, both **Option 2(a)** and **Option 2(b)** can be set to co-exist provided the **From** date and **To** date in both **Option 2(a)** and **Option 2(b)** does not *overlap*.

- **Option 1**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard*. The delegation type is set as **Delegate**. The **From** date and **To** date is empty provided the **Date Configuration Flag for Alternate Actors** is set as false in the **Central Configuration** site for the repository. The **Consider Only During Absence** is set as *True* or *False*.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	Richard	-	-	Yes/No

- **Option 2 (a)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard.* The delegation type is set as **Delegate**. The **From** date and **To** date are set. **Consider Only During Absence** is set as *True* or *False*.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	Richard	13/01/23	15/01/20	Yes/No

- **Option 2 (b)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *John.* The delegation type is set as **Delegate**. The **From** date and **To** date are set. **Consider Only During Absence** is set as *True* or *False*. The alternate actor in this option is different from the alternate actor set in **Option 2(a)**. The **From** date and **To** date in this option does not overlap with the **From** date and **To** date of **Option 2 (a)**.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	John	16/01/23	21/01/23	Yes/No

Scenario 2: For all workflows and all activities of all workflows with delegation type as *Assign*

In this scenario, either one of the following combinations are possible.

When **Option 1** is set for an alternate actor, **Option 2(a)** and **Option 2(b)** cannot be set. When **Option 1** is not set, both **Option 2(a)** and **Option 2(b)** can be set to co-exist provided the **From** date and **To** date in both **Option 2(a)** and **Option 2(b)** does not *overlap*.

- **Option 1**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard*. The delegation type is set as **Assign**. The **From** date and **To** date is empty provided the **Date Configuration Flag for Alternate Actors** is set as false in

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the **Central Configuration** site for the repository. The **Consider Only During Absence** is set as *True* or *False*.

The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	Richard	-	-	Yes/No

- **Option 2 (a)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard.* The delegation type is set as **Assign**. The **From** date and **To** date are set. **Consider Only During Absence** is set as *True* or *False*.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	Richard	13/01/23	15/01/23	Yes/No

- **Option 2 (b)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *John.* The delegation type is set as **Assign**. The alternate actor in this option is different from the alternate actor set in **Option 2(a).** The **From** date and **To** date are set. **Consider Only During Absence** is set as *True* or *False*. The **From** date and **To** date in this option does not overlap with the **From** date and **To** date of Option 2 (a).

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	John	16/01/23	21/01/23	Yes/No

Scenario 3: For a particular workflow and all activities of the workflow with delegation type as *Delegate*

In this scenario, either one of the following combinations are possible.

When **Option 1** is set for an alternate actor, **Option 2(a)** and **Option 2(b)** cannot be set. When **Option 1** is not set, both **Option 2(a)** and **Option 2(b)** can be set to co-exist provided the **From** date and **To** date in both **Option 2(a)** and **Option 2(b)** does not *overlap*.

- **Option 1**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard*. The delegation type is set as **Delegate**. The **From** date and **To** date is empty provided the **Date Configuration Flag for Alternate Actors** is set as false in the **Central Configuration** site for the repository. The **Consider Only During Absence** is set as *True* or *False*.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	Richard	-	-	Yes/No

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Option 2 (a)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard.* The delegation type is set as **Delegate**. The **From** date and **To** date are set. **Consider Only During Absence** is set as *True* or *False*.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	John	16/01/23	21/01/23	Yes/No

- **Option 2 (b)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *John.* The delegation type is set as **Delegate**. The alternate actor in this option is different from the alternate actor set in **Option 2(a).** The **From** date and **To** date are set. **Consider Only During Absence** is set as *True* or *False*. The **From** date and **To** date in this option does not overlap with the **From** date and **To** date of **Option 2 (a)**.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	John	16/01/23	21/01/23	Yes/No

Scenario 4: For a particular workflow and all activities of the workflow with delegation type as *Assign*

In this scenario, either one of the following combinations are possible.

When **Option 1** is set for an alternate actor, **Option 2(a)** and **Option 2(b)** cannot be set. When **Option 1** is not set, both **Option 2(a)** and **Option 2(b)** can be set to co-exist provided the **From** date and **To** date in both **Option 2(a)** and **Option 2(b)** does not *overlap*.

- **Option 1**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard*. The delegation type is set as **Assign**. The **From** date and **To** date is empty provided the **Date Configuration Flag for Alternate Actors** is set as false in the **Central Configuration** site for the repository. The **Consider Only During Absence** is set as *True* or *False*.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	Richard	-	-	Yes/No

- **Option 2 (a)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard.* The delegation type is set as **Assign**. The **From** date and **To** date are set. **Consider Only During Absence** is set as *True* or *False*.

  The following table depicts the scenario:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	Richard	13/01/23	15/01/23	Yes/No

- **Option 2 (b)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *John.* The delegation type is set as **Assign**. The alternate actor in this option is different from the alternate actor set in **Option 2(a).** The **From** date and **To** date are set. **Consider Only During Absence** is set as *True* or *False*. The **From** date and **To** date in this option does not overlap with the **From** date and **To** date of **Option 2 (a)**.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	John	16/01/23	21/01/23	Yes/No

Scenario 5: For a particular workflow and a particular activity of the workflow with delegation type as *Delegate*

In this scenario, either one of the following combinations are possible.

When **Option 1** is set for an alternate actor, **Option 2(a)** and **Option 2(b)** cannot be set. When **Option 1** is not set, both **Option 2(a)** and **Option 2(b)** can be set to co-exist provided the **From** date and **To** date in both **Option 2(a)** and **Option 2(b)** does not *overlap*.

- **Option 1**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard*. The delegation type is set as **Delegate**. The **From** date and **To** date is empty provided the **Date Configuration Flag for Alternate Actors** is set as false in the **Central Configuration** site for the repository. The **Consider Only During Absence** is set as *True* or *False*.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	Richard	-	-	Yes/No

- **Option 2 (a)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard.* The delegation type is set as **Delegate**. The **From** date and **To** date are set. **Consider Only During Absence** is set as *True* or *False*.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	Richard	13/01/23	15/01/23	Yes/No

- **Option 2 (b)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *John.* The delegation type is set as **Delegate**. The alternate actor in this option is different from the alternate actor set in **Option 2(a).** The **From** date and

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To date are set. **Consider Only During Absence** is set as *True* or *False*. The **From** date and **To** date in this option does not overlap with the **From** date and **To** date of **Option 2 (a)**.

The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	John	16/01/23	21/01/23	Yes/No

Scenario 6: For a particular workflow and all activities of the workflow with delegation type as *Assign*

In this scenario, either one of the following combinations are possible.

When **Option 1** is set for an alternate actor, **Option 2(a)** and **Option 2(b)** cannot be set. When **Option 1** is not set, both **Option 2(a)** and **Option 2(b)** can be set to co-exist provided the **From** date and **To** date in both **Option 2(a)** and **Option 2(b)** does not *overlap*.

- **Option 1**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard*. The delegation type is set as **Assign**. The **From** date and **To** date is empty provided the **Date Configuration Flag for Alternate Actors** is set as false in the **Central Configuration** site for the repository. The **Consider Only During Absence** is set as *True* or *False*.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	Richard	-	-	Yes/No

- **Option 2 (a)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *Richard.* The delegation type is set as **Assign**. The **From** date and **To** date are set. **Consider Only During Absence** is set as *True* or *False*.

  The following table depicts the scenario:

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	Richard	13/01/23	15/01/23	Yes/No

- **Option 2 (b)**: Here the **Actor** is *Jim* and the **Alternate Actor** is *John.* The delegation type is set as **Assign**. The alternate actor in this option is different from the alternate actor set in **Option 2(a).** The **From** date and **To** date are set. **Consider Only During Absence** is set as *True* or *False*. The **From** date and **To** date in this option does not overlap with the **From** date and **To** date of **Option 2 (a)**.

  The following table depicts the scenario:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Actor	Alternate Actor	From Date	To Date	Consider only During Absence
Jim	John	16/01/23	21/01/23	Yes/No

Order of Precedence

When you set an alternate actor for an actor, the order of precedence is as follows:

1. A workflow with a specific activity
2. A workflow with all activities
3. All workflows with all activities

Business Activity Monitoring Alert

Business Activity Monitoring (BAM) Alert provides information regarding the process status, activity status, and gives more access to the status information or exceptions through email. BAM alert enables the events for critical data change in the process against the threshold value (maximum or minimum) set by the user. Thus BAM alert notifies the user about the current status of the process and enables the user to control it.

BAM features

- **Alerts for Business Activity Thresholds**
  This feature enables to configure an alert for Business Activity threshold and sends KPI/Overdue alerts as emails with embedded graphs. See Alerts for Business Activity Thresholds for more details.
- **BAM Events Associations**
  This feature provides the user to associate the workflow with a new BAM event based on settings made on Overdue Analysis or KPIs. These settings will be executed when the BAM Report is being processed. Decisions can be made effectively by using this. See BAM Events Associations for more details.
- **BAM Workflow**
  This feature allows the user to create a BAM workflow. On creation of the BAM workflow, BAMData and SKEventData Xmlvariables are created in the Start action. When user attaches the BAM Event to trigger the BAM workflow then these XML variables are populated with the data that can be used further. See BAM Workflow for more details.

Set the Threshold Value in BAM

To set the threshold value in BAM

1. In the Process Designer, point to **Settings**, and then click **BAM Configuration**. The **BAM Configuration** window appears.
2. Define the required threshold value.
   a. Threshold value settings for Overdue
   b. Threshold value settings for KPI Data Points level

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

c. Threshold value settings for KPI(s) level

To set the threshold value settings for Overdue

1. In the **Overdue Analysis** tab, set the threshold value for the total pending count or for a specific period of time.
2. In the **Total Pending Workflows Threshold** drop-down, set the threshold value for the total pending workflows.
3. In the **Slot Period** drop-down, select the slot period for defining the threshold values in each slot.
4. In the **Threshold** drop-down, select the threshold value.
5. Click **Save**

To set the threshold value settings for KPI Data Points level

1. Click the **KPI Data Points** tab to define the data points. The **KPI Data Point(s) Data Point** page appears.
2. Click **New**.
3. In the dialog box that appears, type a name for the KPI Data Points in the **Name** box.
4. Create a condition based on the requirements.
5. Click **Insert** to enter the condition.
6. Click **Save**.

Threshold value settings for KPI(s) level

1. Click the **KPI(s)** tab to define thresholds for the created KPI Data Points. The **Key Performance Indicators for the Workflow** page appears.
2. Click **New KPI**.
3. In the **Name** box, type a name for the KPI.
4. Click **Add New** to add a new Data Point.
5. From the **KPI Data Point** drop-down, select a data point to attach.
6. From the **Threshold** drop-down, select **Value**.
7. Click the **Insert** icon.
8. Click **Save**.

Set BAM Alert for Overdues and KPIs

To set a BAM alert for Overdues and KPI(s)

Step 1: In the BAM Configuration window, click the **Alert Configuration** tab to set the rule for the workflow.

The Overdue and KPI(s) settings can be configured to create a rule.

- Overdue: Create a rule to check pending workflows. Configure the following fields:
  - Total Workflow count is: The total number of pending workflow instances is compared with the value given by the user in the Threshold value.
  - Count in each slot is: The total number of pending workflow instances in the selected slot.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- KPI(s): Create a rule for a specific type of work. Configure the following field:
- Count in each datapoints of KPI(s) is: The number of workflow instances in each datapoint attached to the selected KPI.

  Step 2: In this step, you specify the duration for the rule to execute.

- Recurring: The configured rule will be executed regularly at a specific time.
- Only between: The configured rule will be executed at a specified time interval.
- Always: The configured rule will be executed every time the BAM report is being processed.
- Disable Alert: The configured rule will be disabled.

  Step 3: In this step, you specify the behavior of the alert.

1. Select the **Send an Email** to check box, and then type the recipient's Email Id(s) or select the Email Id(s) by clicking **Select**.
2. Select the **Send the chart(s) as a mail attachments** check box to attach the chart in the mail.
3. The email content entered can be customized using the formatting toolbar.
4. Click **Save** to save the alert settings.

BAM Events Associations

You use this feature to trigger a workflow using BAM Events.

To associate a workflow with a new BAM event, follow the steps given below

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**.

   The **Workflows** page appears.
2. Select an workflow.
3. On the action bar, click **Associate**.

   The **New Association** wizard appears.

   Alternatively, select and right-click the workflow to be associated. Click **Create Associations**.
4. On the action bar, click **BAM** to associate the BAM events.
5. In the **Workflow** drop-down list, select the workflow from which the BAM Events of Overdue and KPI are to be raised.
6. In the **Based** on drop-down list, select either **Overdue Analysis** or **KPI Analysis**.
   a. If you select **Overdue Analysis**, the **Select Overdue Slot** button appears. This is used to select the Overdue slot and the event associated with it.
   b. If you select **KPI Analysis**, then the **Select Slot** button appears. This is used to select the KPI name, attached KPI data points and the event associated with it.
7. Select the date range for **Event** execution using the **Effective From** and **Effective To** options.
8. Click **Attach Event** to trigger the Workflow when the BAM Report is being processed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

BAM Workflow

To create a BAM workflow

1. In the **Enterprise Console**, click the **Menu** button, and then click Workflows. The **Workflows** page appears.
2. On the action bar, click **New**, and then click **BAM**. The **New BAM Workflow** wizard appears.
3. In the **Title** box, type a name for the workflow, and in the **Description** box, type a description for the workflow.
4. Select the BAM event type by which the newly created workflow has to be triggered. You can also do this using the **Associate** option in the Enterprise Console site.
5. Select the date range for event execution using the **Effective From** and **Effective To** options.
6. Click **Save & Continue** to design the workflow. The workflow is created, and the **Workflow Options** wizard appears.

Usage of BAM Workflow

After creating the BAM workflow, the following XML variables get created automatically:

- **SKEventData**: Contains the values related to Overdue or KPI(s) with the Event Types.
- **BAM**: Contains the values related to the BAM Data (Overdue/KPI(s)) with the runtime and threshold values.

When the user attaches the BAM Event to the workflow, the XML Variables get updated with the appropriate data and can be used further in the workflow.

Example to Configure BAM Alerts

Example:

An Administrator for the Travel Desk Process needs to monitor the following:

- The count of domestic bookings that exceed the threshold value defined by him. This can be achieved by setting BAM alerts.
- The total number of pending workflows or the number of pending workflows for a specific period.

The first scenario can be achieved by creating a KPI with the data point ‘Domestic’ with the Threshold Value set. The second scenario can be achieved by setting the threshold value for the Overdue total count or for a specific period of time.

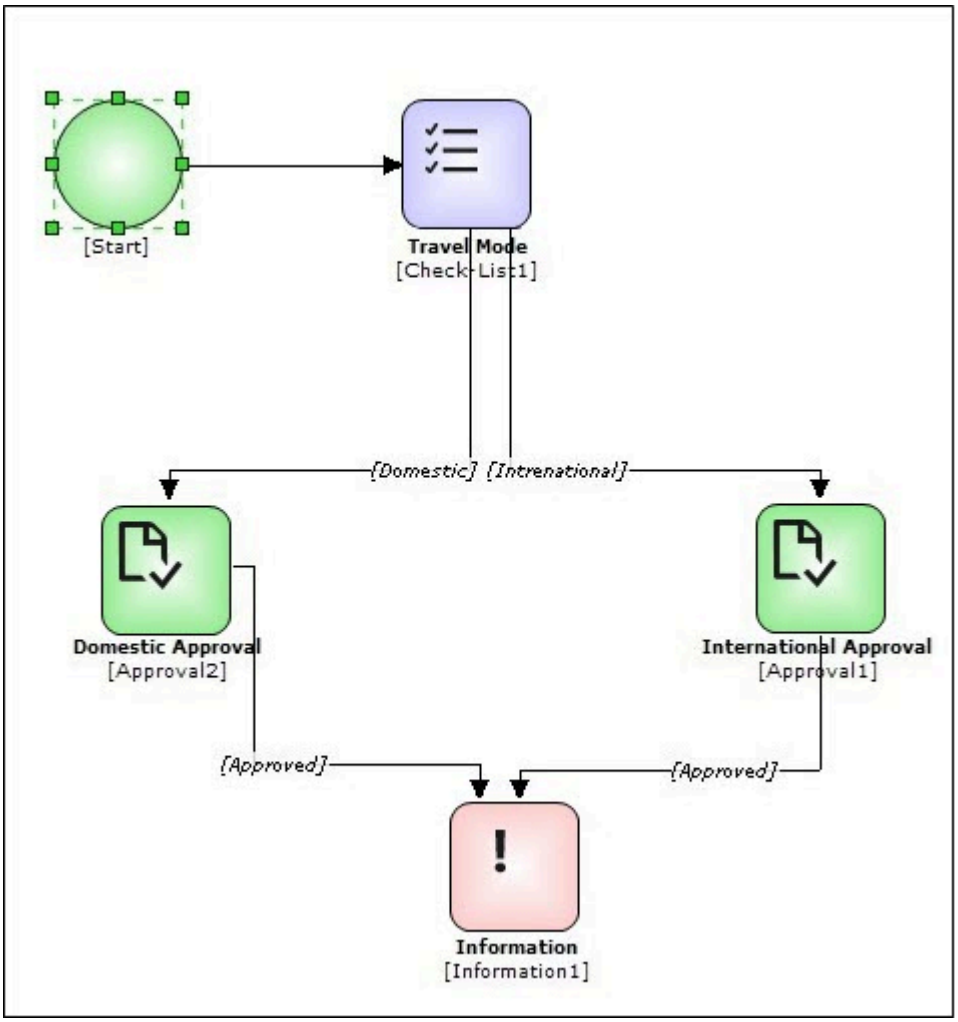
BAM Alert Settings

The following example demonstrates BAM alert settings for a sample travel desk process workflow.

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**. The **Workflows** page appears.
2. On the action bar, click **New**, and then click **BAM**. The **New BAM Workflow** wizard appears.
3. In the **Title** box, type a name for the workflow, and in the **Description** box, type a description for the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click **Save & Continue**. The workflow is created, and the **Workflow Options** wizard appears.
5. Select **Design the Workflow now**, which is the default option, and then click **Finish**. The **Process Designer** appears.
6. Design the workflow. For example, *Travel Desk Process*.



7. Set **Activity Properties** for all activities. Refer *Developer Guide* to configure.
8. Click **Publish** to publish this Workflow. The message about successfully publishing the workflow appears.
9. Click **Close**.
10. In the Workflow list page, right-click the *Travel Desk Process* workflow, and then click **Design Workflow**. The **Process Designer** appears.
11. Click **Settings**, and then click **BAM Configuration**. The **BAM Configuration** window appears.

Enabling and Disabling SOA Folder

To set the threshold value in Overdue level

1. Click the **Overdue Analysis** tab to set the threshold value for the total pending count or for a specific period of time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. In the **Total Pending Workflows Threshold** drop-down, set the threshold value for total pending workflows. In this example, set the threshold value as 2 for the total pending workflows.
3. In the **Slot Period** drop-down, select the slot period for defining the threshold values in each slot. In this example, define the Slot1 period as 1 Hour.
4. In the **Threshold** drop-down, select the threshold value. In this example, enter a threshold value for the number of pending Workflows for the Slot1 period as 2.
5. Click **Save**.

These settings display a graphical view of the pending workflows against threshold.

Create KPI Data Points

To create KPI Data Points

1. Click the **KPI Data Points** tab to define the data points that monitor **Overdue Analysis**. The **KPI Data Point(s) Data Point** window appears.
2. Click **New**. For this example, create '**Domestic**' and **'International**' Data Points.

To create Domestic Data Point

1. In the **Name** box, enter the name for the Data Point as Domestic.
2. In the **Condition based on** drop-down list, select **Activity Inlink Condition**.
3. In the **If Activity** drop-down list, select **Approval2**.
4. In the **has started because of Incoming Link having Condition** drop-down list, select **IN**.
5. Check **Domestic** from the list of check boxes.
6. Click **Insert** to insert the condition in the Condition text area.
7. Click **Save**.

To create International Data Point

1. Click **New.**
2. In the **Name** box, enter the name for Data Point as International.
3. In the **Condition based on** drop-down list, select **Activity Inlink Condition**.
4. In the **If Activity** drop-down list, select **Approval1**. (Approval1 is for International)
5. In the **has started because of Incoming Link having Condition** drop-down list, select IN.
6. Check **International** from the list of check boxes.
7. Click **Insert** to insert the condition in the Condition text area.
8. Click **Save**.
9. The two Data Points are created.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create KPI(s) and Set the Threshold Value

To set the threshold value in KPI Level

1. Click the **KPI(s)** tab to define thresholds for the created KPI Data Points.
2. Click **New KPI**.
3. In the **Name** box, enter the name Travel Desk KPI for the KPI.
4. Click **Add New** to add a Domestic Data Point.
5. In the **KPI Data Point** drop-down, select Domestic data point to attach.
6. In the **Threshold** drop-down, select Value, and then enter 5 in the text box.
7. Click the **Insert** icon.
8. Click **Add New** to add International Data Point.
9. In the **KPI Data Point** drop-down, select International data point to attach.
10. In the **Threshold** drop-down, select **Not Set**.
11. Click the **Insert** icon.
12. Click **Save**.

BAM Alert Configuration Settings

There are the following three **Alert Configuration** settings:

- **Rule**: Create rules for both Overdue and KPI based on the threshold value set.
- **Time**: Set the time for the alert to be triggered.
- **Action**: Set an action, if the rule is satisfied.

BAM Rule Settings

To create a rule for pending Workflow (Overdue)

1. In the **Alert Configuration** tab, click **Rule** to create rules for both Overdue and KPI. By default, the **Rule Settings** window appears.
2. Check **Total Workflow count is**, and then select **GreaterThan**.
3. Check **Count in each slot is**, and then select **GreaterThan**.
4. Click **Select Slots** to set the rule for slots. The **Select Slot** dialog box appears.
5. Select the slot, and then click **OK**.

To creating rule for specific type of work (KPI)

1. Check **Count in each datapoints of KPI(s) is**, and then select **GreaterThan**.
2. Click Select KPI(s) to set the rule for KPI(s) that need to be monitored. The Select KPI(s) dialog box appears.
3. Select the KPI, and then click **OK**.
4. Click **Save**. A message about successfully creating alert appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Click **OK**. This alert is triggered when the BAM report is being processed.

BAM Time Settings

To configure time settings for a BAM workflow

1. In the **Alert Configuration** tab, click **Time** to set the time preference for the alert to be triggered.
2. Select the **Always** option to trigger an alert every time the BAM Report is being processed.
3. Click **Save**. A message about successfully creating alert appears.
4. Click **OK**.

BAM Action Settings

To configure action settings for a BAM workflow

1. In the **Alert Configuration** tab, click **Action** to set the notification email when the rule is triggered.
2. Check **Send an Email to**, and then type the recipient's Email Id(s). Alternatively, click **Select** to select Email Id(s) from the User Lookup. Multiple email addresses can be specified by separating the emails with semicolons.
3. Check **Send the chart(s) as a mail attachments** check box to be embedded in the mail.
4. Type a personalized message as email content to be sent along with the mail.
5. Click **Save**. A message about successfully creating alert appears.
6. Click **OK**.

BAM Events in Enterprise Console Site

To associate a BAM event to trigger a workflow

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**. The **Workflows** page appears.
2. Select the workflow to be associated.
3. On the action bar, click **Associate**. The **New Association** wizard appears.
4. Alternatively, select and right-click the workflow to be associated. Click **Create Associations**.
5. Click **BAM** to associate the workflow with BAM events.
6. In the Workflow drop-down list, select the **Travel Desk Process** workflow.
7. In the **Based On** drop-down list, select either **Overdue** Analysis or KPI Analysis. In this example, select **KPI Analysis**.
8. Click **Select Slot**.
9. Select the **Travel Desk KPI** option.
10. In the **Domestic** drop-down list, select **GreaterThan**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

11. Click **Select**.
12. Select the date range using the **Effective From** and **Effective To** options.
13. Click **Attach Event**. A message about successfully associating the event appears.
14. Click **OK**.

The event gets associated to the workflow. When the **BAM Report** of selected workflow is being processed or the event is raised, the workflow gets triggered.

Trigger a BAM Alert

You can trigger an alert by processing the BAM Report manually, or by clicking **Process Now** in Basic Configuration, or by scheduling the BAM Report process.

To trigger an alert manually

1. In the **BAM Configuration** window, click **Basic Configuration**.
2. Click **Process Now** to trigger the alert manually. A message about successfully completing the BPM updating process appears.
3. Click **OK**.

Business Activity Monitoring and Reports

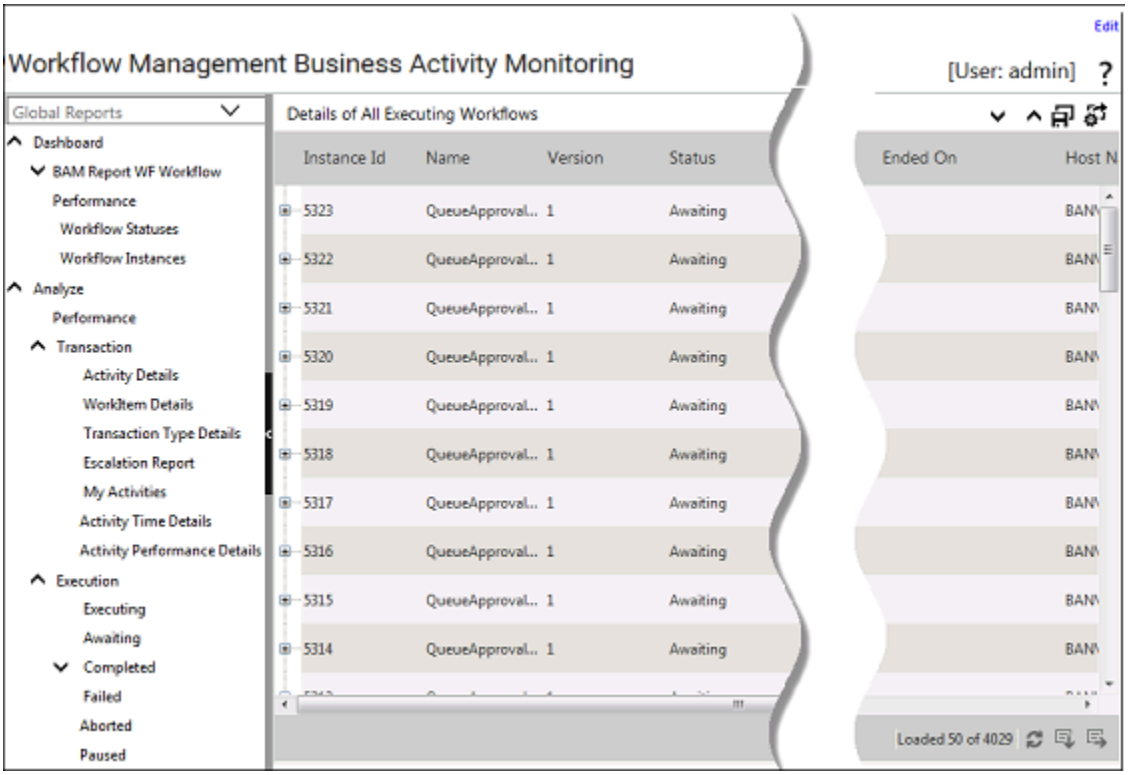
The BAM Control provides extensive reporting capability and true BAM functionality. It provides users with several out-of-the-box reporting and activity monitoring views. In addition, it allows users to configure Data Points and KPIs for comprehensive business activity monitoring. It is one of the crucial controls in AVEVA Work Tasks.

You can see a three-dimensional view of the following BAM Reports by rotating the axis of the graphs:

- KPI Bar Chart
- KPI Pie Chart
- Overdue Analysis
- Workflow instances
- Workflow status

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Default Report View



The BAM Control page displays the following default report view when the Control is executed. You can set the application name and workflow name to the respective property of the BAM Control. If no workflow name is set then the BAM Reports are shown for the entire application, covering all the workflows in it. This constitutes the "Global Reports". At the top right of the Report window are the [+] and [-] icons to expand and collapse all the nodes respectively.

Report Type Tree View

The Default Report window represents the general layout of the BAM Control user-interface, which it shares with all other report windows. The user-interface elements and functionality described in this topic apply to all the other report windows. The window has the **Workflow** list at the upper left. It lists all the workflow names in the application along with the "Global Reports" option. When any workflow name is selected from the list, all the reports will display information related to the selected workflow.

The Report Type tree has the following nodes:

- **Dashboard**: This report type represents the activity monitoring dashboard of the BAM Control. It has two dashboard views, **Workflow Status** and **Workflow Instances**.
- **Analyze**: This report type represents a number of out-of-the-box reports grouped under two sub-types, **Transaction** and **Execution,** each with a set of reports.

Note: The BAM tree in the Business Activity Monitoring interface, by default, shows a one level expanded view instead of a fully expanded tree view.

The report view is on the right of the window. It presents the report information in several columns, with the column displayed depending on the type of report selected. You can right-click a column name to open the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

shortcut menu, where you can perform report list management operations such as sorting and grouping on the columns.

- **Sort Ascending:** Select this option to sort the list on this column in the ascending order
- **Sort Descending:** Select this option to sort the list on this column in the descending order
- **Group By This Field:** You can use this option to group the list by this column
- **Group By Box:** You can use this option to display the Group By box at the top. You can drag-and-drop a column header in this box to group the list by that field. This is an alternative way to group the list.
- **Remove This Column:** You can use this option to remove the selected column
- **Best Fit:** You can use this option to change the column width to best fit
- **Filter Bar:** You can use this option to turn on or turn off the filter bar
- **Select Columns:** You can use this option to expand a menu with all the column headers as shown in the following image. Each column name has a check box that you can use to select it or to remove the selection. After making any change to the column selection, you can click the **Apply Changes** option for the new settings to take effect. You can click the **Reset** option.

Save Layout

You can customize the view and save the layout by clicking the **Save Layout** icon. This opens a window where you can specify a name with which the customized report layout will saved.

You can enter the file name for the saved layout in the **Layout Name** box. After specifying the name, click **OK** to save the layout. You can click **Cancel** to close the window without saving the layout. Any layout saved in this window will be added under the node corresponding to the report type being saved as a custom layout (for example, Executing). You can select it from here to view the report in the customized layout.

In addition to the other icons, a **Delete** icon is displayed when you open a custom layout. You can click the delete icon to delete the open custom layout.

Export Settings

Use the **Export Settings** icon to set the number of records to export.

1. When you click the **Export Settings** icon, the **Export Settings** dialog box appears.
2. Enter the number of records that you want to export.

   You can enter only valid positive integer values.
3. Click **Update**.

   The Reports grid refreshes and displays the number of records entered in the **Export Settings** dialog box.
4. Click the **Export Grid** icon to export the report.

When you enter a value for export, the number of records exported will be the first record to the number entered. For example, if the grid contains 25 records and you want to export 12 records, records from 1 to 12 will be displayed in the refreshed grid and can be exported. You cannot enter a value range for export. Also, if the number entered exceeds the maximum records present, the highest number of records present will be exported.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Export Grid

You can export the contents of the report to different formats using the Export grid tool at the lower right of the window.

Clicking the export tool opens the export type selector, which lists all the available types with radio buttons to select them.

There are three sets of options. The first has the **Export root table,** which can be clicked to select the root table for export. The second set lists all the common file formats, such as HTML, PDF, XML, Excel, TIFF, RTF, and TEXT. The last set has the **Portrait** and **Landscape** options to select the orientation of the exporter report. After making the export settings you need to click Export now to export the file.

Escalation Report:

Description:

The Escalation Report gives in-depth details about escalations. It shows the user name, date, timeout settings, escalation type (Recurring or Non-Recurring), task details, and workflow and application details. The report has a drill down feature which shows the **Process Execution** view for the instance. Double-click the data to view the Process execution details.

Location:

On **Enterprise Console**, clicking the **left <Analysis>** button brings up the **Business Activity Monitoring** (BAM). In the BAM Left Panel Tree view, click the **<Analyze>** node. Under this, you can locate the **<Transaction** > node which has the **Escalation Report** option. See the following image to identify the report.

Report Parameters:

By selecting the 'From' and 'To' dates, the user can get the details of the escalations that occurred for the given date range. To see the 'Process execution view' for the instance, double-click the data.

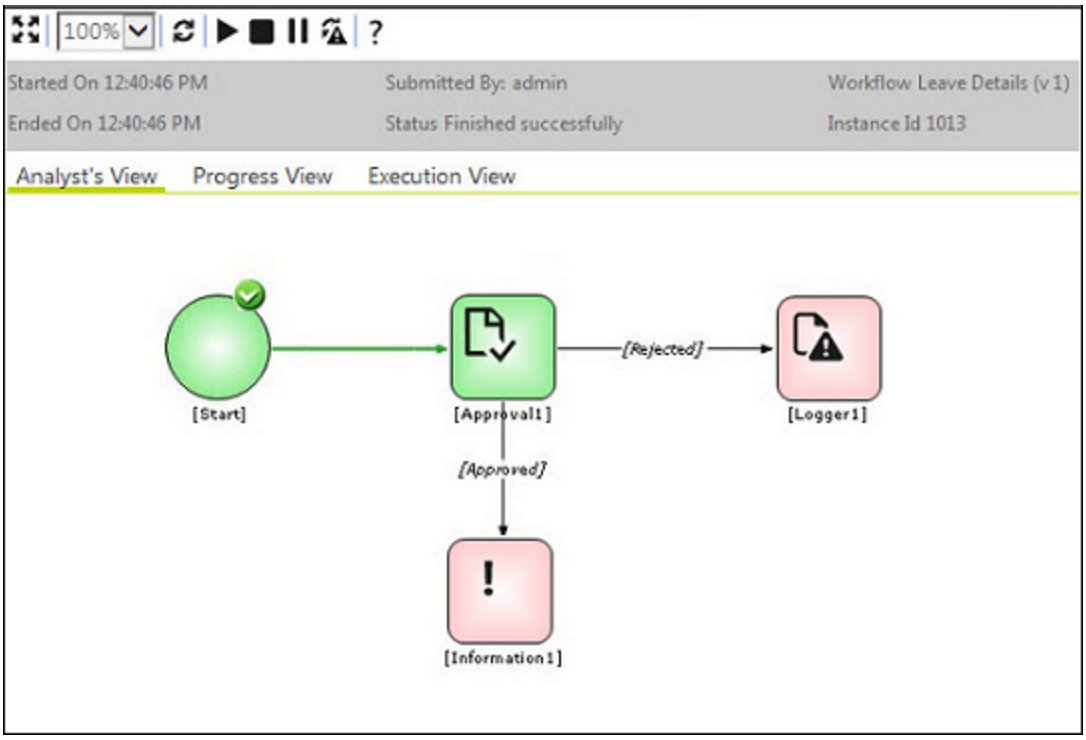
Report view:

It shows the user name, date, timeout settings, escalation type (Recurring or Non-Recurring), task details, and workflow and application details. Escalation report also has the column filter features available in other reports.

Double-click the data for the instance to view the **Process execution** view.

Process execution view:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Scenario:

John Smith works as the Administrator for 'XYZ' organization and wants to monitor the Escalations that occurred in the workflow for a given date range. By using the escalation report, Smith can get the details of the same and can take the necessary steps.

BAM Dashboard View

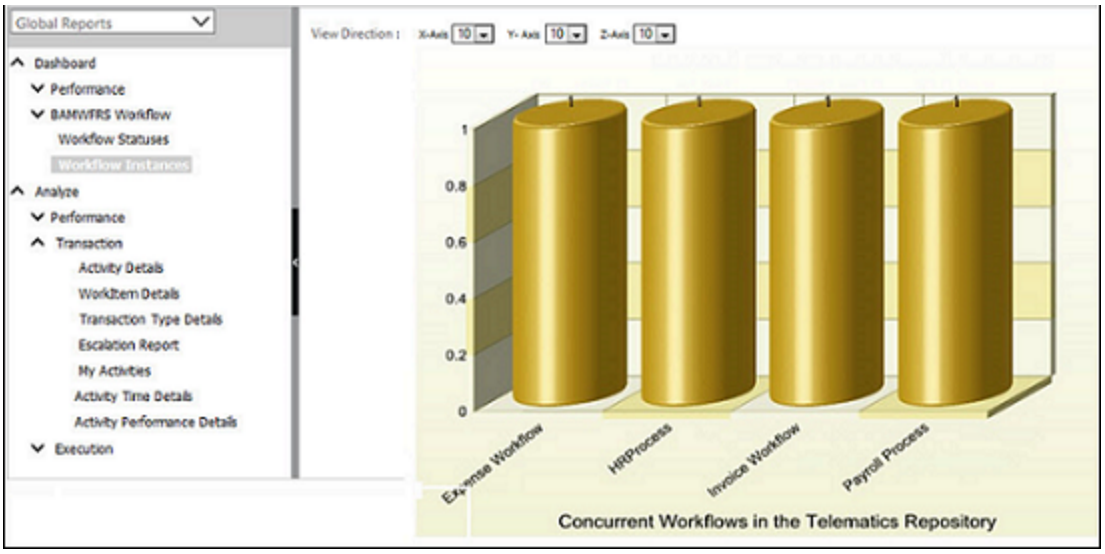
The **Business Activity Monitoring** Dashboard presents an activity monitoring dashboard to the users. It has two default dashboard views - **Workflow Statuses** and **Workflow Instances**, which retrieve workflow execution data dynamically and display it in the respective charts.

The dashboard also shows Overdue Analysis reports if overdue analysis configuration has been set for the workflow. For more information, refer to the BAM Configuration and Overdue Analysis Reports.

Workflow Statuses - This view shows a pie-chart representing the status of workflows being executed by the application.

Workflow Instances - This view shows a pie-chart representing the workflow instances being executed in the application.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



BAM Execution Reports

For token-based repositories such as AIM repository, Azure Active Directory repository, and AVEVA Connect repository, the Human Activities assigned to **Workflow Initiator** will not receive work items in the Inbox for the following events:

- Scheduler
- File Watcher
- BAM
- ArchestrA Event
- Message Queue
- Logic App Event (only for AVEVA Connect)

For non-token-based repositories, the **initiator** will be the user who executed the workflow or the one who created the event.

Execution Reports

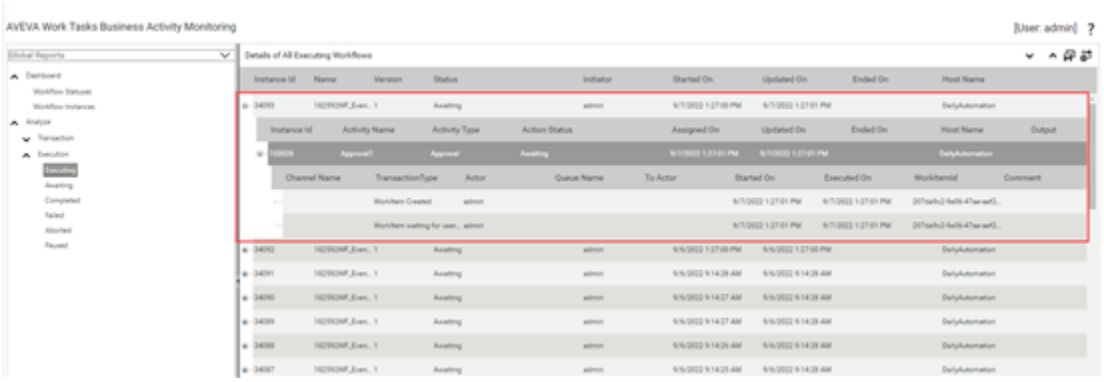
This type of report presents workflow execution information related to execution status. There are four types of reports in this group.

- Executing
- Awaiting
- Completed
- Failed
- Aborted
- Paused

Executing

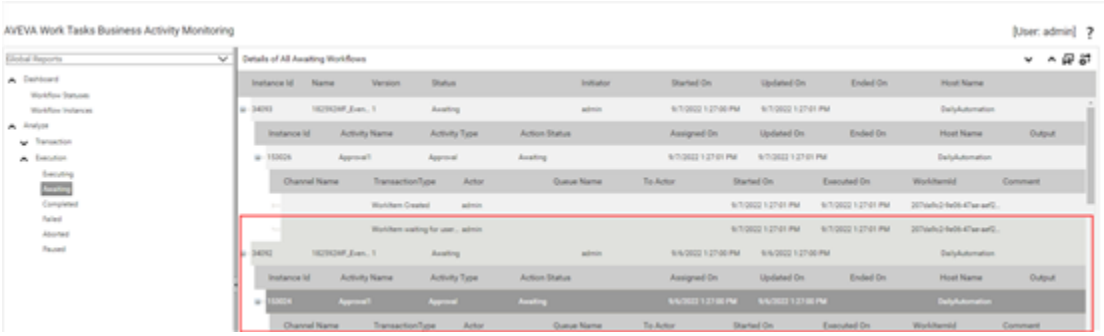
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

This report presents details related to executing workflows. The following image shows the **Executing** report. It has been split into two for convenience.



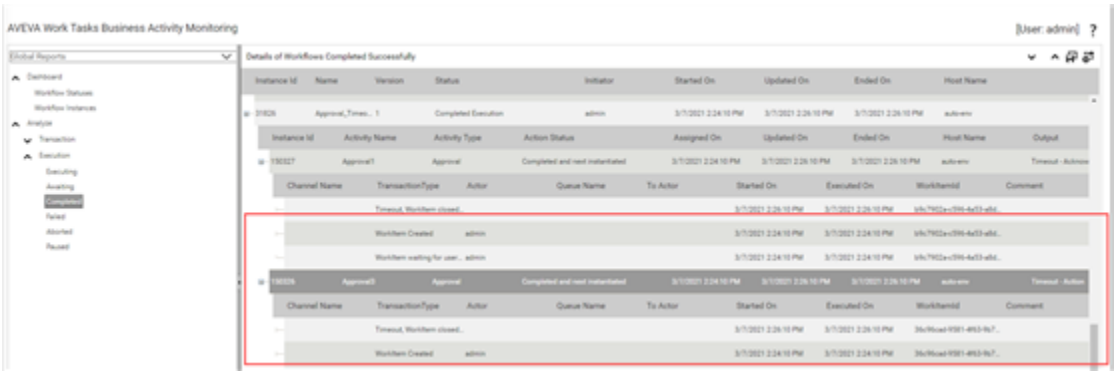
Awaiting

This report presents details related to workflows awaiting execution. The following image shows the **Awaiting** report. It has been split into two for viewing convenience.



Completed

This report presents details related to completed workflows. The following image shows the **Completed** report view. There are no rows in the report because there are no completed workflows to represent.



Failed

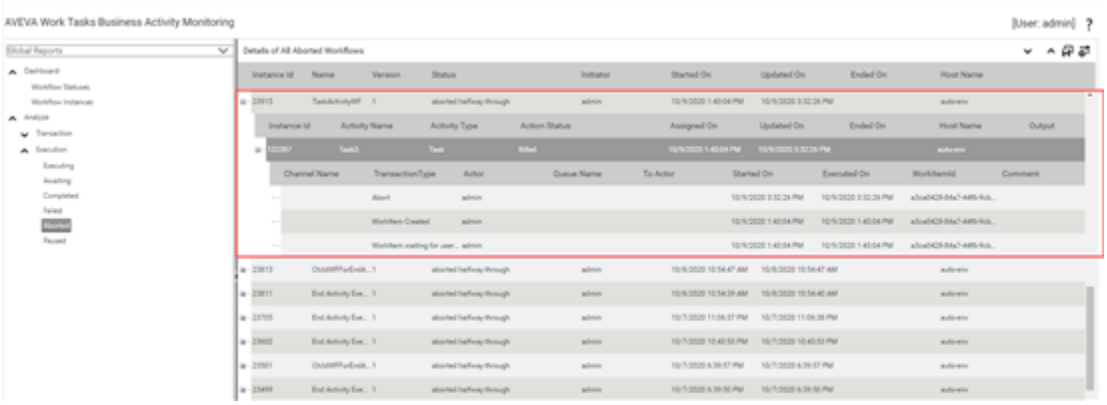
This report presents details related to the failed workflows. The following image shows the **Failed** reports.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Aborted

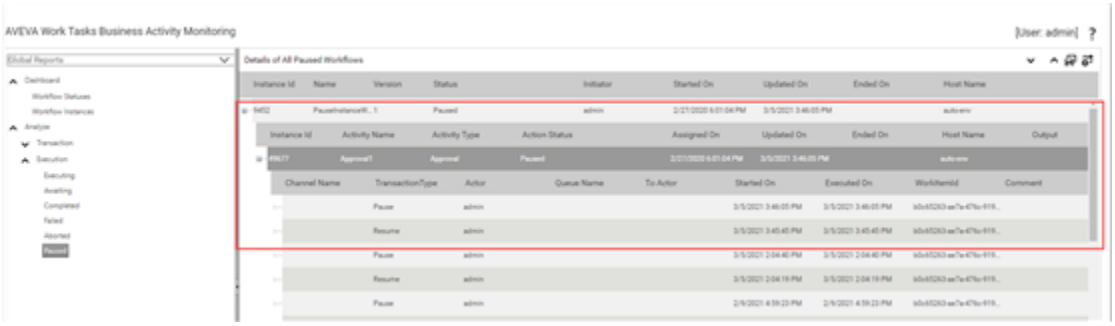
This report presents details related to the aborted workflows. The following image shows the **Aborted** report view.

Quickflows completed with errors is listed under **Aborted** instead of getting listed under **Failed**.



Paused

This report presents details of the paused workflows. The following image shows the **Paused** report view.



BAM Analyze Reports

The Analyze reports constitute the reports available out-of-the-box in AVEVA Work Tasks. It has two sets of reports, Transaction and Execution.

The Analyze section also shows **Performance Analysis** reports if the performance analysis configuration has been set for the workflow. For more information, refer to BAM Configuration and Performance Analysis Reports.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

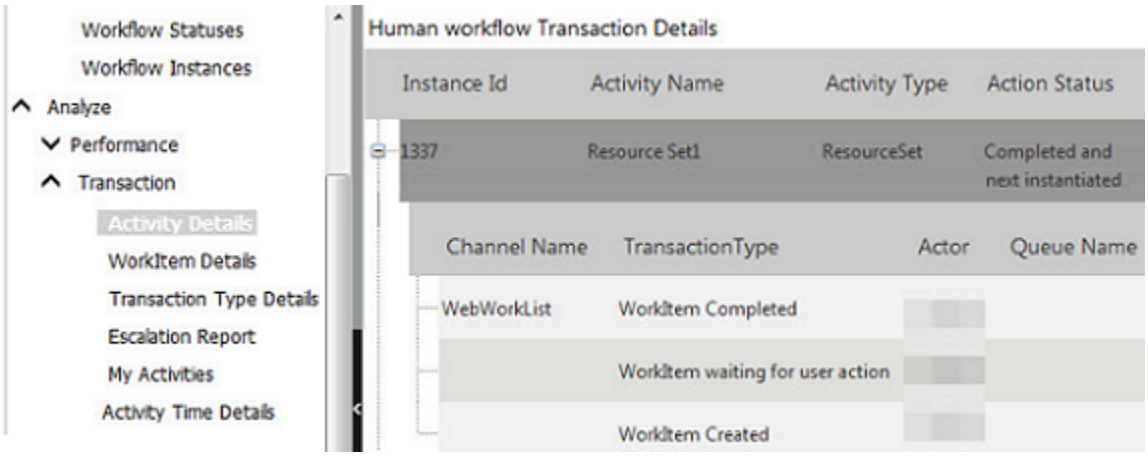
Transaction Reports

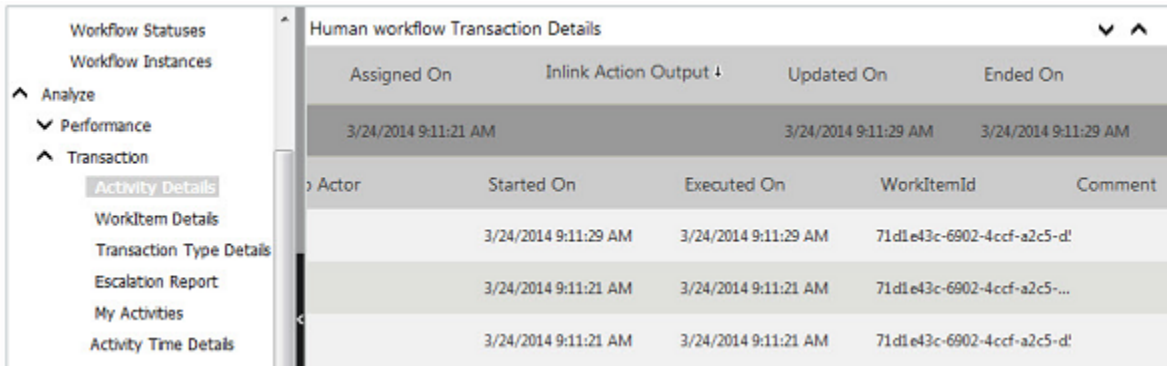
These reports present workflow execution information related to transactions in the workflow. There are seven types of reports in this group:

- Activity Details
- Work Item Details
- Transaction Type Details
- Escalation Report
- My Activities
- Activity Time Details
- Activity Performance Details

Activity Details

This report presents details related to activity-level transactions. The following image shows an **Activity Details** report. It has been split into three for convenience.

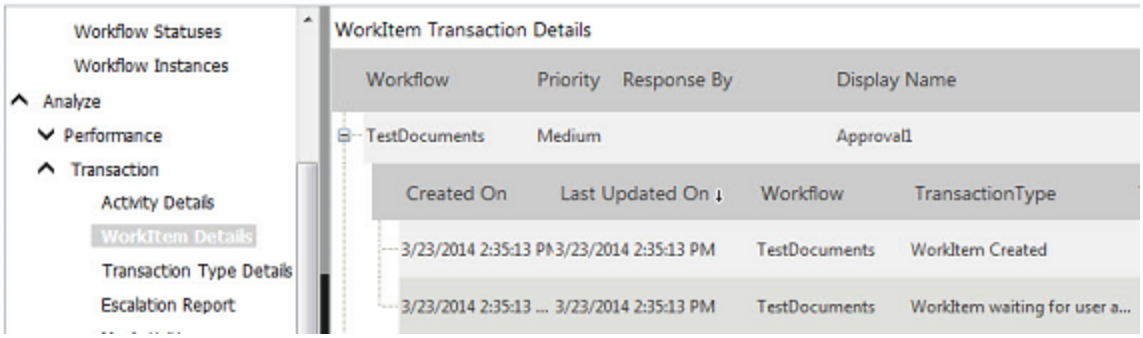


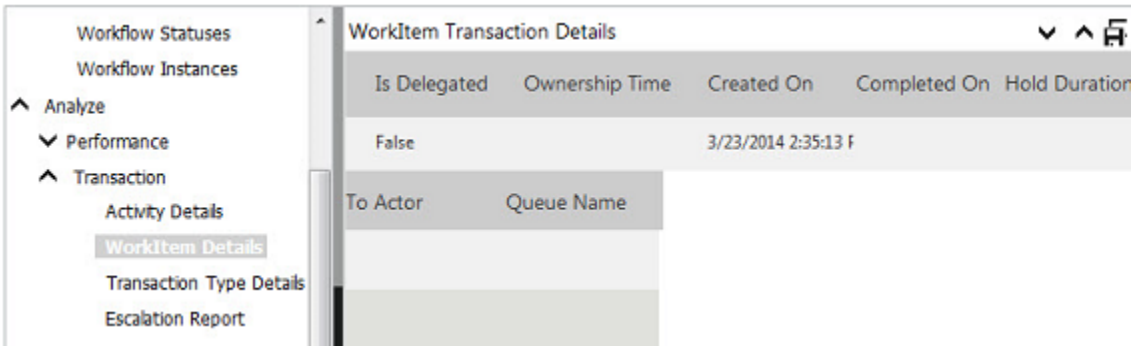


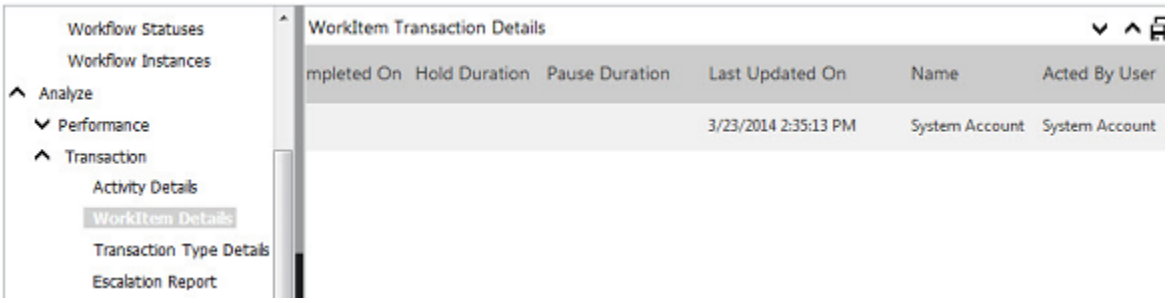
Work Item Details

This report presents details related to work item-level transactions. The following image has been split into three for viewing convenience.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

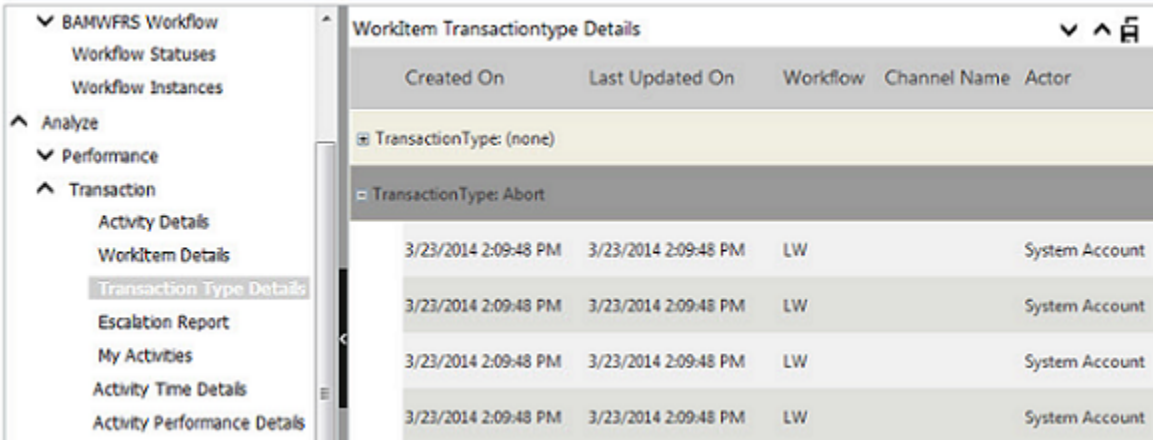






Transaction Type Details

This report presents details of different transaction types as shown in the following example.



Escalation Report

Description:

The Escalation Report gives in-depth details regarding escalations. It shows the user name, date, timeout settings, escalation type (Recurring or Non-Recurring) and task, workflow, and application details. The report has

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

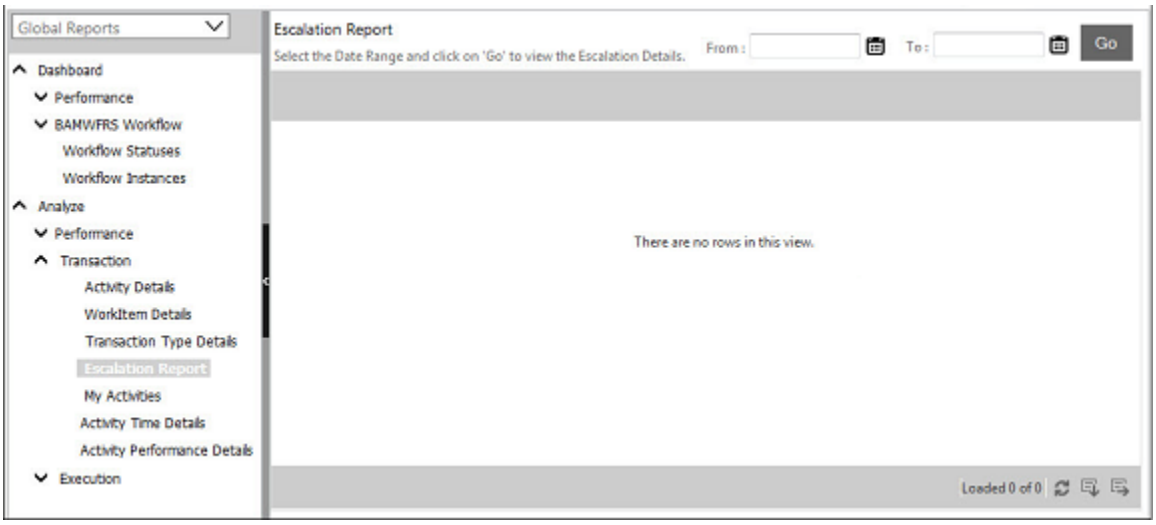
a drill down feature which shows the Process execution view for the instance. Double-click the data displays the Process execution view.

Location:

Selecting Reports from the **Enterprise Console** menu displays the **Business Activity Monitoring** (BAM) interface. In the BAM left panel tree view, click **Analyze** and then **Transaction**. Finally, click **Escalation Report**.

Report Parameters:

By selecting the 'From' date and 'To' date, one can get the details of the escalations that occurred for the given date range. You can see the Process execution view for the instance by double-clicking the data.



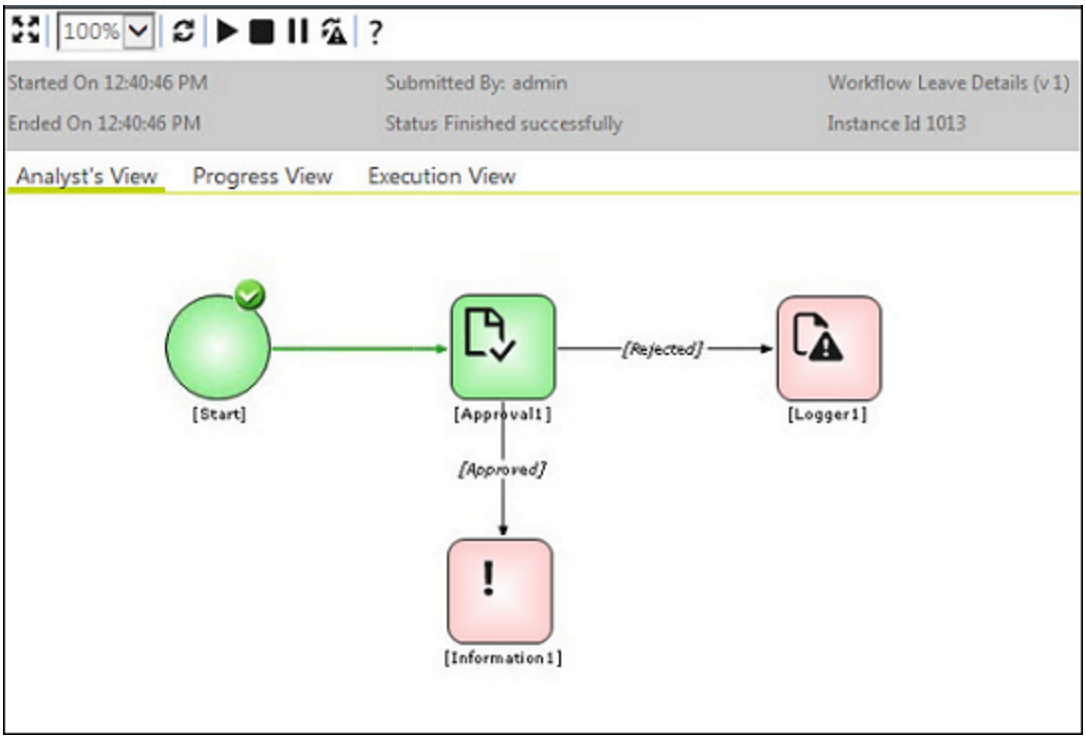
Report view:

It shows the user name, date, timeout settings, escalation type (Recurring or Non-Recurring) and task, workflow, and application details. The **Escalation Report** has the column filter feature available in the other reports.

Double-clicking the data for the instance displays the Process execution view.

Process execution view:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Scenario:

John Smith works as an Administrator for 'XYZ' organization and wants to monitor the escalations that occurred in the workflow for a given date range. By using the Escalation Report, they can get the details of the same and take the necessary steps.

My Activities

Description:

Do you ever want to see the activities that have been acted upon? The My Activities option is useful to keep track of the activities that you have acted on. By default, AVEVA Work Tasks keeps a copy of workflow history that you have acted on in My Activities. Based on security rights, the My Activities option is displayed.

Location:

Selecting Reports from the Enterprise Console menu displays the Business Activity Monitoring (BAM) interface. In the BAM left panel tree view, click Analyze and then Transaction. Finally, click My Activities.

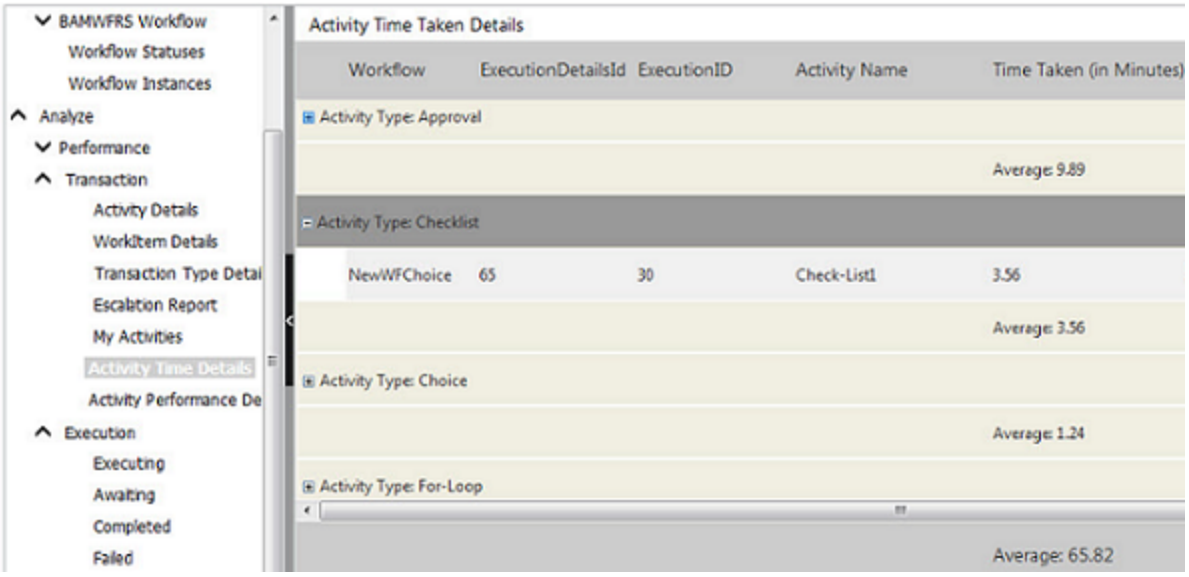
My Activities will give in-depth details regarding activities. It shows the Instance Id, Activity Name, Activity Type, Action Status, Assigned On, Inlink Action Output, Updated On, Ended On, Output, and Workflow details.

It has a drill-down feature which shows the 'Process execution view' for the instance. Double-clicking the data brings up the Process execution view for the workflow.

Activity Time Details

This report presents details related to **Activity Time** transactions, as shown in the following image. The report displays the time taken by each activity and its average time. This is related to all the Activities.

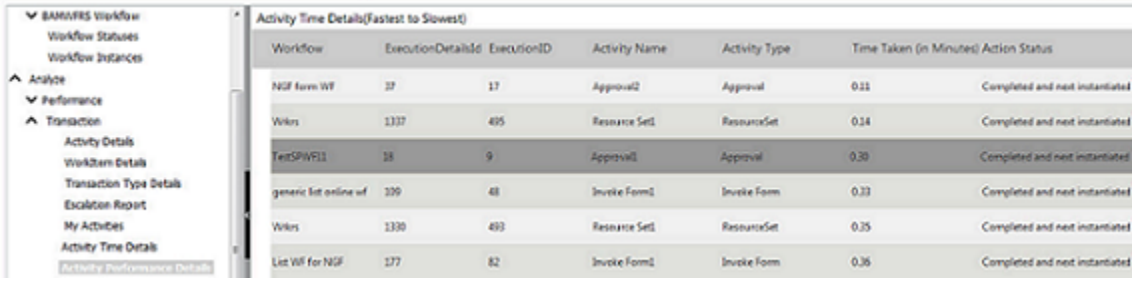
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Activity Performance Details

This report presents details related to Activity Performance transactions, as shown in the following image. This is only related to the **Human Activities**.

The report displays in descending order, from Best Time (least time taken to execute a particular activity) to Slow Time (most time taken to execute a particular activity).



BAM Process Execution View

The **Process Execution Views** show the workflow execution status graphically. You can view the **Execution View** window by double-clicking a top level row in any of the reports. The View represents the workflow up to the activity or execution status where the View was opened.

The toolbar at the top of the Execution View window has the following tools:

- **Full Screen** - Click this tool to toggle between normal and full-screen modes
- **Zoom** - Use this tool to zoom in and out of the process definition image
- **Refresh** - Use this tool to refresh the image
- **Resume paused workflows** - Use this tool to resume paused workflows
- **Abort executing workflows** - Use this tool to abort workflows that are being executed
- **Pause executing workflows** - Use this tool to pause workflows that are being executed
- **Resume workflows finished with errors** - Use this tool to resume paused workflows

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Tip: The Process Execution Views are very useful in navigating through large process definitions. You can click any activity in the View window to open the process definition in the Process Designer, at that point.

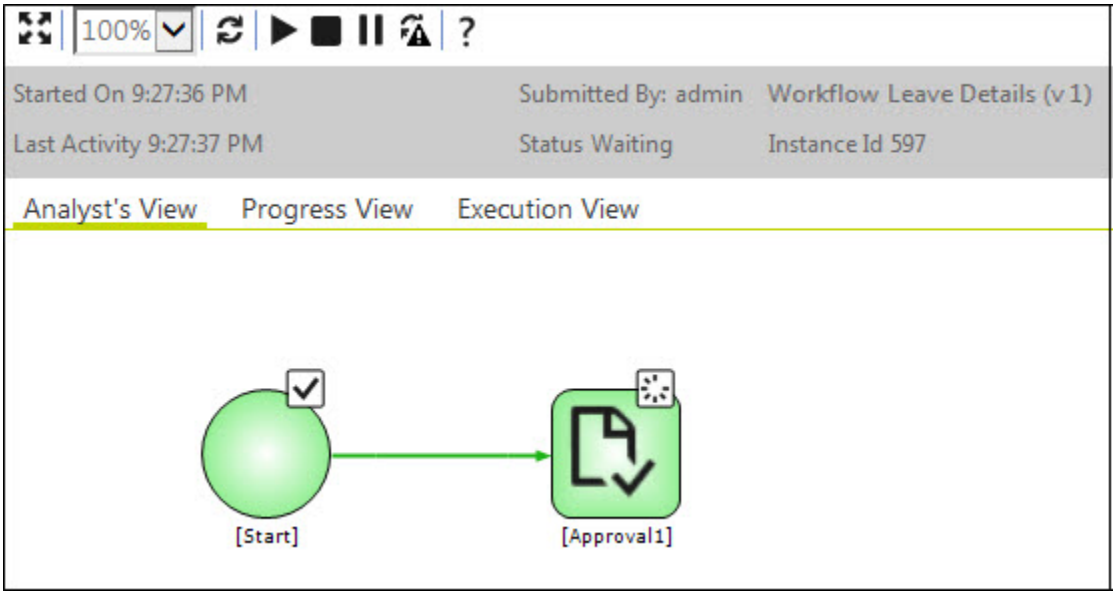
Note: A green circle, with a white check mark superimposed, is used to denote a completed activity or process. It is displayed on the top edge of the activity icon for all completed activities. This icon is displayed at the top right corner of the page on the title bar to denote a completed process.

There are three tabs at the bottom to navigate between the following three views:

- Analyst's View
- Progress View
- Execution View

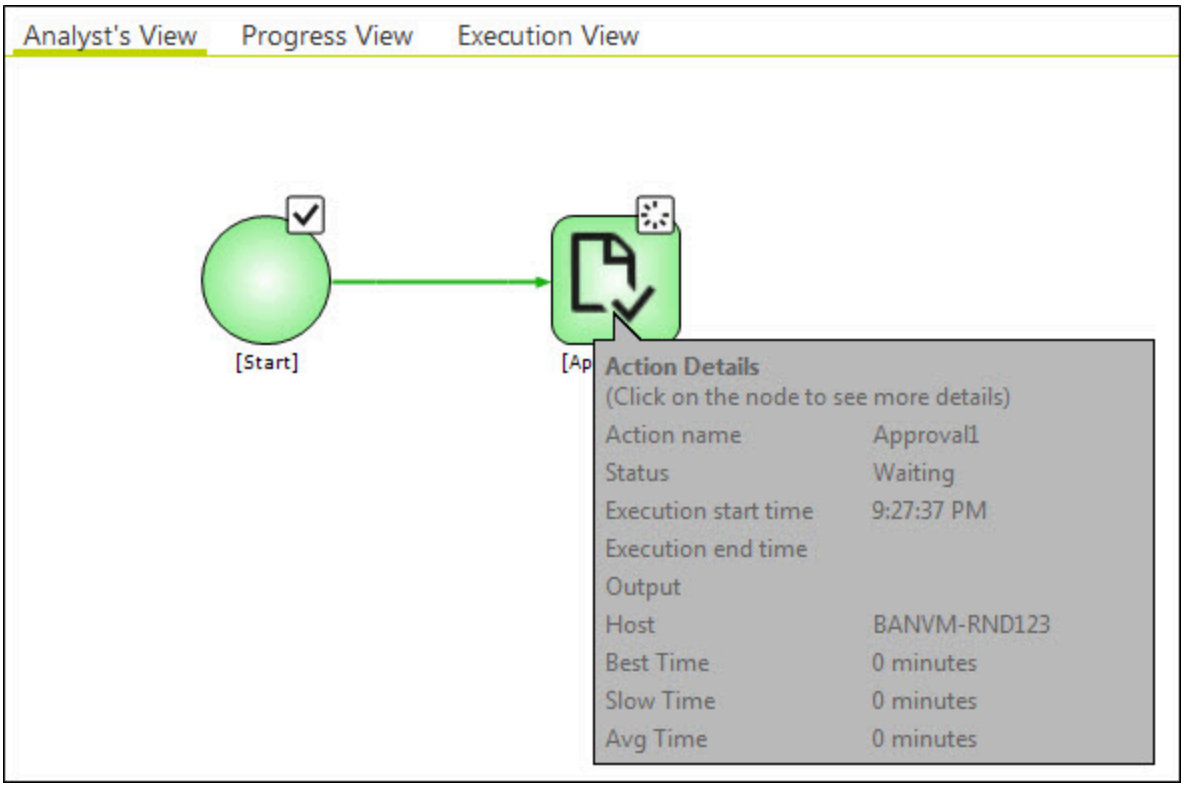
Analyst's View

The Analyst's View is a simplified view of the process design intended for end-users. It shows the process model that was saved as the Business Analyst's View after the business process modeling was completed and before the automation steps were added in the Process Designer. At this stage, the model would have been saved to a file. This file should be assigned to the **Analyst's Process Model Filename for Execution View** property of the Start activity to enable this view. In the following example shown, the Analyst's View has been opened by double-clicking the first row, which has been highlighted.



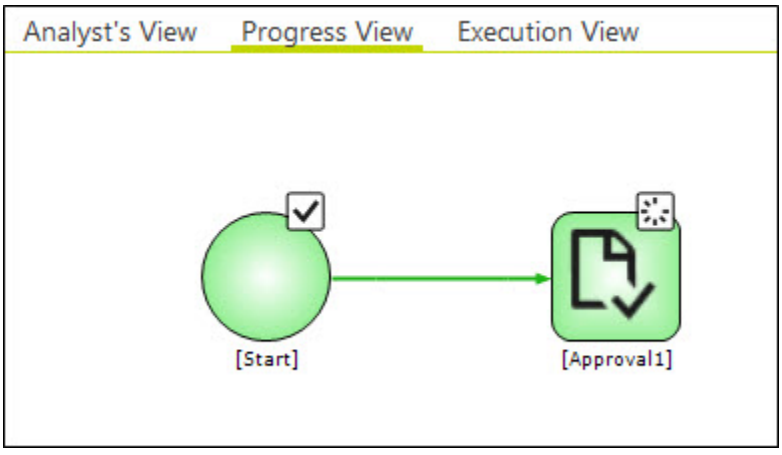
When you hover the cursor over a workflow activity in the Process execution view screen, it shows the Action Details, Best Time, Slow Time, Average Time taken by the activity for a workflow instance, as shown in the following image.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Progress View

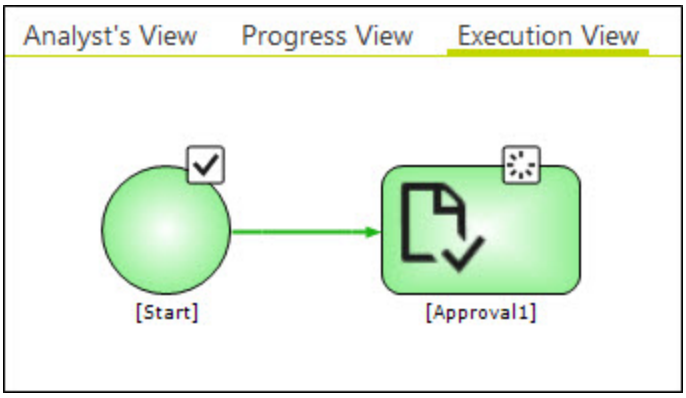
The **Progress View** shows the workflow progress on the **Process Diagram**, up to the point that the **Execution View** is selected. It consists of the execution view mapped on to the Process Diagram. You can click on a completed activity to view the **Step Details Window**. You can view the step details, variables, contents, and content (raw content passed to the workflow) using the corresponding tabs. The links that have already been traversed are represented by thick lines.



Execution View

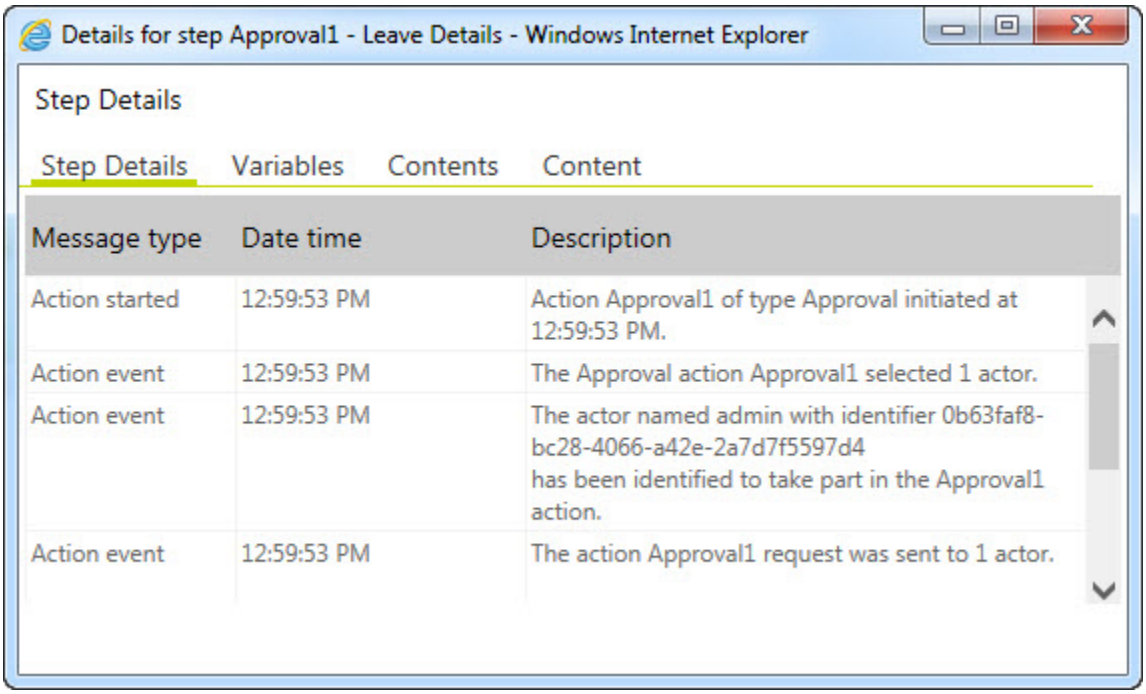
The Execution View shows the workflow execution status up to the point that the **Execution View** is selected. This View is identical to the Graphical View of Workflow Execution that was available in the Report Control of the earlier versions.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



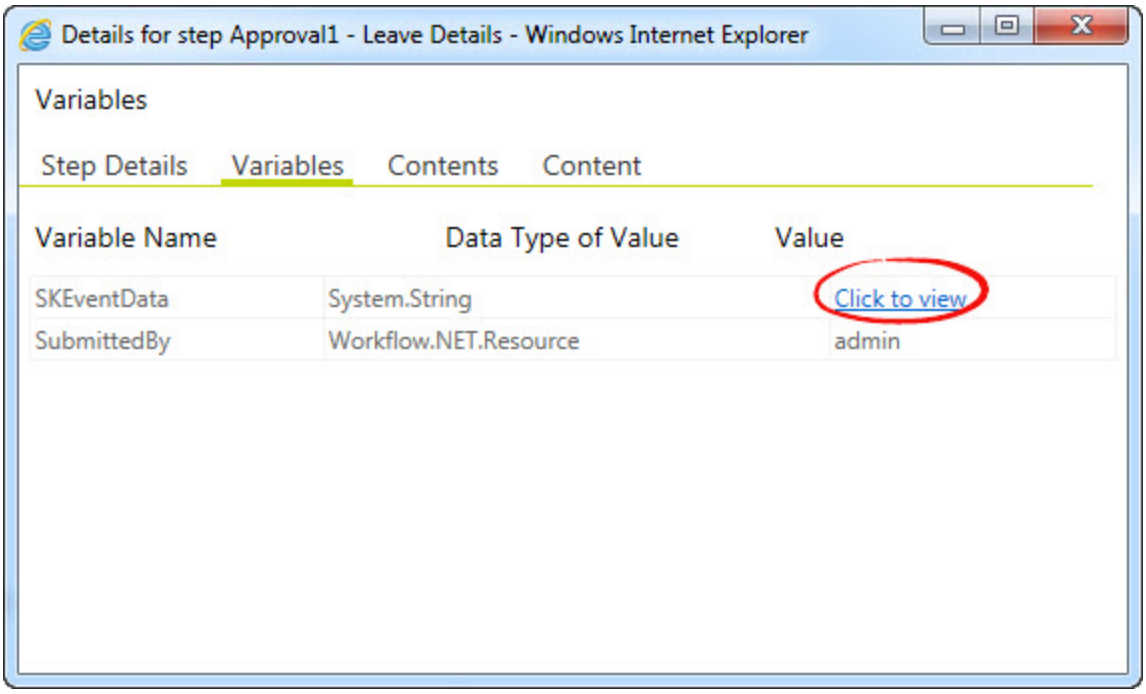
When you click the **Approval** activity, the **Details** page appears and displays the Step Details of the activity. The details screen shows information under the following five tabs - **Step Details**, **Variables**, **Contents**, **Content**, and **Submitted Form Details**.

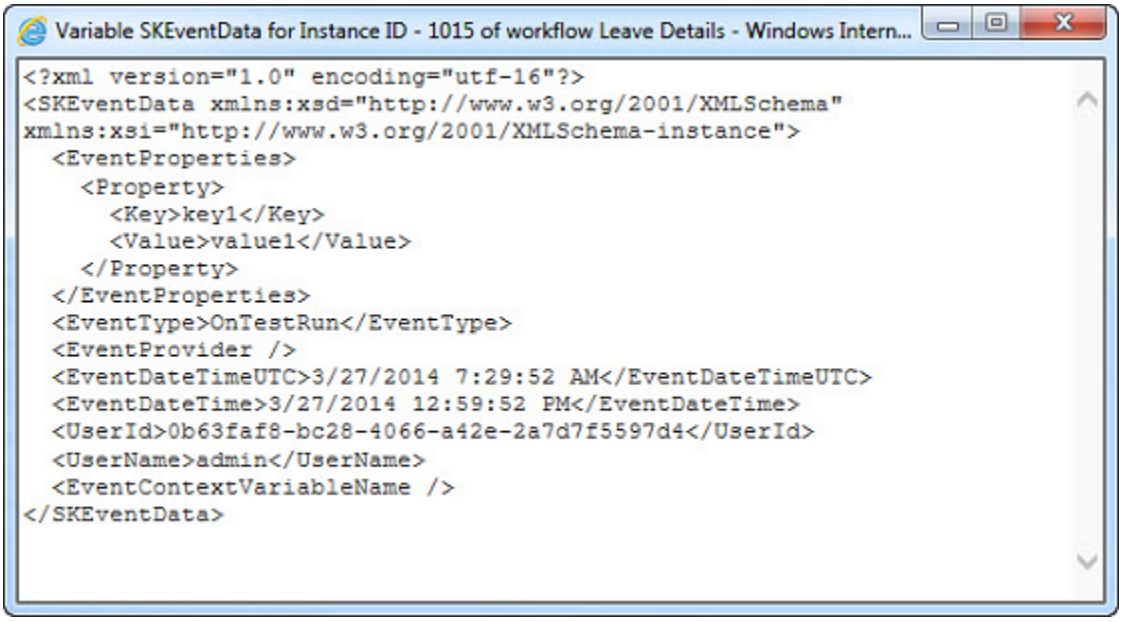
In the **Step Details** tab, the user can view all the details related to the **Approval** activity, such as the time of the action, type of action, name of the actor to whom this action has been assigned to, and the action performed by the actor



When you click the **Variables** tab, the screen showing the details of all the variables used in this activity will be displayed. You can view the runtime values of the XML variable by clicking the **Click to view** link.

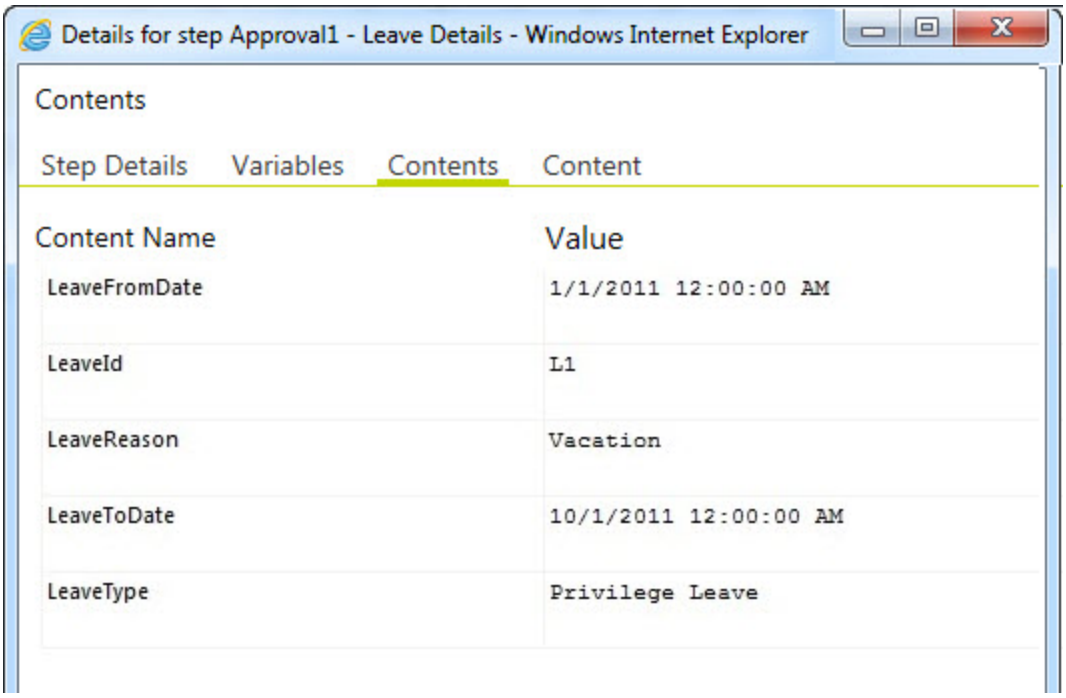
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



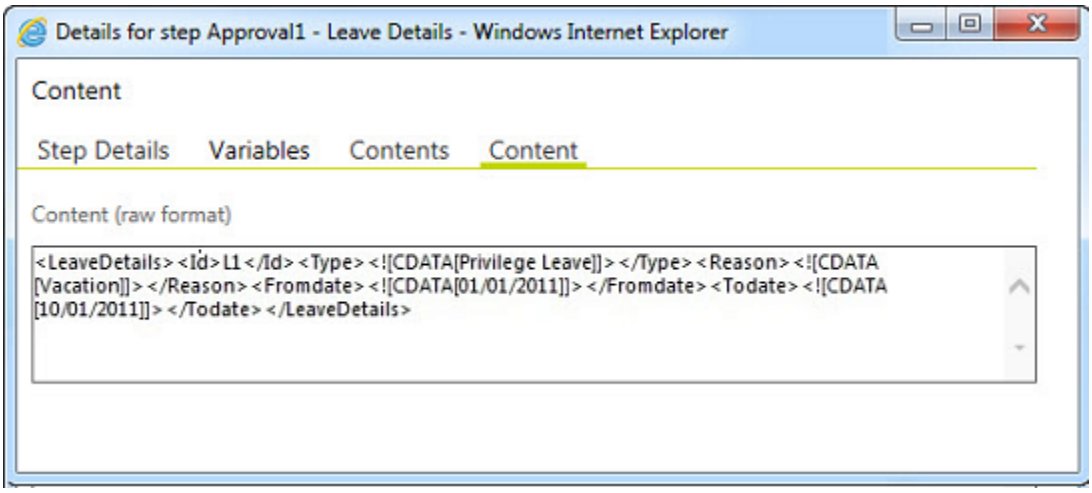


When you click the **Contents** tab, the contents specified in the activity will be displayed. The Contents are the data which can be passed to the workflow while calling Client. Execute API as an XML. They have to be declared in the Start activity along with the Xpath expression.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

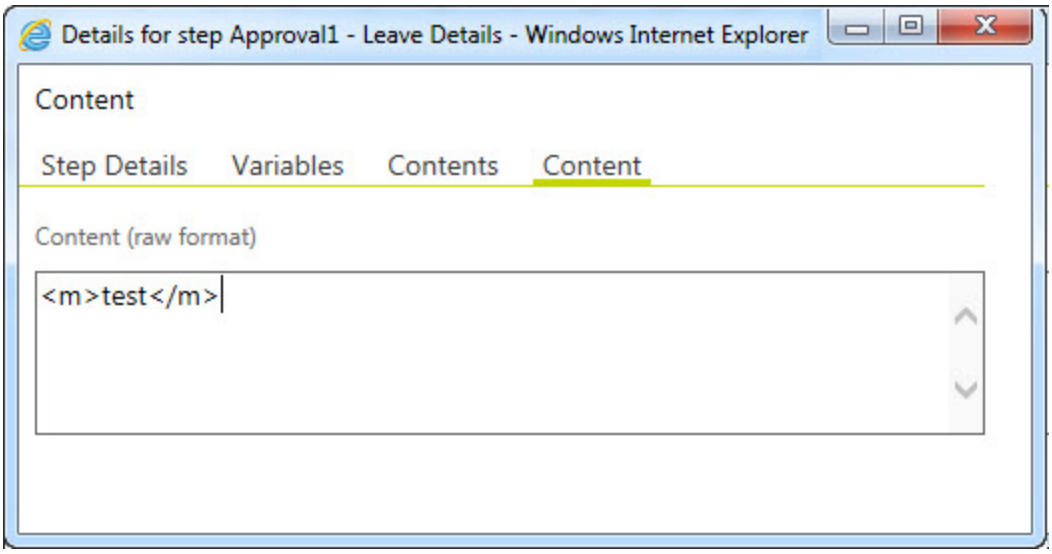


The raw format of the information given in the **Contents** tab will be displayed in the **Content** tab.



When you click the **Content** tab, the data which will be passed to the workflow while calling **Client**. Execute API as an XML will be displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



File Watcher

The **File Watcher** can be quite useful in application integration, by monitoring incoming data files and raising events. It listens to the file system change notifications and raises events when a file(s) in a directory changes. The component can watch files on a local computer or a network drive.

The **File Watcher** raises events when a file is created, renamed, updated or deleted from the specified folder. The File Watcher continuously monitors one or more folders and raises events whenever the above mentioned file activities occur.

The **File Watcher** applications:

1. It can be configured to watch for new files, renames, changes and deletes.
2. Quickly and easily triggers workflows which are associated to the **File Watcher Event**.
3. It can watch multiple files at the same time.
4. Multiple folders can be added to the **File Watcher** list.
5. It can watch folders which present in the local system or in the network path.
6. It raises four events namely **Created**, **Changed**, **Deleted,** and **Renamed**.

See File Watcher Events for more details.

File Watcher Operations

The File Watcher List is the basic component of the File Watcher that contains details of the folders that have to be watched. It has options for maintenance of folders such as adding a new folder, editing an existing folder and deleting a folder.

Security rights are set for adding, editing and deleting a folder. Only the logged in user with rights will be able to add, edit or delete the folder for file monitoring.

To open the File Watcher List

In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **File Watcher**. The **FileWatcher**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

List page appears.

File Watcher Operations

- Add Folder: On the action bar, click **New**. The **Add Folder** wizard appears. Use this option to create a new folder.
- Edit Folder Configuration Settings: On the action bar, click **Edit**. The **Edit** wizard appears. Use this option to set the folder configuration.
- Delete: On the action bar, click **Delete**. A message to confirm the folder deletion appears.

Add Folder

Use the **Add Folder** option to specify a particular folder which has to be watched by "FileWatcher" for various events.

To add a folder

1. Create a folder in any local drive.

   For example, create a folder called FileWatcher in C:\ drive.
   The folder name should not contain the following special characters:
   [#~&+\"'<>|@{}(),;^=!$]
   However, a folder name can contain a space or an underscore.
2. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **File Watcher**.
   The **FileWatcher List** page appears.
3. On the action bar, click **New**. The **Add Folder** wizard appears.
4. In the **Title** box, type a unique title for the folder to be watched. For example, enter the title as 'HR'.

   Note:
   - Ensure that the title is unique for a repository.
   - Ensure that the title does not contain the following special characters:
   [#~&+\"'<>|@{}(),;^=!$]
   However, a title can contain a space or an underscore.
5. In the **Physical Path** box, specify the path to store the folder. For example, type C:\FileWatcher.

   Note:
   - If the physical path is a Network path, then enter the path in the following format: *servername**<shared folder name>*.

   - Ensure that the value specified for **Physical Path** is unique for a data source. If multiple repositories use a single data source, then the same physical path can be used in only one repository.

   - Consider the following scenario where:
   - Multiple data sources point to a single File Watcher folder path.
   - The same path is associated with multiple workflows of different repositories for File Watcher events.
   In this scenario, the File Watcher events will be captured only for one repository, among the multiple data sources available.
6. In the **Buffer Size** box, specify the buffer size. By default, the buffer size is 16000.
7. In **Notify Filter** field, choose either the Maximum or Minimum option. By default, Maximum is chosen.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

8. Choose either **Yes** or **No** for the **Created** option. By default, the **Yes** option is chosen to consume the **Created** event.
9. Choose either **Yes** or **No** for the **Changed** option. By default, the **Yes** option is chosen to consume the **Changed** event.
10. Choose either **Yes** or **No** for the **Deleted** option. By default, the **Yes** option is chosen to consume the **Deleted** event.
11. Choose either **Yes** or **No** for the **Renamed** option. By default, the **Yes** option is chosen to consume the **Renamed** event.
12. Click **Save** to update the changes. A message about successfully saving the new folder appears.
13. Click **OK** to save the new folder.

Edit Folder Configuration Settings

Use the **Edit** option to modify the configuration settings of an existing folder.

To edit a folder

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **File Watcher**.
   The **FileWatcher List** page appears.
2. On the action bar, click **Edit**. The **Edit** wizard appears.
   Alternatively, select and right-click a folder, and then select **Edit**. The **Edit** wizard appears.
   The **Title** and **Physical Path** appears disabled while editing the configuration settings of a folder.
3. Modify the **Buffer Size**, **Notify Filter**, **Created**, **Changed**, **Deleted**, and **Renamed** fields according to the requirement.

Folder Configuration Options	Description
Buffer Size	Sets the size of the internal buffer. The default value is **16000** bytes and must not exceed 64000 bytes. For best performance, use a multiple of 4000 on Intel-based computers. File Watcher keeps track of the file changes, and stores those changes in a buffer and passes them to the APIs. Each file change event can use up to 16 bytes of memory, not including the file name. If there are many changes in a short time, the buffer can overflow. This causes the File Watcher to lose track of the changes in the directory, and it will only provide blank notification. Increasing the size of the buffer can prevent missing file system change events. We recommend you to increase the buffer size to prevent the buffer overflow. Increasing buffer size is expensive because it comes from a non-paged memory that cannot be swapped out to disk. So it is advised to keep the buffer as small as possible. To avoid a buffer overflow, you can use the NotifyFilter property to filter out unwanted

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Folder Configuration Options	Description
	change notifications. **Note:** File watcher internally sets the include subdirectories property to False to filter out unwanted change notifications. See FileSystemWatcher.InternalBufferSize Property for more details.
Notify Filter:	Filters unwanted change notifications in the folder and helps to avoid buffer overflow. The default setting is **Minimum** to get optimum change notifications. The **Minimum** setting includes only created, deleted, and renamed events. Select the **Maximum** option to capture all the events fired which includes created, deleted, renamed, and also the changed events.
Created	Choose the **Yes** option to consume the Created event and the **No** option if you do not want it to be consumed.
Changed	Choose the **Yes** option to consume the Changed event and the **No** option if you do not want it to be consumed.
Deleted	Choose the **Yes** option to consume the Deleted event and the **No** option if you do not want it to be consumed.
Renamed	Choose the **Yes** option to consume the Renamed event and the **No** option if you do not want it to be consumed.

4. Click **Update**. The message "*Folder Configuration updated successfully*. *Please restart the service.*" appears.
5. Click **OK**.
6. Restart the Workflow Engine service to activate the modified settings.

Delete Folder

Use the **Delete** option to delete an existing folder from the folder list.

Note: Ensure that the folder is not associated with any workflow. You will not be able to delete the folder if it is associated with a workflow.

To delete an existing folder

1. In the **Enterprise Console**, click the **Menu** button, click **Settings**, and then click **File Watcher**.
   The **FileWatcher List** page appears.
2. Select the folder to be deleted.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. On the action bar, click **Delete**. For example, select 'HR' with the path C:\FILEWATCHER\ to be deleted. Alternatively, select and right-click a folder, and then select **Delete**.
4. A message to confirm the folder deletion appears.
5. Click **Yes** to delete. The confirmation message about successful deletion of the folder appears.
6. Click **Close**.

Though the 'HR' folder will be present on the C drive, the File Watcher will not watch it for any events.

File Watcher Events

The File Watcher fires events when a file is created, changed, renamed or deleted from the specified folder. Following are the four events:

- Created Event
- Changed Event
- Renamed Event
- Deleted Event

Created Event

The **Created** event gets triggered when the following actions are performed:

- A new file is created in the folder to be monitored.
- A new file is copied to the folder to be monitored.

To trigger a created event

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**. The Workflows page appears.
2. Select the workflow to be associated, and then click **Associate**.
3. Click **FileWatcher** to associate the workflow with a new file watcher event.
4. In the **Event Type** drop-down list, select **Created**.
5. In the **Event Folder** drop-down list, select the folder to be monitored.
6. In the **File Name/Pattern** box, enter a file type. For example, .txt, .doc. Use * for all file patterns. You can also mention a specific file, such as sample.txt.
   If the file pattern is mentioned, then the File Watcher triggers an event only for files that match with the file pattern and ignores all other files that might arrive at the same time. This field is not mandatory. If the file pattern is not mentioned, File Watcher triggers an event for all file patterns.
7. Click the **calendar** icon to select the **Effective From** and **Effective To** dates. The event will be triggered for changes in the folder to be monitored only during this period. These two field are not mandatory.
8. In the **Workflow Version** drop-down list, select version of the workflow which should get triggered for the **Created** event. By default, **Latest** is selected. This field is not mandatory.
9. Click **Attach Event**.
10. Click **OK** in the confirmation dialog box.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example 1:

In the following example, the file type is not specified. So the **Created Event** will be triggered for all file types.

1. Create a new folder in the local drive. For example, create a folder called **FileWatcher** in 'C' drive.
2. Add the folder to the **File Watcher** list through **Add Folder**.
3. Create a workflow called **CreatedEventWF**.
4. Design the **CreatedEventWF** workflow with an Approval activity. Configure the Activity Properties for the same.
5. Publish the workflow.
6. Select **CreatedEventWF** from the Workflows list, and then click **Associate**.
7. Click **FileWatcher** to associate the workflow with a new file watcher event.
8. In the **Event Type** drop-down list, select **Created**.
9. In the **Event Folder** drop-down list, select the monitoring folder 'C:\FileWatcher'.
10. Leave the **File Name/Pattern** box empty.
11. Click the calendar icon to select the **Effective From** and **Effective To** dates if you want the event to be triggered only for a specific period. These two fields are not mandatory.
12. Click **Attach Event**.
13. Click **OK**.
14. Create a new file called sample.txt and save the same to C:\FileWatcher.
15. As soon as a new file is saved (created event) in 'C:\FileWatcher', the 'CreatedEventWF' workflow gets triggered. The approval activity will be assigned to the specified actor.
16. To check the Work items list, click **Inbox**.

Example 2:

In the following example, the file name and type is specified. The Created event will be triggered only for the specified file.

1. Create a new folder in the local drive. For example, create a folder called **FileWatcher** in 'C' drive.
2. Add the folder to the File Watcher list through **Add Folder**.
3. Create a workflow called **CreatedEventWF**.
4. Design the **CreatedEventWF** workflow with an **Approval** activity. Configure the **Activity Properties** for the same.
5. Publish the workflow.
6. Select **CreatedEventWF** from the Workflows list, and then click **Associate**.
7. Click **FileWatcher** to associate the workflow with a new file watcher event.
8. In the **Event Type** drop-down list, select **Created**.
9. In the **Event Folder** drop-down list, select the monitoring folder 'C:\FileWatcher'.
10. In the **File Name/Pattern** text box, enter 'File.txt'.
11. Click **Attach Event**.
12. Click **OK**.
13. Go to C:\FileWatcher and create a new file called 'File.txt'.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

14. As soon as a text file with the name 'File' is created in 'C:\FileWatcher', the workflow gets triggered. The approval activity will be assigned to the specified actor.
15. To check the Work items list, click **Inbox**.

Changed Event

The **Changed** event gets triggered when an existing file is replaced in the folder to be monitored.

To trigger changed event

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**. The **Workflows** page appears.
2. Select the workflow to be associated, and then click **Associate**.
3. Click **FileWatcher** to associate the workflow with a new file watcher event.
4. In the **Event Type** drop-down list, select **Changed**.
5. In the **Event Folder** drop-down list, select the folder to be monitored.
6. In the **File Name/Pattern** box, enter a file type. For example, .txt, .doc. Use * for all file patterns. You can also mention a specific file, such as sample.txt.
   Note: If file pattern is mentioned, the **File Watcher** triggers an event for only those files that match with the file pattern and ignores all other files that might arrive at the same time. This field is not mandatory. If the file pattern is not mentioned, the **File Watcher** triggers an event for all file patterns.
7. Click the calendar icon to select the **Effective From** and **Effective To** dates. The event will be triggered for changes in the folder to be monitored only during this period. These two field are not mandatory.
8. In the **Version** drop-down list, select the version of the workflow which should get triggered for the **Changed** event. By default, Latest is selected. This field is not mandatory.
9. Click **Attach Event**.
10. Click **OK** in the confirmation dialog box.

Example 1:

In this example, the file type is not specified. So the **Changed Event** gets triggered for all file types.

1. Create a new folder in the local drive. For example, create a folder called **FileWatcher** in 'C' drive.
2. Add the folder to the File Watcher list through **Add Folder**.
3. Create a workflow called **ChangedEventWF**.
4. Design the **ChangedEventWF** workflow with an **Approval** activity. Configure the **Activity Properties** for the same.
5. Publish the workflow.
6. Select **ChangedEventWF** from the Workflows list, and then click **Associate**.
7. Click **FileWatcher** to associate the workflow with a new file watcher event.
8. In the **Event Type** drop-down list, select **Changed**.
9. In the **Event Folder** drop-down list, select the folder to be monitored 'C:\FileWatcher'.
10. Click **Attach Event**.
11. Click **OK**.
12. Go to C:\FileWatcher. Copy a file with the same name, that is 'sample.txt'. A file called sample.txt should

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

exist in the folder.

13. As soon as the new file overwrites the existing file in 'C:\FileWatcher', the workflow gets triggered. The approval activity will be assigned to the specified actor.
14. To check the Work items list, click **Inbox**.

Example 2:

In this example, the file name and type is specified. The **Changed Event** gets triggered only for the specified file.

1. Create a new folder in the local drive. For example, create a folder called 'FileWatcher' in 'C' drive.
2. Add the folder to the File Watcher list through **Add Folder**.
3. Create a workflow called '**ChangedEventWF'**.
4. Design the 'ChangedEventWF' workflow with an Approval activity. Configure the Activity Properties for the same.
5. Publish the workflow.
6. Select 'ChangedEventWF' from the Workflows list, and then click **Associate.**
7. Click FileWatcher to associate the workflow with a new file watcher event.
8. In the Event Type drop-down list, select **Changed.**
9. In the Event Folder drop-down list, select the folder to be monitored 'C:\FileWatcher'.
10. In the File Name/Pattern box, enter the file name as '**File.txt'.**
11. Click **Attach Event**.
12. Click **OK.**
13. Go to C:\FileWatcher and overwrite a file with the name 'File' which is of type .txt.
14. As soon as the new file overwrites the existing 'File.txt' in 'C:\FileWatcher', the workflow gets triggered. The approval activity will be assigned to the specified actor.
15. To check the Work items list, click **Inbox**.

Renamed Event

The Renamed event gets triggered when an existing file has been renamed in the folder to be monitored.

To trigger a renamed event

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**. The Workflows page appears.
2. Select the workflow to be associated, and then click **Associate**.
3. Click **FileWatcher** to associate the workflow with a new file watcher event.
4. In the **Event Type** drop-down list, select **Renamed**.
5. In the **Event Folder** drop-down list, select the folder to be monitored.
6. In the **File Name/Pattern** box, enter a file type. For example, .txt, .doc. Use * for all file patterns. You can also mention a specific file, such as sample.txt.
   If file pattern is mentioned, then the File Watcher triggers an event for only those files that matches with the file pattern and ignores all other files that might arrive at the same time. This field is not mandatory. If file

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

pattern is not mentioned, File Watcher triggers for all file patterns.

7. Click the calendar icon to select the **Effective From** and **Effective To** dates. The event will be triggered for changes in the monitoring folder only during this period. These two field are not mandatory.
8. In the **Version** drop-down list, select version of the workflow which should get triggered for the 'Renamed' event. By default, Latest is selected. This field is not mandatory.
9. Click **Attach Event**.
10. Click **OK** in the confirmation dialog box.

Example 1

In this example, the file type is not specified. So the Renamed Event gets triggered for all file types.

1. Create a new folder in the local drive. For example, create a folder called 'FileWatcher' in 'C' drive.
2. Add the folder to the File Watcher list through Add **Folder**.
3. Create a workflow called 'RenamedEventWF'.
4. Design the 'RenamedEventWF' workflow with an Approval activity. Configure the **Activity Properties** for the same.
5. Publish the workflow.
6. Select 'RenamedEventWF' from the Workflows list, and then click **Associate**.
7. Click FileWatcher to associate the workflow with a new file watcher event.
8. In the **Event Type** drop-down list, select Renamed.
9. In the **Event Folder** drop-down list, select the monitoring folder 'C:\FileWatcher'.
10. Leave the **File Name/Pattern** box empty.
11. Click **Attach Event**.
12. Click **OK** in the confirmation dialog box.
13. Go to C:\FileWatcher path and rename a file called 'Sample.txt' to 'Sampletest.txt'.
14. As soon as a existing file is renamed in 'C:\FileWatcher', the workflow gets triggered. The approval activity will be assigned to the specified actor.
15. To check the Work items list, click **Inbox**.

Example 2

In this example, the file name and type is specified. The Renamed Event gets triggered only for the specified file.

1. Create a new folder in the local drive. For example, create a folder called 'FileWatcher' in 'C' drive.
2. Add the folder to the File Watcher list through **Add Folder**.
3. Create a workflow called 'RenamedEventWF'.
4. Design the 'RenamedEventWF' workflow with an Approval activity. Configure the Activity Properties for the same.
5. Publish the workflow.
6. Select 'RenamedEventWF' from the Workflows list, and then click **Associate**.
7. Click FileWatcher to associate the workflow with a new file watcher event.
8. In the **Event Type** drop-down list, select **Renamed**.
9. In the **Event Folder** drop-down list, select the monitoring folder 'C:\FileWatcher'.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

10. In the **File Name/Pattern** box, enter the file name as 'File.txt'.
11. Click **Attach Event**.
12. Click **OK**.
13. Go to C:\FileWatcher path, rename any file to 'File.txt' .
14. As soon as a file is renamed with 'File.txt' in the 'C:\FileWatcher', the workflow gets triggered. The approval activity will be assigned to the specified actor.
15. To check the Work items list, click **Inbox**.

Deleted Event

The Deleted event gets triggered, when a file is deleted from the folder to be monitored.

To trigger the deleted event

1. In the **Enterprise Console**, click the **Menu** button, and then click **Workflows**. The **Workflows** page appears.
2. Select the workflow to be associated, and then click **Associate**.
3. Click **FileWatcher** to associate the workflow with a new file watcher event.
4. In the **Event Type** drop-down list, select **Deleted**.
5. In the **Event Folder** drop-down list, select the folder to be monitored.
6. In the **File Name/Pattern** box, enter a file type. For example, .txt, .doc. Use * for all file patterns. You can also mention a specific file, such as sample.txt.
   If the file pattern is mentioned, then the File Watcher triggers an event only for files that match with the file pattern and ignores all other files that might arrive at the same time. This field is not mandatory. If the file pattern is not mentioned, File Watcher triggers an event for all file patterns.
7. Click the calendar icon to select the **Effective From** and **Effective To** dates. The event will be triggered for changes in the folder to be monitored only during this period. These two field are not mandatory.
8. In the **Workflow Version** drop-down list, select version of the workflow which should get triggered for the 'Deleted' event. By default, Latest is selected. This field is not mandatory.
9. Click **Attach Event**.
10. Click **OK** in the confirmation dialog box.

Example 1

In this example, the file type is not specified. So the **Deleted Event** will be triggered for all file types.

1. Create a new folder in the local drive. For example, create a folder called 'FileWatcher' in 'C' drive.
2. Add the folder to the File Watcher list through Add Folder.
3. Create a workflow called 'DeletedEventWF'.
4. Design the 'DeletedEventWF' workflow with an **Approval** activity. Configure the **Activity Properties** for the same.
5. Publish the workflow.
6. Select '**DeletedEventWF'** from the Workflows list, and then click **Associate**.
7. Click **FileWatcher** to associate the workflow with a new file watcher event.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

8. In the **Event Type** drop-down list, select **Deleted**.
9. In the **Event Folder** drop-down list, select the monitoring folder 'C:\FileWatcher'.
10. Leave the **File Name/Pattern** box empty.
11. Click the calendar icon to select the **Effective From** and **Effective To** dates if you want the event to be triggered only for a specific period. These two fields are not mandatory.
12. Click **Attach Event**.
13. Click **OK**.
14. Go to C:\FileWatcher, delete a file Sample.doc'.

    As soon as the 'Sample.doc' file is deleted in 'C:\FileWatcher', the workflow gets triggered. The approval activity will be assigned to the specified actor.
15. To check the **Work** items list, click **Inbox**.

Example 2

In this example, the file name is specified. The **Deleted** event will be triggered only for the specified file.

1. Create a new folder in the local drive. For example, create a folder called **FileWatcher** in 'C' drive.
2. Add the folder to the File Watcher list through Add **Folder**.
3. Create a workflow called **DeletedEventWF**.
4. Design the **DeletedEventWF** workflow with an **Approval** activity. Configure the **Activity Properties** for the same.
5. Publish the workflow.
6. Select '**DeletedEventWF**' from the Workflows list, and then click **Associate**.
7. Click **FileWatcher** to associate the workflow with a new file watcher event.
8. In the **Event Type** drop-down list, select **Deleted**.
9. In the **Event Folder** drop-down list, select the monitoring folder 'C:\FileWatcher'.
10. In the **File Name/Pattern text** box, enter the file name as 'File.txt'.
11. Click **Attach Event**.
12. Click **OK**.
13. Go to C:\FileWatcher path, delete a file 'File.txt'.

    As soon as a file 'File.txt' is deleted in the 'C:\FileWatcher', the workflow gets triggered. The approval activity will be assigned to the specified actor.
14. To check the Work items list, click **Inbox**.

Limitations of File Watcher

File Watcher has the following limitations:

1. If you add more number of files at the same time, there are chances of missing some events in the added files.
2. When you open a file from the folder to be watched, modify it, and then save it, proper events might not get triggered.
3. To modify the file in the Watching folder:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

a. Create a copy of the file in a different location.
b. Modify the file.
c. Copy and paste the modified file in the monitoring folder.

4. When the same file is copied more than once to the monitoring folder, then File Watcher checks for the last modified date time. If the date time is same, then no event is raised. If the last modified date time is different, then the **Changed Event** is raised. If you copy the same file after sometime, the **Changed Event** is raised.
5. File Watcher does not support desktop files.
6. File Watcher does not support sub-folders. If you create, change, rename, or delete a file in another folder in the folder to be watched, no event gets triggered.
7. Folder names with special characters, other than underscore '_' and space, are not allowed.
8. You can add a network folder to the folders list. To enable to watch the network path, the Workflow Engine service should start with a user account who has network access. Local user account cannot watch the network path.

Frequently Asked Questions

Q1: How to add an FTP folder to the File Watcher?

Ans: File Watcher can be configured to a network path which is mapped to the FTP folder.

For example, consider the network path \\ServerName\Sharedfolder is an FTP folder. You can associate this folder path to the File Watcher, and then can use the FTP folder for file operations. If the FTP site for the share folder is ftp://220.227.90.135/Sharedfolder/, then you can add the network path (UNC format) to the list. When external users perform any file operations on this FTP folder through FTP access, then the workflow gets triggered.

Since FTP access is not direct, authentication is not required as the Folder (Network/Local) access rights are enough.

Q2: Is it possible to give more than one file pattern?

Ans: Only one file pattern can be specified but the user can add multiple associations with different file patterns.

Q3: Are special characters are allowed for folder names?

Ans: Yes. The only special characters you can use in folder names are space and underscore.

Object Access

In a business environment, different softwares and processes are in place for varied business functions. Most of the information is interchange between software and departments. In certain cases, this information is manually incorporated in the software. This process is time consuming and prone to human errors. To make the environment fully automated, these software need to interact with each other.

AVEVA Work Tasks provide this facility through Integration Activities but the user is expected to know the respective area. The inbuilt security feature called SOA (AVEVA Work Tasks Object Access) enables the product to interact with other applications.

SOA allows users to use their existing .net assemblies with AVEVA Work Tasks's built in security features. Only privileged users are authorized to access critical information.

To enable the feature, certain configuration needs to be done to the setup in AVEVA Work Tasks such as SOA

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Folders and SOA Assemblies. Also, one needs to instantiate a class in a workflow to use the methods and properties in workflow activities via the expression editor.

Configuration

SOA configuration consists of SOA repository registration such as SOA Folders and SOA Assemblies. In SOA Folders, users can register a path or folder which consists of assemblies and reference assemblies.

Existing .net assembly methods or properties can be used in AVEVA Work Tasks by following a few steps. First register the Folder in which the assembly resides, and then register the methods or properties using Assemblies. Once the assembly is registered, the class will be available in AVEVA Work Tasks to create an instance. Once the instance is created, the method and properties etc will be available using the expression editor and SOA activities.

In SOA assemblies, users can register an assembly from a registered folder or from System GAC. Registered assembly methods can be accessed in AVEVA Work Tasks, using the following SOA features:

Features

In the SOA feature, you use an existing .NET object directly inside a workflow to specify business logic or to access object properties. These features are light weight and are activated JIT when accessed. They are not serialized and hence do not add up to the workflow footprint and allow using high level Business objects for Decisions and Actions. They are integrated in expressions hence can be used to specify the runtime value of any of the properties in AVEVA Work Tasks.

- **SOA in expression builder:** SOA registered assembly methods can be accessed in the expression builder as normal functions. Users can choose and drop functions into the expression editor and can also supply parameters also from the tree view. The expression editor allows simple scripts as well as function calls. All the activities which have the expression editor interface will have SOA enabled.
- **Others:**
  - An SOA Assembly can be registered multiple times, with different method/parameter selection to use the AVEVA Work Tasks security features, expose methods/properties to different users based on security privileges.
  - SOA Static Objects - All static methods of registered assemblies will be listed to use as instance object.
  - Users can write scripts to make small business rules.
  - Users can even update variables which are defined in workflows.

Enterprise Server

In the Enterprise Server scenario, the assembly and its referenced assemblies that are registered should be on all servers.

Prerequisites

AVEVA Work Tasks has the following prerequisites:

- The registered assemblies should be on Server machines folder.

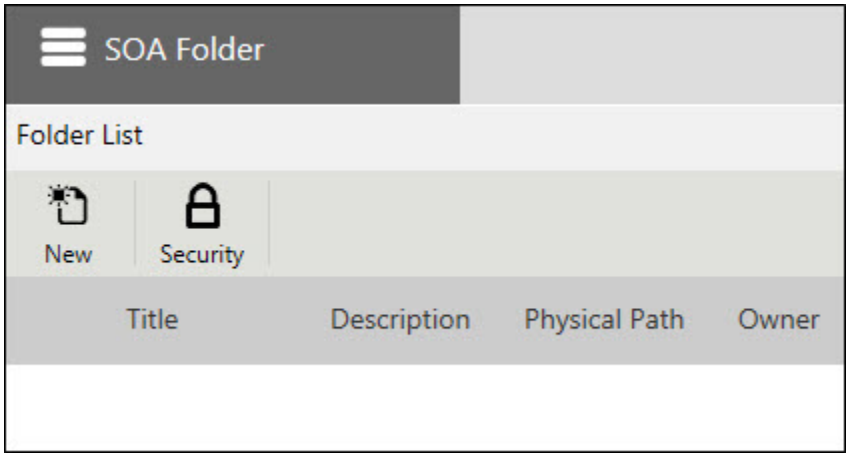
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SOA Folders

SOA Folder is the basic component of SOA configuration and contains the list of folders. Folders may contain assemblies which can be accessed in AVEVA Work Tasks using the expression editor or SOA activities. There are options for maintenance of folders such as adding a new folder, viewing the details of a registered folder, removing or deleting a folder, disabling a folder, enabling a folder, and managing security settings.

To configure SOA folder

- In the **Enterprise Console**, click menu icon, click Settings, and then click **SOA Folder**. The **Folder List** page appears.



- To know about adding a folder, see Adding New SOA Folder Details.
- To know about viewing an existing folder, see Viewing Registered SOA Folder Details.
- To know about deleting a folder, see Deleting an SOA Folder.
- To know about disabling and enabling a folder, see Enabling and Disabling SOA Folder.
- To know about defining security settings at List and ListItem level, see Security Settings for SOA Folder.

Add New SOA Folder Details

To add new SOA folder details

1. On the Action bar, click **New**. The **Folder Configuration** - **New** window appears.
2. In the Title box, type a name for the folder, and in the Description box, type a description for the folder. SOA Folder does not support the following special characters for folder name:

   ~`|!@#$%^&*.,?\":;{}[]()-_+-/\\='
3. In the **Physical Path** box, type the physical path.
   Physical Path provides the actual physical path where the assemblies reside. For example, C:\SOA\Dlls. The physical path must exist on the server. The physical path must exist in all the machines which are part of the Farm and must contain the assemblies which are configured in SOA assemblies.
4. Click **View Assemblies** to verify the assemblies which you want to include.
5. Click **Save** to register the new assembly. You can click **Cancel** to cancel or ignore the entries made.

Note: When you create a folder or item in AVEVA Work Tasks, the Item Status is Published. However, this is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

not applicable for Forms and Workflows folders and items.

View Registered SOA Folder Details

Use the **View** option to view the information about the registered folder.

To view an SOA folder details

1. Select the folder to be viewed.
2. On the Action bar, click **View**. The **Folder Configuration** - **View** window appears.
   Alternatively, select and right-click a folder, and then click **View**.

Delete an SOA Folder

Use the **Delete** option to delete the existing configured folder details.

To delete an SOA folder

1. Select the folder to be removed.
2. On the Action bar, click **Delete**. A confirmation message to delete the folder appears.
   Alternatively, select and right-click a folder, and then click **Delete**.
3. Click **OK** to delete the selected folder. A confirmation dialog box appears.
4. Click **Close** to exit.

Enable and Disable SOA Folder

You can enable and disable an SOA folder.

To enable Folder

1. Select and right-click the disabled folder to be enabled, and then click **Enable**. A confirmation message about successfully enabling the folder appears.
2. Click **OK**.

To disable Folder

1. Select and right-click the folder to be disables, and then click **Disable**. A confirmation message about successfully disabling the folder appears.
2. Click **OK** to disable the selected folder. Disabled folders are shown in faded background.

Note: The **Disable** option acts as a toggle button. Once you disable a folder, this option changes to Enable.

ResourceSet - Subject

Security Settings for SOA Folder

Use the **Security** option to configure security features. You can use this option to define security at List and ListItem level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To configure security for a folder

1. Select the folder in which you would like to assign security at ListItem level.
2. On the action bar, click the **Security** icon to define security at SOA Folder list level. The security settings page for the folder list appears.
   Alternatively, select and right-click a folder, and then click **Security Settings**.
3. Define the security settings at Folder ListItem level.

Assemblies

This section provides information on the different ways of registering, viewing, editing, organizing, and deleting assemblies.

Register Assemblies From GAC

Use the **From GAC** option to list assemblies from the GAC. You can choose assemblies from GAC which you want to use in AVEVA Work Tasks.

To give alias names to classes and assemblies using the From GAC option

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **SOA Assembly**. The **Assembly List** page appears.
2. On the Action bar, click **New**, and then click **From GAC**. The **Select an Assembly** wizard appears.
3. Select an Assembly, and then click **Next** to view details of the selected assembly. The Objects-Accessibility dialog box appears.
4. Select either a Class or a Method from the left pane to display its details.
5. Type an alias for the selected Class or Method in the **Enter Alias For** box.
6. Click **OK** to save the settings.
8. Click **Finish & Publish** to complete the Object selection.

GAC Assembly Fields

- **Assembly:** Lists the available assemblies. You can sort assemblies in ascending or descending order, by clicking the row head.
- **Version:** Displays the list of versions of the particular assembly. You can sort the versions in ascending or descending order by clicking the row head.

Register Assemblies From Assembly Folder

Use the **From Assembly Folder** option to select an assembly from the SOA registered folders. The the SOA registered folders appears under the tree view that shows folders, and the assemblies under each folder as nodes or branches of the tree. You can select all registered folders.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To give alias names to classes and assemblies using the From Assembly Folder option

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **SOA Assembly**. The **Assembly List** page appears.
2. On the Action bar, click **New**, and then click **From Assembly Folder**. The **Select an Assembly** wizard appears.
3. Select an Assembly to register, and then click **Next** for object selection. The Objects Setup dialog box appears.
4. Select either a Class or a Method from the left pane to display its details.
5. Type an alias for the selected Class or Method in the **Enter Alias** For box.
6. Click **OK** to save the settings.
7. Click **Finish & Publish** to complete the Object selection.

Register Assemblies From Web

Use the From Web option to register a web service to use in AVEVA Work Tasks. The registered web service method can be used in the expression editor and in API.

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **SOA Assembly**. The **Assembly List** page appears.
2. On the Action bar, click **New**, and then click **From Web**. The **Specify the Web Uri** wizard appears.
3. Type the required information.

Parameters	Description
Name	Name of the web service.
Web Uri	Valid web URI. For example, http://ws.cdyne.com/psaddress/addresslookup.asmx?wsdl
Description	Description for the Web URI. For example, Web service for accessing address or address validation.
User Name	User name for web service authentication.
Password	Password for the web service authentication.
Domain	Domain for the web service authentication.

4. Click **Test Connection** to validate the URI. If the URI is valid, a test connection succeeded message appears. You can then can save the entry.
5. Click **Save** to save the valid entries.

Create Folder for Registered Objects

Use the **Folder** option to create a folder and register objects in that folder.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To create a folder for registered objects

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **SOA Assembly**. The **Assembly List** page appears.
2. On the Action bar, click **New**, and then click **Folder**. The **New Folder** wizard appears.
3. In the **Title** box, type a name for the folder, and in the **Description** box, type a description for the folder. Folder names must not allowed contain special characters, such as :"\\"'"<>;|*/{}?.

View Registered Assembly Details

Use the **View** option to view the registered objects in the SOA Assemblies. If the selected item is an assembly type, you can view the details of the registered assembly. If it is web type, you can view the details of the web service.

The View option allows the user to view all entries except the password. The password will be hidden.

To view the registered objects

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **SOA Assembly**. The **Assembly List** page appears.
2. Select the folder that you want to view.
3. On the Action bar, click **View**. The **Summary** dialog box appears.
   Alternatively, right-click the selected folder, and then click **View**.

Edit an Object in SOA Assembly

Use the Edit option to edit an existing registered object in the SOA assemblies and SOA web services. In case of SOA assemblies, you can choose different methods or properties based on their requirements.

To edit a registered object

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **SOA Assembly**. The **Assembly List** page appears.
2. Select the folder you want to edit.
3. On the Action bar, click **Edit**. The Object setup dialog box appears.
   Alternatively, right-click the selected folder, and then click **Edit**.
4. Select either a Class or a Method from the left pane to display its details.
5. Enter an alias for the selected Class or Method in the **Enter Alias For** text box.
6. Click **Finish & Publish** to complete the **Object** selection.
7. If the item selected is of web type, you can modify the web service registration details. Use the **Edit** option to edit **URI**, **Description**, **User Name**, **Password**, and **Domain** of a web service entry in the **Assembly List**.

Organize SOA Assembly

Organize an SOA Assembly by performing one of the following:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Moving the SOA Assembly to a folder.
- Making the SOA Assembly a child item of another SOA Assembly.

To organize an SOA Assembly by moving the SOA Assembly to a folder

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **SOA Assembly**. The **Assembly List** page appears.
2. On the Action bar, click **Organize**. The **Organize SOA Assembly** dialog box appears.
3. Drag an SOA Assembly to a folder. The SOA Assembly is moved to the folder.
4. Click **Close**. The **Organize SOA Assembly** dialog box is closed.

To organize an SOA Assembly by making the SOA Assembly a child item of another SOA Assembly

1. In the **Enterprise Console**, click the **Menu** button, click **Tools**, and then click **SOA Assembly**. The **Assembly List** page appears.
2. On the action bar, click **Organize**. The **Organize SOA Assembly** dialog box appears.
3. Drag an SOA Assembly to another SOA Assembly. The SOA Assembly is now a child item of another SOA Assembly.
4. Click **Close**. The **Organize SOA Assembly** dialog box is closed.

Delete a Registered Assembly

Use the **Delete** option to delete a registered assembly. Use this option with caution since it might affect the workflows where the assembly is used or else where. We recommend to use the Disable option instead of the Delete option.

To delete a registered assembly

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **SOA Assembly**. The **Assembly List** page appears.
2. Select the registered assembly to be deleted.
3. On the Action bar, click **Delete**. A confirmation message to delete the registered assembly appears. Alternatively, right-click the registered assembly, and then click **Delete** from the context menu.
4. Click **Yes** to delete the selected assembly. A message about successful deletion of the item appears.
5. Click **Close** to exit.

Manage Security for a Assembly

Use the **Security** option to configure the security feature.

To configure security for an assembly

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **SOA Assembly**. The **Assembly List** page appears.
2. Select an assembly in which you would like to define security at the List and ListItem level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Click **Security**. The security settings page appears. You can define the security rights at Global, Repository, Across List, and List Specific levels.
   Alternatively, right-click the registered assembly, and then click Security Settings from the context menu.

   Example: Assume that the 'Human Resources' folder consists of three classes called 'Company', 'Employee', and 'Leave'. There are three users called 'Allen', 'Kim', and 'Jim'. Allen has Admin rights and has defined the security rights at the ListItem level. Allen gives Kim the rights to view only the Employee class and Jim can view only the Company class.

   This means that only the Admin user is able to view all the classes of the Human Resources folder. If Kim logs in, Kim can view only the Employee class and if Jim logs in, Jim can view only the Company class.

Instantiate an Assembly Class in Workflow

To instantiate a class in a workflow, use the Start activity and SOA Object Instances property. This will provide you the interface to create an instance of a class for further use in a workflow. You can create multiple instances of a class if required.

To instantiate an assembly class

1. Design a workflow.
2. Right-click the Start activity, and then click **Activity Properties**.
3. Click SOA Object Instances to set the property. See SOA Object Instances for more details.

SOA Object Instances in SOA Property

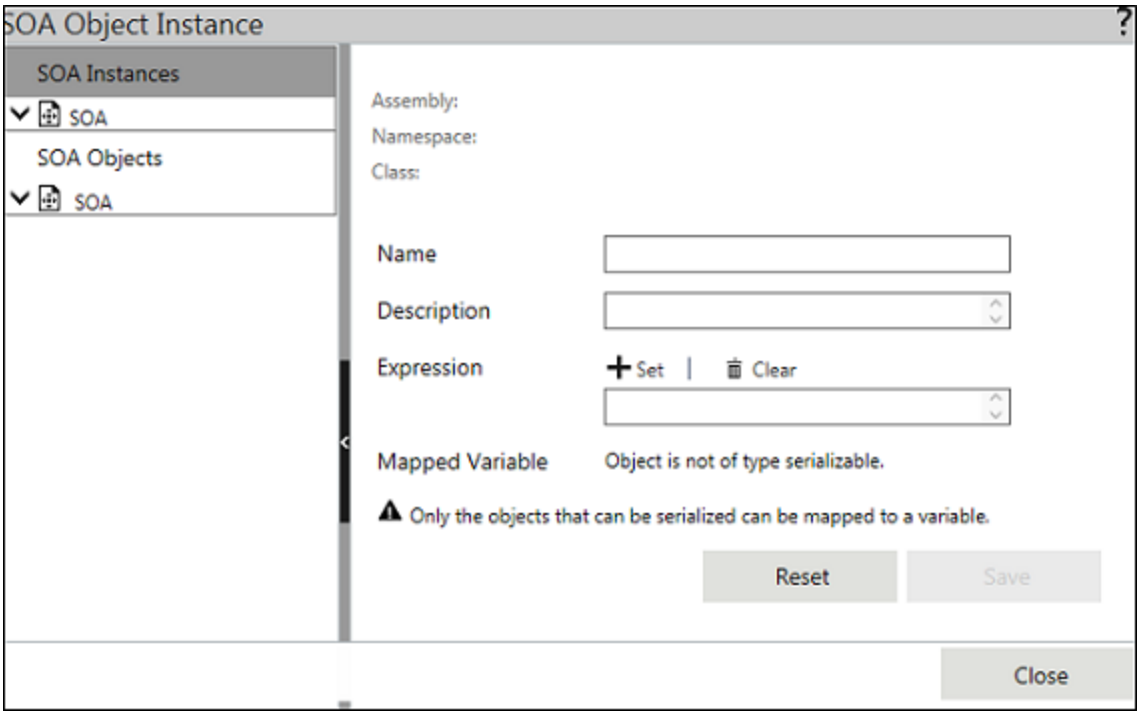
The **SOA Object Instances** property is an important property for SOA. Without starting an instance of a class, you cannot use the methods available in a class that are not static. The static members of the class can be used without creating any instance.

You can create multiple instances of a class. Creating instances of classes sometimes requires parameters. You can use the object instance property editor to select objects, method, or functions for constructor parameters.

You can drag and drop the constructor (.cntr) into the editing area and update the constructor parameters, if any. You can select the parameters from workflow instance properties or from other registered SOA classes. If type casting or conversion are required, then you can opt for functions available under Date, Math, or String in the object browser.

The following image shows the SOA Object Instance property window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The **SOA Object Instance** window contains the following fields:

Fields	**Description**
SOA Instances	Displays the already registered instances and also allows the user to edit or delete an instance.
SOA Objects	Displays the registered objects and allows the user to drag and drop or click and drop a constructor to instantiate a new class.
Object Browser	The Object Browser shows the available AVEVA Work Tasks Objects that can be used as parameters for instantiating a class by selecting them as parameters for the constructor. Users can edit or delete already created constructors from this option.
Reset	Clears the editor and constructor information from the Editor.
Name	A name for the instance.
Description	A descriptive name can be given for the instance of the class.
Expression	Expressions to be set.

SOA Instances

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To instantiate an Assembly class in a workflow

1. From **SOA Objects**, drag and drop or click and drop a constructor to create an instance.
2. Using **Object Browser**, pass the parameters to the constructor.
3. Click **Save**. You can view the saved instances in **SOA Instances**.

To edit an existing instance

1. Click the SOA Instances tab on the left hand side of the SOA Property Editor.
2. Right-click the saved instance which needs to be edited, and then click **Edit** from the context menu.
3. **Update** and **Delete** buttons, which allow the user to update or delete the selected instance, will be visible.

To delete an existing instance

1. Select an instance in the SOA Instances tab.
2. Click **Delete** to delete the instance. This will prompt for confirmation of deletion.
3. Click **OK**.

SOA Instance Members

The SOA property window comes with a new node with the instance name as node text for an already created instance. It shows all the available methods or properties of that instance. Members of each instance can be used to build the expression in the SOA property page. For example, the ‘Employee’ instance has been created here. Members of this instance are displayed in the **Object Browser**.

SOA Static Members

To use static members, no need to create an instance of the class. After an assembly is registered in the SOA Assembly list, all the static members of the registered class get loaded in the SOA Property Editor/Expression editor and can make use the same for passing the parameter.

For example, the ‘Employee’ class has the 'GetCompanyName()' static method. After the class is registered in the SOA Assembly list, 'GetCompanyName()' will be displayed in the **Object Browser** under **SOAStaticObjects**. The user can work with these static methods without creating an instance of Employee class.

Note: Static properties in SOA: SOA, in the normal behavior of having an instance of class, does not support static properties. It works fine with SOA static objects. Example: The assembly ‘Employee.dll’ has a static property called ‘Company Name’ to get the name of the company. If an instance is created for the class ’Employee’ in SOA, the use of static property by utilizing a created instance throws a compilation error in the Expression Editor. Static property needs to be called by using the **type name** where as the static property in **SOAStaticObjects** will work normally.

SOA Objects

All saved instances will be listed in Object **Browser/Workflow Instances/SOA Objects**. The user can make use of these objects and build an expression in the **SOA Property page** or the **Expression Editor**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

SOA Object Instances in Expression Editor

You can create simple expressions with the help of the Expression editor which is a user interface for creating expressions in AVEVA Work Tasks. With SOA integration, the Expression editor comes with a new node that shows all the instances created and methods available under each instance.

You can work with these methods or properties in the expression. With the new feature, even the update variable functionality can be achieved in expression just by choosing the workflow instance and ensuring that the variable name is equal to the method or property.

To use a method, just click on the method in the object browser and drop it in the editor area, if the method has a parameter which can be replaced from the object browser or the workflow instance.

SOA Web Objects in Expression Editor

After a web service is registered, the web methods are accessible in Expression editor and can be used in the same way as the rest of the SOA Objects.

Difference Between Integration and SOA

The following are the differences between integration and SOA:

- Script activity object life cycle is limited to the script. In SOA, the object life cycle is automatically managed by AVEVA Work Tasks and there are activities using JIT.
- Use methods and properties of object without using intermediate variables.
- Using SOA activity is easier than using Script activity.
- SOA is security enabled whereas Script activity security is not enabled.

Security Features for Folders and Assemblies

Security privileges can be set for both folders and assemblies. Assemblies can be registered in AVEVA Work Tasks with different method or property selection and access can be given to different groups.

Limitations of SOA

The SOA has the following limitations:

- Supports only .net assemblies.
- Does not support remote assemblies.
- Does not support call back mechanism.
- Does not support expression as parameter to the method which is used as constructor parameter in the SOA Property Editor.
- Does not support complex expressions, such as an expression inside an expression.
- Does not support unmanaged assembly.
- Does not support 32-bit built dll in a 64-bit machine.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Method with 'Void' as the return type cannot be used as the last statement in the Expression Editor as the last statement expects a return type. Void type can be used above the last statement and the last statement will execute and return the value.

Example

Wrong

```
Int a=10;
Variable.Variable1 = a;
UpdateDatabase();  //Method with Void as return type
```

Correct

```
Int a=10;
Variable.Variable1 = a;
UpdateDatabase();  //Method with Void as return type
a;  //Return some value after the execution of  the method with void as return type.
```

- If Push Notification is enabled in Framework 4.0 environment, SOA is not supported.

SOA Activity

AVEVA Work Tasks provides the **SOA Execute Activity**. Use the **SOA Execute Activity** to directly use SOA in the expression editor. Based on the status, **SOA Execute Activity** returns successful or unsuccessful result.For more information about **SOA Execute Activity**, see *Developer Guide*.

Use SOA Execute Activity in Workflows

SOA Execute is **Object Access** activity supported by AVEVA Work Tasks. The SOA functionality of AVEVA Work Tasks eliminates the need for custom coding to a great extent by allowing stored data within processes to be easily accessed and reused. This activity is used to access a particular method from a registered Assembly. This section contains information about:

- Registering the Assembly in a folder
- Adding the Assembly folder into the SOA Folder
- Adding the Assembly to the SOA Assembly
- Create SOA Object Instance in the workflow
- And finally, displaying the value in an Information activity

Prerequisites

- Download and install the latest AVEVA Work Tasks.
- Create a repository.
- Successfully log on to the repository using administrator credentials.
- Basic knowledge of how to create a custom assembly.
- Basic knowledge of how to create a workflow.

Understanding SOA Execute Activity

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Step 1: Creating a Custom Assembly

1. Open **Microsoft Visual Studio**.
2. In the **File** menu, click **New**, and the select Project from the menu.
   The **New Project** dialog box appears.
3. In the right pane, select **Class Library**.
4. Create a **.net** Class Library.
   Example: In the Name text box, enter '**SOATest**'.
5. Set the path to save the project.
   Example: In the Location drop-down, set the location as C:\SOA using the Browse button.
6. In the **Solution Name** text box, give the directory name.
   By default, it displays the directory name as a specified project name.
7. Click **OK**.

Sample code for creating the class

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace SOATest
{
public class Test
{
public Test()
{

}
public string GetName()
{
return "Welcome to SOA";
}
}
}
```

When you build the project, the dll gets created in the **C:\SOA\SOATest\SOATest\bin\Debug** folder.

Step 2: Adding the Assembly Folder to the SOA Folder

1. Log on to the **Enterprise Console** using administrator credentials.
2. On the Enterprise Console, click **Menu** icon, point to **Settings**, and then click **SOA Folder**.
   The **Folder List** page appears.
3. Click **New** to add a new folder.
4. In the **Folder Configuration** window, enter the Title as "**SOAFolder**" and Description as **"My first SOA**".
5. Specify the Physical Path, that is the path where the folder exists.
6. Give the path and click on the **View Assemblies** button.
   We can now see our dll file listed in the **List of Assemblies**.
7. Click **Save** to save all the information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

8. After you save, you can see the dll listed in the SOA Folder window.

Step 3: Adding the Assembly to SOA Assembly Repository

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **SOA Assembly**.
   You can use any of the following options to add the assembly:
   - From GAC
   - From Assembly Folder
   - From Web
2. Consider adding **From Assembly Folder**.
3. Click **New**, and then click **From Assembly Folder**.
   The **Select an Assembly** window appears.
4. Select the dll file which we had created, that is SOATest.dll, and then click **Next**.
5. After you select the assembly, you can browse through the namespaces, classes, and methods of the assembly. There is an option to give alias name for the selected methods in the next window, that is Object Setup.
6. Select the methods for which you want to give the alias name, else click **Finish & Publish**.
7. The Assembly will get added to the Assembly list. Now create a workflow and implement this SOA Assembly in it.

Step 4: Creating a Workflow

1. If you are familiar with the steps of creating a workflow, you can directly navigate to Step 5.
2. In the **Enterprise Console**, click menu icon, and then click **Workflows**.
   The **Workflows** page appears.
3. On the action bar, click **New**, and then click **Workflow**.
   The **New Workflow** wizard appears.
4. In the **Title** box, type a name for the workflow, and in the **Description** box, type a description for the workflow.
5. Click **Save & Continue**.
   The workflow is created, and the **Workflow Options** wizard appears.
6. Select **Design the Workflow now**, which is the default option, and then click **Finish**.
   The **Process Designer** appears.

Step 5: Creating Object Instance

You have to instantiate an **SOA Object Instance** to use it in the **SOA Execute** activity.

1. Right-click the **Start** activity, and then select **Activity Properties**.
2. Click on the edit icon near the SOA Object Instances property to instantiate a class for the specific purpose.
3. The **SOA Object Instance** window appears. Select the appropriate constructor for the specified class.
4. Expand the tree view for the SOA from the SOA Objects in the left pane.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Expand till Test, and then click on **Actor**. It will display the details in the right pane.
6. Click **Set**.

   It opens a new window which displays the constructor.
7. Click **OK** to save the Constructor.

   The constructor will be saved as shown below:
8. Then click **Save** to save the SOA Instance.

   We now have to display the return value from the method as we specified in the Assembly.
9. To create a variable, click on the **edit** icon near the **Variables** property in the properties of the **Start** activity.
10. Create a new variable of **Type** "String" and click **Update** to save the variable.
11. Click the **Save** button in the Activity Properties pane to save all the data entered.

Step 6: SOA-Execute Activity

To use an SOA, add an **SOA-Execute** activity to the Process Designer.

1. Select **Integration Activities** from the Activities pane on the left.
2. Drag and drop an **SOA-Execute** activity to the Process Designer as shown below.
3. Right-click the SOA-Execute activity to get the Activity Properties pane as shown below.

Execute an SOA Expression

1. The SOA-Execute activity uses the SOA through the Expression Editor.
2. Click on the icon near the **Execute an SOA Expression** to execute the expression.
3. Now we can see the **Expression Editor** where we will build the expression.
4. Expression Editor comes up with a new node which shows all the available methods or properties.
5. We will build a small expression to get the return value from the method.
6. Drag and drop the GetName() method from the Test node to the right hand side pane and build the expression: string result = Test.GetName();
7. Now we have to save the result to some variable. We save it to the variable that we created in the Start Activity, that is the SOAVariable.
8. Drag and drop the SOAVariable from the Variables node in the left pane to the expression on the right hand side and build the expression:

   ```
   string result = Test.GetName();
   ```

   Variable.SOAVariable = result;Click OK to save the expression.
9. Click **OK** to save the expression.
10. Click the **Save** button in the Activity Properties pane to save all the data.
11. We have successfully added the **SOA-Execute Activity**.

Step 7: Display the Return Value of the SOA Method in the Information Activity

1. To add **Information activity**, select Human Activities from the Activities pane on the left.
2. Drag and drop an Information activity to the Process Design area.
3. Right-click the Information activity in the Design area and select Activity Properties from the context menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Now the **Activity Properties** pane appears on the right side using which we need to configure the following:
   - Name of the activity: Specify a Name
   - The Actor to whom this information should be sent: **Select the Actor** to whom this activity will be assigned. For this example, we will send the information to the Actor who submitted the workflow. Click the icon edit in the Assign Actor(s) property. The Assign Actor(s) window pops up. We have to assign the Information task to the user who has alerted the workflow. For this, we have to specify a small expression: Id = Variable Variable.SubmittedBy.ID
     - Select the required provider from the **Providers** drop down.
     - From the first drop down, select **Id.**
     - Keep the operator as default "=".
     - From the next drop down, select **Variable**
     - In the next drop down, select **Variable.SubmittedBy.ID** and click the **Add** button as shown in the following image:
     - Click **Update** to save the property value.
   - The subject of the actual information.
     - Click **edit** near the **Subject** property to enter the subject for the Information which appears in the actor's **Inbox**.
     - To customize the **Subject**, we can use the **Field** controls. Select **Variable** from the first drop down. Select the variable we had declared in the **Start** activity from the second drop down that is, **Variable.SOAVariable** as shown below and click **Append**.
     - You will get the **Subject** line in this format:
     - Click **Update** and then the **Save** button in the Activity Properties pane to save all the data entered.

- Now, we have to link all the activities that is, **Start**, **SOA Execute1**, and **Display SOA Output**.

  Set the output value from the **SOA Execute1 to Display SOA Output** as "**Successfu**l".

Publish the SOAWorkflow

1. To publish the workflow, click the **Publish** option from the menu bar on the top.

   A message box pops up indicating that the workflow has been published successfully.
2. On clicking **Close** you will be redirected to the page showing the Workflows list.
3. Run the published workflow. Select the listed workflow and right-click it.
4. Select the **Publish and Test Run** option from the context menu.

   The **Process** execution view window appears showing the execution status. You will that notice the **Start**, **SOA Execute1 and Display SOA** output activities have been executed successfully.

Note: If the Status is "Execution Pending" in the Process execution window, then click the Refresh button to get the latest status.

You can also view the result value by clicking on any of the activities in the **Process** execution window and then clicking on the **Variables** tab.

View the Inbox

You have to view the information task assigned to the actor who alerted the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. To see the activity in the **Inbox**, select the **Inbox** option from the **Enterprise Console** menu.
2. In the **Inbox** you can find the **Display Web Service Data** activity with the Subject "**Welcome to SOA**".
3. Click on the Information task in the Inbox to open a pop up details view for the task.

   You have successfully created the **SOAWorkflow**.

Security Framework

Security means that the information is protected from unauthorized users. Security also encompasses the concept of integrity, that is, the assurance that information is protected from unauthorized modification or corruption.

In an organization, a security policy is how an organization plans to protect the information from the employees who are categorized based on their designations or roles. The security rights provided to different users depend on the organizational hierarchy.

The need for security is to assign access rights to this hierarchy. If there is no hierarchy in the organization, then there would be no need for a assigning security rights. But most organizations have many hierarchies that have different access rights. The security mechanism in the organization provides restricted access to data. By making access decisions, the security mechanism enforce rules about who can access what.

For example, an administrator and a developer cannot have the same access rights, because obviously, the administrator is superior and has access to most or all of the rights. Similarly, an employee with a higher designation, say a Project Manager, should ideally have more rights as compared to a Team Lead. It is for this very purpose that access rights have to be clearly defined.

AVEVA Work Tasks Security

AVEVA Work Tasks has enterprise rights and security features. The rights and security can be defined for groups and users can be mapped to these groups. The user will be entitled to perform only those tasks for which the right has been given to the respective user.

AVEVA Work Tasks allows effective collaboration during workflow creation and process and allows you to define user-based access rights for workflow design functions. You can define security for the items, forms or folders at the list level and also define security for the forms or lists residing under a folder. It also offers comprehensive security for reporting Work item, Calendar, Queue, and BAM.

The Security Framework allows the administrative user to administer, manage and provide access rights and define roles and responsibilities for the user, based on their individual role or based on the ‘group’ they are part of. It makes managing users across various features of AVEVA Work Tasks much easier and makes applications more secure.

AVEVA Work Tasks Security provides all the features for implementing security at different levels for users and groups and has in-built security groups and rights, which can be extended and customized.

AVEVA Work Tasks Security Features

The following are the features of AVEVA Work Tasks security:

1. AVEVA Work Tasks security has pre-defined security groups. These groups can be used to provide access rights to roles or users. Users can create new security groups if required.
   For example, map a user who belongs to HR to one group and map a user who belongs to Finance to another

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

group. This allows them to view only the items or folders related to their work inside the list.

2. AVEVA Work Tasks security has a set of pre-defined rights. For example, Can Consume Item, Can Manage Item, Can Create Item.
3. There are mainly three levels where security can be applied. They are- Repository Level, List Level and ListItem Level. The List level security allows you to secure the access to various lists available in a AVEVA Work Tasks Repository. The ListItem level security settings allow you to secure the access to various items available in a list.
4. AVEVA Work Tasks security allows the user to secure the Process Designer. This can be done by providing security to its elements as mentioned below:
    a. Provide security for each Category, Action, and Property to ensure that only the authorized user is able to access each of these elements.
    b. Provide security for each menu item to restrict the users for each action on the design. For example: Importing or Exporting workflows, validating and publishing them, settings, running the wizards, testrun.
5. AVEVA Work Tasks security allows you to secure the work items. This can be done by restricting whether a user can forward or kill work items, can set an alternate actor or preferences, can view resource calendar, and queue.
6. AVEVA Work Tasks security allows you to secure the Queue. This can be done by restricting whether a user can view the queue control, add a new queue.
7. AVEVA Work Tasks security allows you to secure the Calendar control: Restricting whether the user can view, add, delete Calendar.
8. The user can exclude options from the Enterprise Console menu. Options like Inbox, Workflows, Reports, Fill Form, Tools (Manage Forms, Manage List and SOA Assembly) and Settings (User, Calendar, Queue) can be excluded.

Example: An organization has 'Manager', 'Designer', 'Technical Consultant', 'Employee', 'Finance team', 'HR team', 'Support team', and 'Other User' roles. These roles having the following responsibilities:

- Manager - Monitor the representatives' reports
- Designer - Design workflows and assign task to others
- Technical Consultant - Create lists and forms
- Employee - Fill forms created by the Technical Consultant and design cases
- Finance team - Fill and view only finance related lists
- Support team - Work on the assigned task
- HR team - Fill and view only HR related lists
- Other User - Create only forms or lists.

In the above case, security can be used to exclude various options from the Enterprise Console menu.

The **Support team** should have access only to the **Inbox** option, so that they can act only on the work items assigned to them.

The **Designer** should have access the **Inbox** and **Workflows** options, so that these users can design workflows, assign the tasks to some other user to act on the assigned work items.

The **Technical Consultant** should have access only **Manage Forms** and **Manage Lists**, so that these users can design

list or forms. For example, 'Expense List', 'Manage List', 'Leave Application List' and 'Travel List'.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Employees** should have access to only the **Fill Form** option, so that these users can fill forms like 'Leave List', 'Expense List' and 'Travel List'.

The **HR team** should have access only to the **Workflows** option, so that these users can view only HR related workflows. For example, 'Employee Details' and 'Leave Application List'.

The **Finance team** should have access only to the **Workflows** option, so that these users can view only Finance related workflows. For example, 'Expense List' and 'Travel List'.

The **Manager** should have access only to the **Reports** option, so that the manager can access reports. This user does not design any process and only monitors the representatives' reports.

Other users should have access only to the **Manage Forms** or **Manage List** option, depending on whether they are responsible for designing the forms or for designing lists.

Introduction to Security Framework

AVEVA Work Tasks introduced the security framework to secure the repository in various ways.

Security framework uses security group, security rights, and security group resource mapping to control the access to features in the repository.

- Security rights: CanCreateItem, CanDeleteItem, and CanDesignWorkflow etc.
- Security groups: Administrator, Contributor, Reader, and Limited Access.
- Security Group Resource Mapping: Security Group Resource Mapping is mapping users or roles to security groups.

Roles can be mapped to Security Groups along with the Priority. Users can also be explicitly mapped to Security Groups.

On logging in to the repository, the user will have the privileges given to the security group to which they belong.

Example:

If the user logs in and belongs to the administrator security group, they will have all the privileges of the administrator security group.

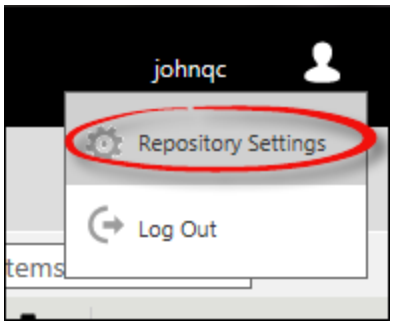
View Security Mapping at Repository Level

You can view mapping between resources and security groups at the Repository level. This displays the resources mapped for the security group.

To view security mapping at repository level

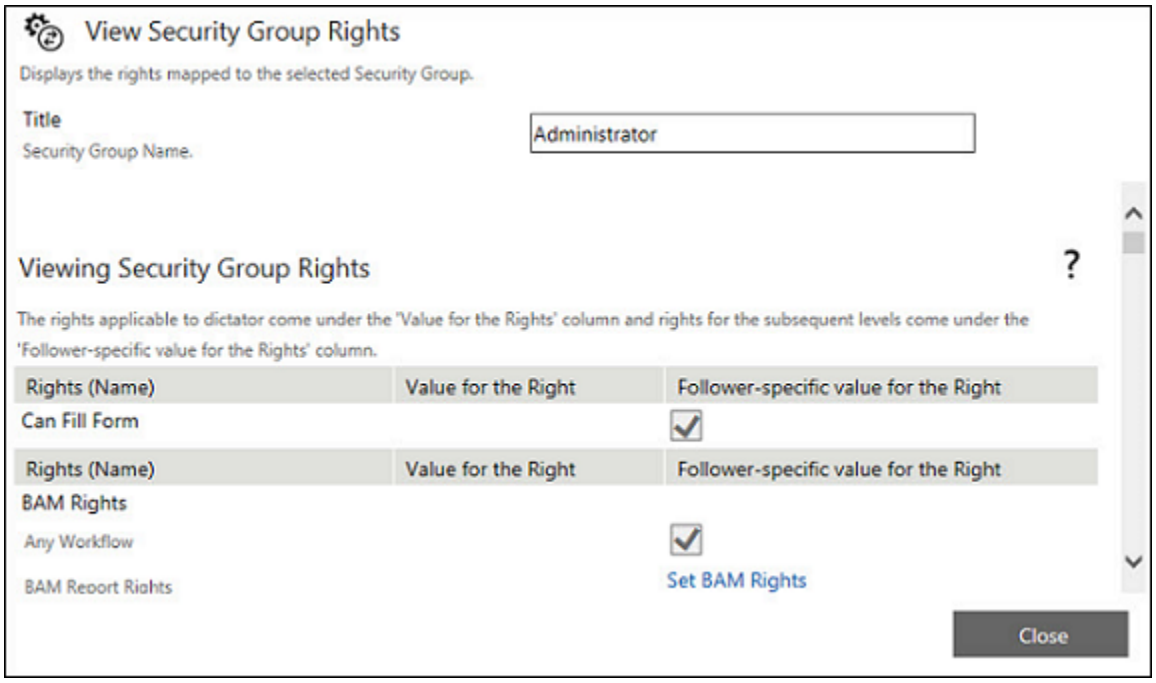
1. Log on to the **Enterprise Console**.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Click the **Standard Settings** tab. The **View Effective Security Rights For User/Role** and **View Security Mapping at Repository Level** options appears.
4. Click **View Security Mapping at Repository Level**. The View Security Mapping at Repository Level page appears. This page displays the User or Role mapped with the security group.

   Click a security group name to view the rights mapped to it. The **View Security Group Rights** page appears.



Components of Security Framework

The security framework contains the following components:

- Security Rights
- Security Group
- Security Resource Mapping

Security Rights

Rights are of vital importance in an application.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Definition

Each feature in the Repository, to which you can control the access, can be defined as right.

Security Rights in AVEVA Work Tasks

There is a set of pre-defined rights in the repository to which you can control the access. These rights define the user privileges at various levels. Apart from the standard rights available in the repository, the user can create custom rights. The user can also create new security rights at the repository level. These rights can be extended and modified at different levels (List and ListItem).

For example: Creating an item inside a list is a feature of a list. You can control access to this feature by defining a right called 'Create Item'. If this right is enabled, the user can create an item inside a list.

Examples of Repository features that can be controlled using Security Rights

- Add, Delete, Edit, Publish, Unpublish, Disable, and Enable items inside a List
- Viewing All Versions, Child Items, Properties, and Associations of an item inside a List
- Saving different Versions and Creating Associations of an item
- Design Workflow, View Workflow, and Perform tasks on an item
- Viewing Others' Draft items, Viewing Pending items, and Viewing Folder
- Settings Security at Repository, List and ListItem levels
- All features of AVEVA Work Tasks Server Controls

For more information about security rights, see:

Security Rights Available for a Repository

Types of Security Rights

Difference between Value for the Rights and Follower Specific Value for Rights

Dictator and Follower

Manage Security Rights

Security Rights Available for a Repository

The following table summarizes the AVEVA Work Tasks standard rights available for a Repository with categorization:

Right Name	Description
BAM Rights	
Workflowscope_Any	This right controls the access to view all workflow reports.
BAM Report Rights	This right controls the access to BAM Report Rights to specify the rights for BAM Tree Menu Rights, Views,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Right Name	Description
	Actions and BAM Configuration.
Can View BAM Control	This right controls the access to view the BAM Report section.
Can View Global Reports	This right controls the access to view the Global Reports option in the drop-down of the BAM Report section.
Workflowscope_InitiatedWorkflow	This right controls the access to view the report of the triggered workflow in the BAM Report section.
Workflowscope_OwnedWorkflow	This right controls the access to view the reports of the workflows that are triggered by another user who is logged in.
Calendar Control Rights	
Can Add Calendar	This right controls the access to add a new calendar.
Can Add Resource Calendar	This right controls the access to add a new resource calendar.
Can Delete Calendar	This right controls the access to delete an existing calendar.
Can View All Calendars	This right controls the access to view all calendars.
Can View Base Calendar	This right controls the access to view the base calendar.
Can View Calendar Control	This right controls the access to view the calendar control.
Generic List Actions	
Can See All Versions	This right controls the access to view all versions of the selected item.
Can Change Owner	This right control the access to the change owner option of the selected item.
Can View Child Items	This right controls the access to view the child items of the selected item.
Can Create Item	This right controls the access to create an item at the list level and to create a child item inside another item at the ListItem level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Right Name	Description
Can Create Folder	This right controls the access to create a new folder.
Can Delete Item	This right controls the access to delete items.
Can Disable Item	This right controls the access to disable items.
Can Edit Item	This right controls the access to edit items.
Can Enable Item	This right controls the access to enable the disabled listitems.
Package Item	This right controls the access to add artifacts to a package template.
Can Publish Item	This right controls the access to publish items.
Can Save As	This right controls the access to save items.
Can Save As Next Major Version	This right controls the access to save the next major version of an item.
Can Save As Next Minor Version	This right controls the access to save the next minor version of an item.
Can Save As Next Version	This right controls the access to save the next version of an item.
Can Save As Specific Version	This right controls the access to save a specific version of an item.
Can TestRun	This right controls the access to view the execution of the published workflow and also the execution view of an item.
Can UnPublish Item	This right controls the access to unpublish the published listitems.
Can View Audit History	This right controls the access to view the audit history of an item. Currently, this is implemented for the workflow list and dynamic lists only.
Can View Item Details	This right controls the access to view the item details in the dynamic list in the read-only mode. This right control is a drop-down list with allow, disallow and owned items only at the dynamic list and list items levels. For the rest of the levels (repository, other list and list item levels), this right control is a check box.
Can View Properties	This right controls the access to view the properties of

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Right Name	Description
	an item.
Can Consume Item	This right controls the access to consume an item (it may be in lookups or any place where list items get consumed).
Can View Disabled Item	This right controls the access to view disabled items.
Can View Item	This right controls the access to view an item.
Can View Others' Draft Folder	This right controls the access to view others' drafted folders.
Can View Others' Draft Item	This right controls the access to view others' draft items. By default, the user can see the items published by others but not their draft items.
Show Only Owned Item	This right controls the access to view only owned listitems. **Note:** This right will take the highest preference. For example, if the logged-in user has some other rights like CanViewPendingItem, CanViewDisabledItem, CanViewOthersDrafItem set to true then in that case, the CanViewPendingItem, CanViewDisabledItem, CanViewOthersDrafItem etc. rights will not be useful for the logged in user because the user can view only the owned listitems.
Show Work In Progress Item	This right controls the access to view work in progress items.
Manage Package List Rights	
Install	This right controls the access to install the package.
Uninstall	This right controls the access to uninstall the package.
Package Template List Rights	
Export	This right controls the access to export the package template.
Generate	This right controls the access to generate the package template.
Mark as Shipped	This right controls the access to mark the package template as shipped.
Queue Control Rights	

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Right Name	Description
Can Add New Queue	This right controls the access to add a new queue.
Can View Queue Control	This right controls the access to view queue control.
Manage Queues	This right controls the access to modify/delete queues.
Web Parts Rights	
Add Web Parts	This right controls the access to add web parts to the home page.
Delete Web Parts	This right controls the access to delete the web parts properties of the home page.
Edit BAM Personalization Scope	This right controls the access to change the personalization scope for the BAM control page.
Edit Calendar Personalization Scope	This right controls the access to change the personalization scope for the Calendar control page.
Edit Home Page Personalization Scope	This right controls the access to change the personalization scope for the logged-in user.
Edit Queue Personalization Scope	This right controls the access to change the personalization scope for the Queue control page.
Edit Web Parts	This right controls the access to edit the web parts properties of the home page.
Edit Personalization Scope	This right controls the access to change the personalization scope for the Work items list control page.
Default HWorkitemhome Page	This right controls the access to specify the default home page for the security group.
Personalize BAM Control	This right controls the access to personalize the BAM control.
Personalize Calendar Control	This right controls the access to personalize the Calendar control.
Personalize Queue Control	This right controls the access to personalize the Queue control
Personalize Work item Control	This right controls the access to personalize the Work items list control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Right Name	Description
Reset User Personalization Scope	This right controls the access to reset the personalization data for “user scope”.
Show Home Page Web Part Settings	This right controls the access to see the home page web parts setting.
Show Ribbon Bar	This right controls the access to make visible/invisible the ribbon bar of the list control web part.
Workflow List Rights	
Can Create Associations	This right controls the access to create associations for list items.
Can Create BAM Workflow	This right controls the access to create BAM Workflow.
Can Create FileWatcher Workflow	This right controls the access to create File Watcher Workflow.
Can Create List Workflow	This right controls the access to create List Workflow.
Can Design Workflow	This right controls the access to design Workflow.
Can Overwrite Published Workflow	This right controls the access to modify (re-design) the published workflow. It is specific to the “Workflow List” and set to false for all security groups by default.
Can Overwrite Published Form	This right controls the access to modify (re-design) the published Form. It is set to false for all security groups by default.
Can Provision Now	This right controls the provisioning feature of workflow.
Can Run Workflow Wizard	This right controls the access to the wizard feature to create workflow.
Can Do Tasks	This right controls the access to tasks like design workflow, start workflow wizard, provision now etc on an item in the workflow list.
Can View Associations	This right controls the access to view the associations of an item in the workflow list.
Can View Workflow Design	This right controls the access to view the process definition (design) of an item in the workflow list.
Can View Exceptions	This right controls the access to view the exceptions of an item in the workflow list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Right Name	Description
Process Designer Rights	This right controls the access to set the action rights and menu rights. See Process Designer Rights for more details.
Work Item Rights	
Can Forward Work Item	This right controls the access to forward a work item to another actor which is assigned.
Can Kill Work Item	This security right is to grand/deny permission to abort workflow instances.
Can Set Alternate Actor	This right controls the access to set an alternate actor.
Can Set Preferences	This right controls the access to set preferences.
Can View Queue	This right controls the access to queue control.
Can View Resource Calendar	This right controls the access to view resource calendar.
Can View Work Item Control	This right controls the access to view the work item control itself.
AVEVA Work Tasks Pro App	
Default Home Page	This right controls the access to specify the default home page for the security group in the AVEVA Work Tasks Pro app. **Note:** Ensure that you log off from and log in to the Work Tasks Pro app after you make any changes in the dashboard form or its configuration.
Other Rights	
Can Fill Form	This right controls to access the form to fill.
Can Access List	This right can be used to control the access to lists that are available in the repository.
Can Manage Security	This right controls the access to define security. If the Can Manage Security right is set to false then the Repository Settings option is not displayed at the right corner.

See Types of Security Rights for more information on security rights.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Types of Security Rights

There are three types of Security rights. These security rights have to be added for each action or functionality which requires restricted access.

- **Allow/Disallow** - If the Allow option is selected, the users belonging to the particular security group will be able to perform the action. If the Disallow option is selected, the users will not be able to perform the action. See Example for Allow or Disallow for more details.
- **Requires Approval** - If the Requires Approval option is selected, the users belonging to a particular security group need approval from their manager to perform that action. For sending an approval, it triggers the workflow that is specified as its rights value. The action will be performed after the request is approved. See Example for Requires Approval and Requires Approval With Prompt for more details.
- **Requires Approval With Prompt** - If the Requires Approval With Prompt option is selected, the users belonging to a particular security group need an approval from their manager. For sending an approval, it will prompt before triggering the workflow that is specified. The action will be performed after the request is approved. See Example for Requires Approval and Requires Approval With Prompt for more details.

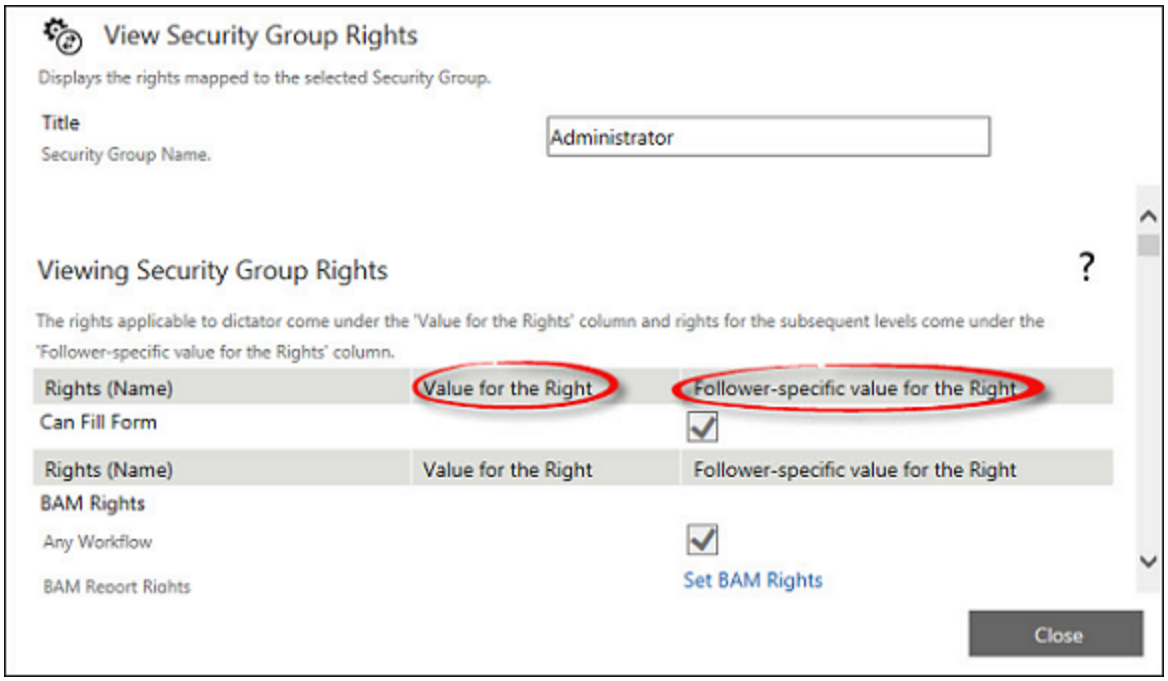
Difference between Value for the Rights and Follower Specific Value for Rights

Every Security Right has two properties that need to be set. These are **Value** and **Follower Value**.

- **Value** - This is the value that you want to set for the Security Right. This value is applicable for the right to which you set it for, and is applied for the current level.
- **Follower Value** - This is the value that you want the right to hold for the follower. If there is no security defined at a particular entity level then that entity will inherit Follower Value from its immediate dictator. First a check is be done to view if the current level has a security definition. If there is no security defined at this level, then the security defined for the immediate dictator will be considered. See Dictator and Follower for more information.

  Even if the child items are granted permission, it will not be accessible unless its parent has the required permission. For example, if a user does not have permission for a folder then user will not have access to the items under that.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Example:

Let us assume that the 'HR Process', 'Leave Process' and 'Holiday Process' are items inside Workflow Lists. 'Leave Process' and 'Holiday Process' are child items of 'HR Process'.

Assume that there is an explicit ListItem level security is applied for 'HR Process' and 'Holiday Process' and there is no explicit ListItem level security applied for 'Leave Process'.

The 'HR Process' and 'Holiday Process' will use the rights specified to apply security at their levels. Since no security is set for 'Leave Process', it will inherit the security defined for its parent item. Here, 'Leave Process' inherits the security settings from 'HR Process'. Hence, the follower value rights that are defined at 'HR Process' will be applied to 'Leave Process'.

The 'Leave Process' is called the follower of the 'HR Process' (in security context) and 'Holiday Process' is called a **child** of the 'HR Process', not its follower.

Dictator and Follower

Dictator - It is the level (Repository, List, or ListItem) where the security is defined. If a list or list item does not have security defined then the security applied at the parent (Repository) level will be implemented. Here, the Repository is a dictator to list and listitem when list and listitem do not have their own security.

Follower - It is the level (List or ListItem) where the security is not defined. It cannot apply its own security and therefore has to inherit from its immediate dictator, that is, its immediate parent level where the security is defined.

- **Example 1**: Security is available at the Repository level. The security is defined for some lists like Workflow list and Forms list. When the logged-in user comes to the Workflow or Forms list, the list level security gets applied. When the user comes to the User list or some other list where no security definition is available at the list level, security is inherited from the above level i.e. the Repository level.
- **Example 2**: At Workflow list, the security is defined but it is not defined at the list item level. In this scenario,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

when the logged-in user comes to the Workflow list, the security rights available at the Workflow list will be applied for the user. When the user is at the list item level, there is no security definition. Hence, the listitem will inherit security from its dictator (in this case, the list).

- **Example 3**: Security is defined at the listitem level but the child items do not have a security definition. When the logged-in user comes to the listitem level, the security definition available at this level gets applied for the user. When the user comes to the child items where security is not defined, the security definition will be inherited from its immediate dictator; here it is the parent item.

Security Group

Security Definition

The set of rights defined for a group is termed as Security Definition.

Security Group

A **Security Group** is a combination of rights along with their values and used to privilege or restrict the users.

AVEVA Work Tasks Security Groups

AVEVA Work Tasks security feature comes with pre-defined security groups that can be mapped to users as well as to roles at different levels. Once a resource is mapped to the security group, the resource will have the privileges and restrictions which are set in the security group in the form of rights. The definition of security groups can be customized at different levels.

User can also create new security groups at the Repository level that can be extended and modified at different levels (List and ListItem). Security groups will be created with different combinations of rights set.

AVEVA Work Tasks has come up with the following four standard security groups which can be extended and modified to various level, but cannot be deleted or renamed. These security groups shipped along with the product are:

- Administrator
- Contributor
- Limited Access
- Reader

The Administrator group has the highest privilege of security rights followed by the Contributor and Reader groups. The Limited Access group has the least privilege rights, with minimum security rights.

The security groups are created with a set of permissions assigned for entities like List, List Item, Application, Menu etc. Different security groups will be created for different combinations of permission sets.

While creating a security group, all the global permission sets will be defined. If any permission sets are required at the list level, they will also be defined here. For example, the Association permission set for the process list.

Example: Security groups will be created at the Repository level but their definition can be modified at List or List Item level. So for an ‘Administrator’ security group which has the “Can Delete List item” right enabled at the Repository level, this delete right can be disabled at the List or List item level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Security groups and their permissions

- Administrator - This group will have view, add, modify and delete rights to the entities.
- Reader - This group will have only view rights to the entities.
- Contributor - This group will have view and add rights to the entities.
- Limited Access - This group will have limited rights to the entities.

Note: The Create item security right, if assigned to a custom group, does not work for Workflow and Forms.

Manage Security Groups

Security Group is a list which lets the user create a new security group and modify, view or delete an existing security group. It also lets the user view the effective mapping for a particular security group. Effective mapping displays the information about the security group and its mappings, shows where all the security group is mapped to the resource and modified. While adding a new security group, value for the rights can be set.

To open the Manage Security Groups window

1. Log on to the **Enterprise Console**.
2. Click **Repository Settings**. The **Repository Settings** page appears.
3. Click **Advanced Settings** tab.
4. Click **Manage Security Groups**. The **Security Groups** page appears.

See Also:

- Creating Security Group
- Modifying Security Group
- Delete Resource Mapping
- Viewing Security Group
- Effective Mapping

Create Security Group

Create Security Group

Use the **New** option to create custom security groups with the required permutations and combinations, apart from the existing security groups. A security group can be created only at the repository level.

Note: To set the Default Home page for the security group, enable the permissions for **Can Fill Form** and **Can Consume Item**.

To create a new security group

1. Log on to the **Enterprise Console**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Click **Repository Settings**. The **Repository Settings** page appears.
3. Click **Advanced Settings** tab.
4. Click **Manage Security Groups**. The **Security Groups** page appears.
5. On the action bar, click **New**. The **Create Security Group** window appears.
6. Select the desired option.
   - New Security Group: Select **New Security Group** to create a security group from scratch. This screen displays the rights with default values. The user can edit them and create a new security group altogether.
   - Copy From Existing Security Group: Select **Copy From Existing Security Group** to create a security group by coping the rights from an existing security group. The user can customize this group later.

To create a new security group

1. Select **New Security Group**, and then click **Continue**.
2. In the **Title** box, type a name for the security group, and in the **Description** box, type a description for the security group.
3. In the **Viewing Security Group Rights** section, set the rights for the new security group.
4. Click **Save** to assign rights to the security group. A message about successfully creating the security group appears.
5. Click **OK** to save the security group.

To copy from existing security group

1. Select **Copy From Existing Security Group**.
2. Select a security group from the drop-down list, and the click **Continue**. The new security group will have the set of rights and predefined values of the selected security group.

   Note: This will inherit the existing security group from the repository level only and not from other levels (e.g. List and List Item level).
3. In the **Title** box, type a name for the security group, and in the **Description** box, type a description for the security group.
4. In the **Viewing Security Group Rights** section, modify the **Value for the Rights** and **Follower-specific value for the Rights** according to the requirements.
5. Click **Save** to assign rights to the security group. A message about successfully creating the security group appears.
6. Click **OK** to save the security group.

Note: The *LastUpdatedDateTime* column in the Security Group list displays the modified date and time of a particular security group, only when it is updated at the Repository level.

To view security mapping, see View Security Group.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Modify Security Group

Modify Security Group

Use the Edit option to edit the existing settings of a security group. You can also synchronize the edited settings.

To edit a security group

1. Log on to the **Enterprise Console**.
2. Click **Repository Settings**. The **Repository Settings** page appears.
3. Click **Advanced Settings** tab.
4. Click **Manage Security Groups**. The **Security Groups** page appears.
5. Select the security group to be modified, and then click **Edit**. The **Modify a Security Group** window appears. Alternatively, select and right-click the **Security Group**, and then click **Edit**.
6. In the **Viewing Security Group Rights** section, modify the security rights.
7. Click **Save**. The **Synchronizing the Security Group** window appears.
8. Click **Yes** to synchronize the edited settings, else click **No**.
9. Click **Submit**. The *Security Group is modified successfully* message appears.
10. Click **OK**.

Note: The **LastUpdatedDateTime** column in the Security Group list displays the modified date and time of a particular security group, only when it is updated at the Repository level.

Example for Allow or Disallow

Example for Allow or Disallow

Consider a scenario where a user called 'Rose' belongs to the 'Reader' security group. The 'Can Delete Item' right is initially set as 'Disallow' for 'Reader'. The example shows how to set this right to 'Allow' so that this user will be able to delete the Item.

To set the right to Allow

1. Log on to the **Enterprise Console** as 'Rose'.
2. Select Workflows.
3. Select any workflow, and click **Delete**. The following warning message gets displayed:

   You do not have rights to perform this action.
4. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. You now have to edit the security group definition of the 'Reader' to allow deleting an item.
6. To edit the Reader security group, log on as **Administrator**.
7. Click **Repository Settings** (top-right corner).
8. Click the **Advanced Settings** tab.
9. Click **Manage Security Groups**. The **Security Group** page is displayed.
10. Select **Reader**, and click **Edit**. In the **Generic List Actions** rights (name), the **'Can Delete Item'** is selected as **Disallow**.
11. Select **Allow**.
12. Click **Save**. The **Synchronization** screen is displayed. Click **Yes** to synchronize the security groups to the below levels. Click **OK** in the confirmation window.
13. Log on as 'Rose'.
14. Select **Workflows** from the **Enterprise Console** menu.
15. Select any workflow and click **Delete**.
16. Click **Yes** and then click **Close** to delete the item successfully.

Example for Requires Approval and Requires Approval With Prompt

Example for Requires Approval and Requires Approval With Prompt

Requires Approval

Consider the Can Delete Item right - If a user belonging to a particular group deletes an item, it will trigger the security approval workflow and only on the approval of the security workflow, the item will get deleted. If the **security approval workflow** is rejected then the item will not get deleted.

Requires Approval With Prompt

Consider the Can Delete Item right - If a user belonging to a particular group deletes an item, the user will be prompted that an approval workflow will be triggered to perform the particular action and the item will get deleted only on the approval of the security workflow. If the security approval workflow is rejected then the item will not get deleted.

Example: On the List level, John who is a manager, initiates a approval process for deletion of an item in the workflow list. John has to enable the approval workflow. The manager must import the security approval workflow from AVEVA\WorkTasks\security_approval_workflow.xml and the select the user for the approval activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To set up Security Approval Workflow

1. Set up the security approval workflow. See Set up Security Approval Workflow to setup.
2. Right-click the **Request for Approval** activity and select **Activity Properties**. Set the **Assign Actor(s)** property. In this example, the administrator user is set as a actor.
3. In the **Publish** menu, select Publish to publish this workflow.
4. Click **Repository Settings > Advanced Settings > Manage Security Groups**.
5. Select the **Contributor** security group and click **Edit** to edit the security group.
6. For **Can Delete Item**, select **Requires Approval With Prompt** from the drop-down list.
7. Open the **Workflow Lookup** screen. Select the **Workflow** and **Version**.
8. Click **Select**.
9. For the **Can Publish Item** right, select **Requires Approval** from the drop-down list.
10. Open **Workflow Lookup** screen. Select the **Workflow** and **Version**.
11. Click **Select**. The Modify a Security Group screen is displayed.
12. Click **Save**.
13. Click **Yes** to synchronize the security groups to the below levels.
14. Login into the **Enterprise Console** as a user who has Contributor rights. Here, George has Contributor rights.
15. Select any item and click **Publish** to publish this item. For example, **DraftFolder**.
16. Click **Close** and then click **Publish**. The message shown below is displayed. Click **OK**.
    The security approval workflow that is attached to the item is pending, you cannot perform this action now.
17. Login as the administrator because the assigned actor in the security approval workflow is the administrator.
18. Click **Inbox**. The Work items list comes to the administrator's Inbox. If the administrator approves, a user who has administrator rights will be able to publish the item. If the administrator rejects, the contributor will not be able to publish the **DraftFolder** item.
19. Login as George.
20. Select HR Process workflow and click **Delete**. When the user belonging to the Security Group tries to delete an Item in any of the Lists, for example an item in the Workflows list, the Security workflow will be initialized. The user will be prompted with a message and the action will result as per the workflow result. The confirmation message is displayed. Click **OK**.
21. Click **Close**.
22. Login as an administrator user because the assigned actor in security approval workflow is 'Administrator'.
23. Click **Inbox**. The Work items list comes to the administrator user's Inbox. If the administrator approves, a user with administrator rights will be able to delete the item. If the administrator user rejects, the contributor will not be able to delete the **Holiday Process** workflow item.

View Security Group

View Security Group

Use the View option to view the security group rights for the selected security group.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To view the security group

1. Log on to the **Enterprise Console**.
2. Click **Repository Settings**. The **Repository Settings** page appears.
3. Click **Advanced Settings** tab.
4. Click **Manage Security Groups**. The **Security Groups** page appears.
5. Select the security group to be viewed, and then click **View**. The **View Security Group Rights** window appears.
   Alternatively, select and right-click the security group, and then click **View**.
6. Click **Close**.

Delete Security Group

Delete Security Group

Use the **Delete** option to delete the selected security group. After you delete a security group, it is removed from all the places where the security definitions are created. In addition, the users mapped to these deleted security groups will have only limited access.

To delete a security group

1. Log on to the **Enterprise Console**.
2. Click **Repository Settings**. The **Repository Settings** page appears.
3. Click **Advanced Settings** tab.
4. Click **Manage Security Groups**. The **Security Groups** page appears.
5. Select the security group to be deleted, and then click **Delete**. The **Delete Security Group** dialog box appears.
   Alternatively, select and right-click the security group, and then click **Delete**.
6. Click **Yes** to delete the security group, else click **No**. The *Security Group is deleted successfully* message appears.
7. Click **OK**.

You cannot delete the default security groups of **AVEVA Work Tasks**. You can delete only the user-defined security groups.

Effective Mapping of a Security Group

Effective Mapping of a Security Group

Use the Effective Mapping option to view the resources mapped for the security groups at different levels.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To view effective mapping

1. Log on to the **Enterprise Console**.
2. Click **Repository Settings**. The **Repository Settings** page appears.
3. Click **Advanced Settings** tab.
4. Click **Manage Security Groups**. The **Security Groups** page appears.
5. Select a security group and then click **Effective Mapping**. The **View Effective Mapping** window appears. Alternatively, select and right-click a security group, and then click **Effective Mapping**.
6. Click a **List/Item Name** to view the rights mapped to the selected **Security Group** at the repository level. The **View Security Group Rights** window appears.
7. Click **Close**.

To view the effective mapping at list level

1. To view the effective mapping at list level, enable security at the list level. If there is no security at this level, first enable it. For more information about enabling security at the list level, see Enabling Security at List Level.
2. Go to any list using the menu icon in the **Enterprise Console**. For example, assume that security is already enabled at the Workflow list level. In the Enterprise Console, click menu icon, and then click Workflows. The Workflows page appears.
3. On the action bar, click **Security**. The security settings page appears.
4. Click **View Security Mapping at List Level** to view security mapping at the workflow list level. The **View Security Mapping at List Level** appears.
5. Click any security group to display the rights mapped to the group.

To view the effective mapping at listitem level

1. To view effective mapping at listitem level, enable security at the listitem level. If there is no security at this level, first enable it. For more information about enabling security at the list level, see Enabling Security at ListItem Level.
2. Go to any list using the menu icon in the **Enterprise Console**. For example, assume that security is already enabled at the Workflow list level. In the Enterprise Console, click the Menu icon, and then click **Workflows**. The Workflows page appears.
3. Right-click the required workflow list item, and then select **Security Settings** from the context menu. For example, right-click the 'FileWatcher' workflow and then select **Security Settings** from the context menu. The security settings page appears.
4. Click **View Security Mapping at List Item Level** to view the security mapping at workflow listitem level.

   The **View Security Mapping at ListItem Level** appears.
5. Click any security group to display the rights mapped to the group.

Security Resource Mapping

Resource Mapping are pre-defined groups with their own set of rights that are applicable to the users, who are a

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

part of these groups. Resources can be added to the groups either individually or added based on their roles.

Mapping users or roles in the security groups defined for that repository is known as security resource mapping.

A repository can be mapped to any of the following Resource Provider:

- Active Directory
- Forms
- ASP.NET membership
- List

Default Resource Mapping to Security Groups

- Administrator Mapping - Admin should be configured while creating repository.
- Default Mapping - By default, the Admin is mapped to Administrator security group and all created users are mapped to Limited Access security group.
- If the resource mapping does not exist for a user then that user will be a part of the 'Everyone' role.
- By default, the 'Everyone' role is mapped to Limited Access security group. If resource mapping does not exist for a user then the rights defined for limited access will be applied to that user.
- Role(s)/Individual user(s) can be mapped to different Security Groups at the Global, List or List Item level. The users will have access rights to the entities as defined in the Security Group.
- While mapping Roles to Security Groups, there is an option to set a priority number. In case a user is part of multiple roles and these roles are mapped to different Security Groups, then the security will take effect based on the priority number. The Role mapping with the least priority number to the security group will the effective security group for the user.
- The mapping defined at the lower level will take precedence over any mapping defined above it. Hence, a mapping done at the List level will override any mapping done at the Repository level. Similarly, mapping done at the List item level will override any mapping done at the List or Repository level.

Manage Security Group - Resource Mapping

Security Groups are mapped to different AVEVA Work Tasks Role(s) or individual user(s) at the Global, List or List Item level. The users will have access rights to the entities as defined in the Security Group.

Security Groups allow the user to set priority if the user has multiple roles and if these roles are mapped to different security groups. While defining the Security Group to Role mapping, there is an option to set a priority number. If the user is a part of multiple roles, which are mapped to different Security Groups, then the security will take effect based on the lowest priority number given to the roles.

To open the Security Group Mapping:

1. Log on to the **Enterprise Console**.
2. Click **Repository Settings**. The **Repository Settings** page appears.
3. Click **Advanced Settings** tab.
4. Click **Manage Security Group - Resource Mapping**. The **Security Group Mapping** page appears.

See Create a Resource Mapping, Edit Resource Mapping, Delete Resource Mapping to add, modify, or delete a

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

security group.

Create a Resource Mapping

Create a Resource Mapping

Use the **New** option to map roles, map users, or map specific users from a role to different security groups. If you map a role to a security group, then all the users of that role will be mapped to the assigned Security Group. You can also map specific users from a role to a security group.

Mapping Roles: This allows the user to map a role to a specific security group.

To map role with a security group:

1. In the **Security Group Mapping** page, click **New** on the action bar. The **Security Group Resource Mapping** page appears.
2. Click the **Map Roles** tab to map the selected role with a security group.
3. In the **Role Lookup** field, enter the role to map the selected role with a security group. Use the Role Lookup window to search for the required role.
4. In the **Security Group** drop-down list, select the security group to map with the selected role. For example, select 'Contributor' from the Security Group drop-down list.
5. In the **Priority** text box, enter the priority. For example, enter the priority as '4'.
6. Click **Save** to map the resource. The message *Resource mapping is saved successfully appears*.
7. Click **OK**. The selected role is mapped with the security group, and all the rights of the group are applied.

Note: If the logged-in user belongs to multiple roles, the least priority will be considered

Mapping Users: This allows the user to map an individual user to a specified security group.

To map users with a security group

1. In the **Security Group Mapping** page, click **New** on the action bar. The **Security Group Resource Mapping** page appears.
2. Click the **Map Users** tab to map the user with a security group. The **Mapping : User to a Security Group** page appears.
3. In the **User Lookup** field, enter the user to map the selected user with a security group. Use the **User Lookup** window to search for the required role.
4. In the **Security Group** drop-down list, select the security group to map with the selected user. For example, select 'Limited Access'.
5. Click **Save** to map the resource. The message Resource mapping is saved successfully appears.
6. Click **OK**. The selected user is mapped with the security group, and all the rights of the group are applied.

Note:
- The same user cannot be mapped with multiple security groups at the same level.
- The users with special characters **^ ( ) ' <** cannot be mapped with the security group.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Mapping Users from Role: This allows the user to map specific users from a role to a specified security group.

To map users from a role with a security group

1. In the **Security Group Mapping** page, click **New** on the action bar. The **Security Group Resource Mapping** page appears.
2. Click the **Map Users from Role** tab to map the user from a specific role with a security group. The **Mapping : User from Role to a Security Group** page appears.
3. In the **Role Lookup** field, select the role from which the users need to be mapped with a security group and then click **List Users**.
4. In the the **Select Users from Role** list box, select the users to be mapped with the security group.
5. In the **Security Group** drop-down list, select the security group to map with the selected user. For example, select 'Limited Access'.
6. Click **Save** to map the resource. The message Resource mapping is saved successfully appears.
7. Click **OK**. The selected user is mapped with the security group, and all the rights of the group are applied.

Note:
- The same user cannot be mapped with multiple security groups at the same level.
- The users with special characters **^ ( ) ' <** cannot be mapped with the security group.

Edit Resource Mapping

Edit Resource Mapping

Use the **Edit** option to edit security group mapping settings for the individual user or a specified role.

To modify a security group:

1. In the **Security Group Mapping** page, select the security group name to be modified. For example, select the 'Limited Access' security group, and the user name 'John'.
2. On the action bar, click **Edit**. The **Mapping : User to a Security Group** page appears.
3. Change the required parameter, such as **User Name**, **Security Group**, or the **Provider Name**.
4. Click **Save** to edit the resource. The message *Resource mapping is saved successfully* appears.
5. Click **OK**. The selected user is mapped with the security group, and all the rights of the group are applied.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Delete Resource Mapping

Delete Resource Mapping

Use the **Delete** option to delete the resource mapping for the selected user or role.

To delete a security group:

1. In the **Security Group Mapping** page, select the security group name that you want to delete.
2. On the action bar, click **Delete**. A confirmation message to delete the security group appears.
3. Click **OK** to delete, else click **Cancel** to **cancel** the operation. The confirmation message about successfully deleting the resource mapping appears.
4. Click **OK**

View Effective Security Rights for User or Role

You can use the **View Effective Security Rights for User or Role** page to view the effective security group and its definition, based on the selected role or user.

To view Security Rights for User or Role

1. Log on to the **Enterprise Console**.
2. Click **Repository Settings**. The **Repository Settings** page appears.
3. Click **Standard Settings** tab.
   - Click **View Effective Security Rights For User/Role** to view the effective security for a specified user or role at Repository level.
   - Click **View Security Mapping at Repository Level** to view the security mapping for all users or roles at Repository level.

See View Effective Security Rights for Roles and View Effective Security Rights for User/Role for more details.

View Effective Security Rights for Roles

You can search for a role to view the security group and rights assigned to it.

To view security and its rights for the role

1. In the **Standard Settings** tab, click **View Effective Security Rights For User/Role**. The **Role Based** and **User Based** tabs appear.
2. Click **Role Based** tab. The **View Effective Security Rights For Role** section appears.
3. Select a role using the **Role Lookup** dialog box.
4. Click **View Rights** to view the effective security group and rights for the selected role.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Click **Security Snapshots** to view the role name along with the security group to which the selected role belongs.

View Effective Security Rights for User/Role

You can search for a particular user and view the security group and its rights assigned to the user.

To view security and its rights for the role

1. In the **Standard Settings** tab, click **View Effective Security Rights For User/Role**. The **Role Based** and **User Based** tabs appear.
2. Click **User Based** tab. The **View Effective Security Rights For User/Role** section appears.
3. Select a user using the **User Lookup** dialog box.
4. Click **View Rights** to view the effective security group and rights for the selected user.
5. Click **Security Snapshots** to view the user name, roles mapped for the user along with priority, user mapping and effective security group for the selected role.

Levels of Security

AVEVA Work Tasks provides security at different levels across which the user privilege can be managed.

You can set the security at the following three levels:

- Repository Level
- List Level
- ListItem Level

Repository Level Security Settings

The security settings set at this level are applied to the entire repository. This can be overwritten at the List Level and Item Level.

The following are the right privileges given to the user at the Repository level:

- Manage Security Group
- Manage Security Right
- Allow Security Group- Resource Mapping
- Manage Left Navigation
- View Audit Trails
- Delete Audit Trails

See Setting Security at Repository Level for more details.

List Level Security Settings

The security settings at this level can be applied only to that list. In case there are user defined security settings

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

for the Workflow list, they are applicable only to that workflow list. The security settings of other lists will not be affected. A user can navigate to a list only if the user is given top level access rights for that list. For example, Workflow list, User List, Navigation List.

See Setting Security at List Level for more details.

ListItem Level Security Settings

The security settings at this level can be applied only to that item. If the ListItem has any child item and there is no value set for the rights, then the follower security settings will be rendered at the item level.

See Setting Security at ListItem Level for more details.

Set Security at Repository Level

The security rights can be applied at all levels that is, Repository Level, List Level and List Item Level through **Repository Settings**. Only the **Administrator** can view the Repository Settings option. All other security groups cannot view this option as they do not have access rights.

- If the user has **Administrator** rights, then the **Repository Settings** option will be displayed at the right corner along with the **Log Out** option. For other group rights like **Contributor**, **Limited Access** and **Reader**, the Repository Settings option will not be displayed.
- If the **Can Manage Security** is set to **True** for any of the Security Groups then the **Repository Settings** option will be displayed at the right corner. If the **Can Manage Security** right is set to **False** to any of the Security Groups then the Repository Settings option will not be displayed at the right corner.

To apply security setting

1. Log on to the **Enterprise Console**.
2. Click **Repository Settings**. The **Repository Settings** page appears.
3. The **Repository Settings** page displays the following two tabs:
   a. **Standard Settings** tab shows the standard settings options.
   b. **Advanced Settings** tab shows the advanced settings options.

Set Security at List Level

To apply security settings at the List level

1. Log on to the **Enterprise Console**.
2. In the Enterprise Console, click menu icon, and then select any list.
   For example, select the **Workflows** list. The **Workflow** page appears.
3. On the Action bar, click **Security** to set security at the List level. The security settings page appears.
   By default, the **Standard Settings** tab is selected. Click the **Advanced Settings** tab to view the advanced settings options.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

See Enabling Security at List level by enabling resource mapping for more details.

See Enabling Security at List level by customizing the security group definition for more details.

Set Security at ListItem Level

To apply security settings at ListItem level

1. Log on to the **Enterprise Console**.
2. In the Enterprise Console, click menu icon, and then select any list item.
   For example, select the **Workflows** list, and then click any workflow name.
3. On the Action bar, click **Security** to set security at the List level. The security settings page appears.
   By default, the **Standard Settings** tab is selected. Click the **Advanced Settings** tab to view the advanced settings options.
   Alternatively, select and right-click a workflow, and then select **Security Settings** from the context menu to set security at the ListItem level.

See Enabling Security at ListItem level by enabling resource mapping for more details.

See Enabling Security at ListItem level by customizing the security group definition for more details.

Overwrite Definitions from Above Level

The mapping defined at the lower level will take precedence over any mapping defined above it. So, a mapping done at the List level will override any mapping done at the Global (Repository) level. Similarly, the mapping done at the List item level will override any mapping done at the List level if available or else will override at the Global (Repository) level.

Choose between Mapping and Customization

You can choose both or any one of the following options to apply the security settings:

Resource Mapping

Choose this option if you want to privilege or restrict the user(s) at one level, and want to retain the definition at other levels.

For example, the user 'John' is a Contributor access rights user at the Repository level. You want him to provide him Reader rights at the Workflow list level but want retain the security defined at the repository level. To do this, choose **Resource Mapping** while enabling security at the Workflow list level, select this user and map him to the Reader security group.

Once Resource Mapping is enabled at the list level, the user should be able to identify that mapping is enabled at the list level.

Security Group Customization

Choose this option if you want to modify or customize the definition of a security group.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For example, you want all reader (Reader access rights) users who do not have rights at the repository level to create and edit items in any list. You want all reader users to have rights at the Workflow list level to create and edit an item, you should choose **Security Group Customization** and modify the reader security group definition so that all reader users can add and edit an item.

Once Security Group Customization is enabled at the list level, the user should be able to identify that customization is enabled at the list level.

Enable Security at List Level

You can enable list level security in the following ways:

- By enabling Resource Mapping
- By customizing the Security Group definitions

What happens when List level or List Item level mapping is enabled?

The resource mapping available at its parent level is nothing but Repository level copied to the list level. Once the resource mapping is copied to the list level, the user can add, modify, and delete the mapping.

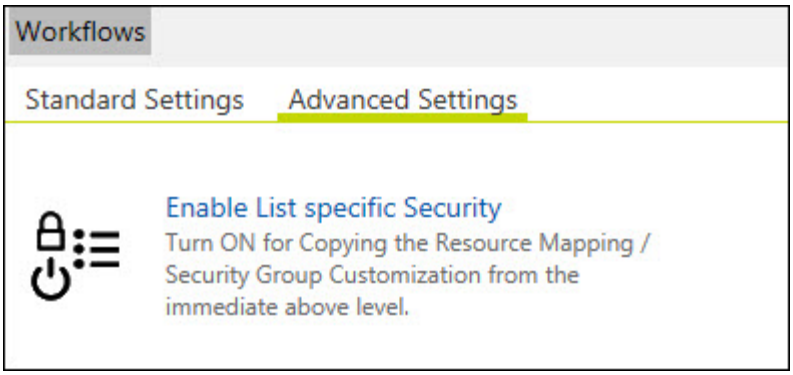
Note: Before mapping is enabled, the changes done at the Repository or Dictator level will not effect the list level mapping. For example, assume that there are 10 user mapping at the repository level. After enabling mapping at the list level, all 10 users will be copied to the list level. If you add a one extra user mapping at the repository level, the newly added mapping at the repository level will not be available at the List level.

Enable Security at List Level by Enabling Resource Mapping

To enable resource mapping at the List level

1. Log on to the **Enterprise Console**.
2. In the Enterprise Console, click menu icon, and then select any list.
   For example, select the **Workflows** list. The **Workflow** page appears.
3. On the action bar, click **Security** to set security at the List level. The security settings page appears.
   By default, the **Standard Settings** tab is selected.
4. Click the **Advanced Settings** tab to enable list level security at the Workflow list level.
   The **Enable List Specific Security** indicates that the Workflows list does not have list specific security.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



5. Click **Enable List Specific Security**. The **Enable Security Wizard** page appears.
6. Check **Resource Mapping** to change the mapping of the role or user at the Workflows list level.
7. Click **Next**. The **Enable Security Wizard** page appears.
8. Click **Next**. The **Security Group Mapping** page appears. You can add a resource mapping, edit a resource mapping, or delete a resource mapping.

Add a Resource Mapping

Add a Resource Mapping

To add a new resource mapping at the List level

1. Log on to the **Enterprise Console**.
2. In the Enterprise Console, click ☰ menu icon, and then select any list.
   For example, select the **Workflows** list. The **Workflow** page appears.
3. On the Action bar, click **Security** to set security at the List level. The security settings page appears.
   By default, the **Standard Settings** tab is selected.
4. Click the **Advanced Settings** tab to enable list level security at the Workflow list level.
   The **Enable List Specific Security** indicates that the Workflows list does not have list specific security.
5. Click **Enable List Specific Security**. The **Enable Security Wizard** page appears.
6. Check **Resource Mapping** to change the mapping of the role or user at the Workflows list level.
7. Click **Next**. The **Enable Security Wizard** page appears.
8. Click **Next**. The **Security Group Mapping** page appears. You can add a resource mapping, edit a resource mapping, or delete a resource mapping.
9. On the action bar, click **New** to add a new resource mapping. The **Mapping : Role to a Security Group** appears.
10. Follow the same steps as explained in Create a Resource Mapping for mapping role and user.

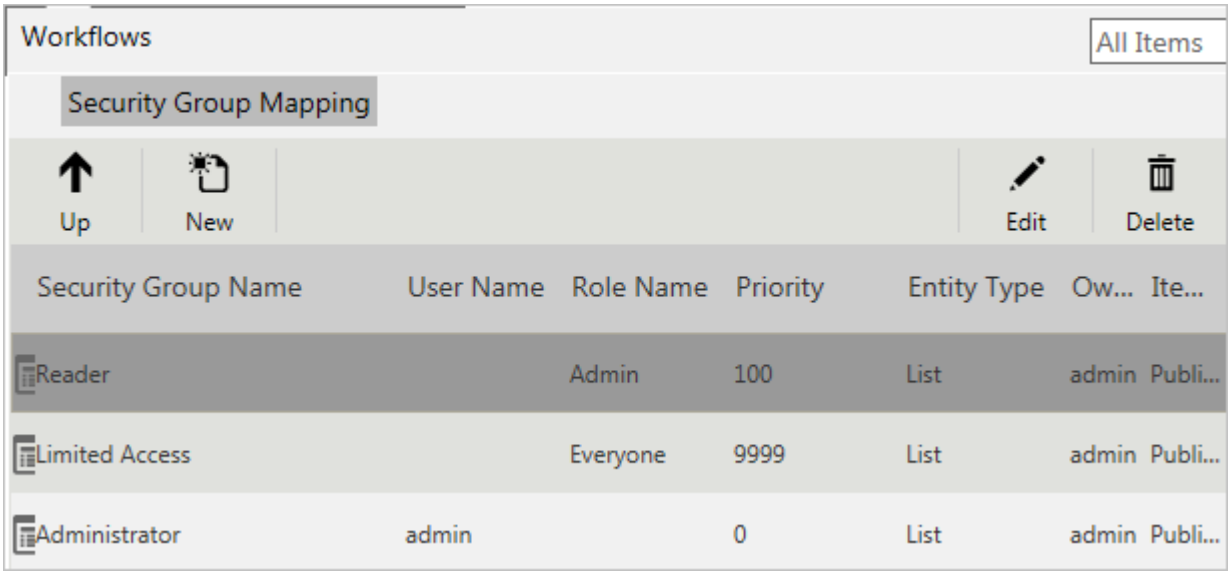
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Edit a Resource Mapping

Edit a Resource Mapping

To edit a existing resource mapping at List level

1. Log on to the **Enterprise Console**.
2. In the Enterprise Console, click menu icon, and then select any list.
   For example, select the **Workflows** list. The **Workflow** page appears.
3. On the Action bar, click **Security** to set security at the List level. The security settings page appears.
   By default, the **Standard Settings** tab is selected.
4. Click the **Advanced Settings** tab to enable list level security at the Workflow list level.
5. Click **Security Group - Resource Mapping for this List**. The **Security Group Mapping** page appears. You can add a resource mapping, edit a resource mapping, or delete a resource mapping.



6. Select the required user or role to be modified, and then click **Edit** to modify a resource mapping.
   The **Mapping : Role to a Security Group** appears.
7. Follow the same steps as explained in Edit Resource Mapping.

Delete a Resource Mapping

Delete a Resource Mapping

See Delete Resource Mapping.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Enable Security at List Level by Customizing the Security Group Definition

List Level Customization

The security group definition available at the repository level can be customized at the list level by copying all the security groups available at the repository level to the list level.

You can change the definition available for a security group and customize the security group definitions at the list level and various item levels.

Note: Do not update the security definition at levels if it is explicitly customized at those levels.

Security Group Customization

Example: The example shows you how to customize the security group definition at list level. The Contributor security group at the repository level is customized at the List level (here, Workflow list item is considered). Considered that the contributor user is able to create an item, folder and delete an item at the repository level. But the contributor user cannot create an item or folder and delete an item only at the workflow list level.

To customize Security Group

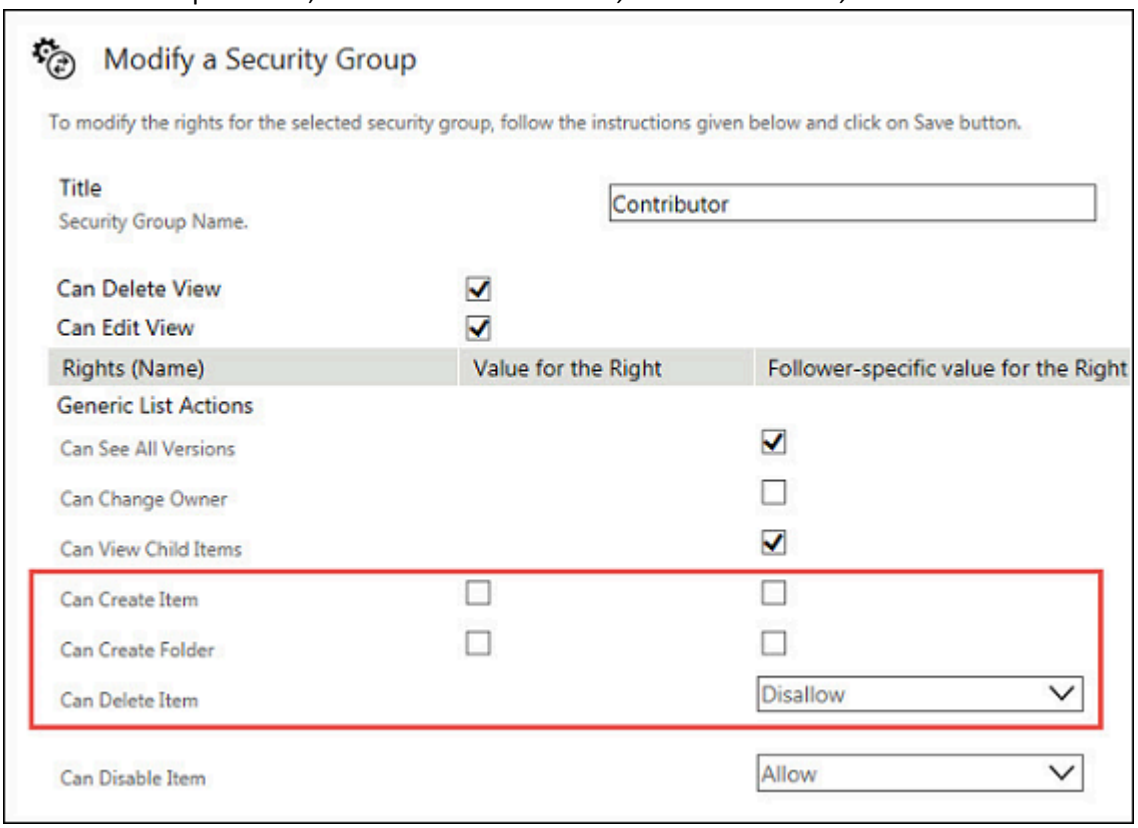
1. Log on as a user who belongs to the **Contributor** security group.
2. In the **Enterprise Console**, click menu icon, and then click **Workflows**. The **Workflows** page appears.
3. Before security settings, the contributor is able to create an item or folder or delete an item by using the options available in **New Workflow** > **New Folder** and **Actions > Delete** in the ribbon bar.
4. To customize this security group definition at the Workflows list level, log on as the administrator user.
5. Click **Repository Settings**. The **Repository Settings** page appears.
6. Click **Standard Settings** tab, and then click **View Security Mapping at Repository Level**. The **View Security Mapping at Repository Level** page appears. This page shows the mapping at the repository level before enabling security at the list level.
7. Click the **Contributor** security group name to view the rights mapped to the security group at the repository level.

   In the Enterprise Console, click menu icon, and then click Workflows. The Workflows page appears.
8. On the Action bar, click **Security**.
9. Click **Advanced Settings** tab.
10. Click **Enable List Specific Security**. The **Enable Security Wizard** page appears.
11. Check **Security Group Customization**.
12. Click **Next**.
13. Click **Next** to continue. The **Workflows** page displays the Security Group information.
14. Select the **Contributor** security group name, and click **Edit** on the action bar. The **Modify a Security Group** page appears.
    Alternatively, select and right-click the **Contributor** security group name, and then click **Edit** from the context

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

menu to customize the security definition.

15. Based on the requirement, uncheck *Can Create Item, Can Create Folder, and Disallow Can Delete Item*.



16. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
17. Choose **Yes** to apply the changes to all its later levels, else click **Cancel**.
18. Click **Submit**. The Security Group is modified successfully message appears.
19. Click **OK**.
20. Click the *Contributor* security group to view the rights mapped to the security group at the list level.
21. Click **View** to check the security rights of the user. The **View Security Group Rights** page appears.
22. Log on as the *Contributor* user.
23. In the **Enterprise Console**, click menu icon, and then click **Workflows**. The **Workflows** page appears.

Now if the Contributor user tries to create an item or folder, or delete an item, the following warning message appears as the Contributor security group is customized only at the Workflow list level.

You do not have rights to perform this action.

The contributor user can still create an item or folder, or delete an item at other list levels such as, Users List, SOA Folder, and File Watcher list.

The Workflows list has its own security. Other list levels such as, Users List, SOA Folder, File Watcher list do not have any security enabled at their list level. Whatever security is defined at the repository level applies to these lists also.

You can enable Resource Mapping and customize the security group at the same time.

The *LastUpdatedDateTime* column in the Security Group list displays the modified date and time of a particular security group, only when it is updated at the Repository level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Remove Mapping Available at List Level

To remove the mapping available at the workflow list level

1. Log on to the **Enterprise Console** as an administrator.
2. Click menu icon, and then click **Workflows**. The **Workflows** page appears.
3. Click **Security**. By default, the **Standard Settings** tab is selected.
4. Click the **Advanced Settings** tab. If the resource mapping is enabled at list level, you can also see the following links in the **Advanced Settings** page.
   a. Security Group - Resource Mapping for this List
   b. Remove Security from this List
   c. Customize Security Group for this List
5. Click **Remove Security from this List** to delete the mapping from the immediate above level. The **Remove Security Wizard** appears.
6. Click **Resource Mapping**, and then click **Next**.
7. Click **Next** for confirmation. The mapped resources are deleted.

Remove Customization Available at List Level

To remove the customization available at the list level

1. Log on to the **Enterprise Console** as an administrator.
2. Click menu icon, and then click **Workflows**. The **Workflows** page appears.
3. Click **Security**. By default, the **Standard Settings** tab is selected.
4. Click the **Advanced Settings** tab. If the resource mapping is enabled at list level, you can also see the following links in the **Advanced Settings** page.
   a. Security Group - Resource Mapping for this List
   b. Remove Security from this List
   c. Customize Security Group for this List
5. Click **Remove Security from this List** to delete the mapping from the immediate above level. The **Remove Security Wizard** appears.
6. Click **Security Group Customization**, and then click **Next**.
7. Click **Next** for confirmation. The mapped resources are deleted.

Enable Security at ListItem Level

By enabling security at listitem level, you can overwrite the security definition at the list or repository level. For example, the user who has contributor rights at the 'Workflow' list level can have administrator rights at the 'Workflow' listitem level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

There are two ways to enable security at the listitem level:

- By enabling Resource Mapping
- By customizing the Security Group definitions

What happens when Listitem level mapping is enabled?

From this point, first it will look at the resource mapping available at the Listitem level. It will not look at the mapping available for the List level or Repository level because Listitem has its own security.

The list item security will copy from the resource mapping available at the List level, if the mapping is available, otherwise it will copy from the repository level.

To apply security settings at the ListItem level

1. Log on as the administrator user.
2. In the Enterprise Console, click menu icon, and then select any list item. For example, select any list item inside the **Workflows** list.
3. Click the listitem, and then click **Security Settings** from the context menu to set security at the ListItem level. The security settings page appears. By default, the **Standard Settings** tab is selected.
4. Click the **Advanced Settings** tab to view the advanced settings options. You can enable list item specific security here.

Enable Security at ListItem Level by Enabling Resource Mapping

Let us understand how to enable security at the ListItem level with a working scenario.

Assume the following exists:

- Two folders, 'HR List' and 'Finance List' in a workflow list
- Two workflows, 'Leave Policy' and 'Employee Details' in the 'HRList' folder
- Three workflows, 'Travel Policy', 'Domestic Policy', and 'International Policy' in the 'Finance List' folder
- Two users, Daniel and Jim. Both users have administrator rights.
  - 'Daniel' belongs to the HR department in the repository
  - 'Jim' belongs to the Finance department in the repository

The intention is to allow the user 'Daniel' to view only HR related workflows while restricting the user 'Jim' from viewing HR related workflows.

To enable security at the ListItem level

1. Double-click **HR List**, and then create the two workflows, 'Leave Policy' and 'Employee Details'.
2. Click **Workflow**. Open 'Finance List', and then create three workflows, 'Travel Policy', 'Domestic Policy', and 'International Policy'.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. In the Enterprise Console, click menu icon, click **Settings**, and then click **Users**.
4. Create users, 'Daniel' and 'Jim'. 'Daniel' belongs to the HR department and 'Jim' belongs to the Finance department.
5. By default, these users have Limited Access rights. Map both 'Daniel' and 'Jim' to the **Administrator** security group.
6. Restrict the user 'Daniel' [HR department] to view only HR related workflows.
   Before enabling security at the List Item level, both users are able to view both folders.
   The HR department user needs to view only the HR related workflows. Also, map the Finance department user to the limited access security group at this item level so that this user cannot view HR related workflows.
7. In the Enterprise Console, click menu icon, and then click **Workflows**. The Workflow page appears.
8. Select and right-click the **HR List** workflow, and then click **Security Settings** from the context menu. The security settings page appears.
   Alternatively, select the 'HR List' workflow, and then click **Security** from the action bar to customize the security definition.
9. Click **Advanced Settings** tab.
10. Click **Enable ListItem Specific Security**. The Enable Security Wizard appears.
11. Click **Resource Mapping**, and then click **Next**.
12. Click **Next**. You will be redirected to the **Security Group Mapping** page.
13. Select the user 'Jim', and then click **Edit** on the action bar. Alternatively, select and right-click the user 'Jim', and then click **Edit** from the context menu.
14. Map 'Jim' to the Limited Access security group.
15. Click **Save** to save the mapping.
16. Log on to the Enterprise Console as 'Daniel' to verify the security configurations. As 'Daniel' belongs to the HR department, they can view only HR related workflows. Similarly, you can map roles to the security groups.

Add a Resource Mapping

Add a Resource Mapping

To add a new resource mapping at the ListItem level

1. Log on as the administrator user.
2. From the **Enterprise Console** menu, select any list.
3. Right-click any item inside the list, and select **Security Settings** to set security at the list item level.
4. Click the **Advanced Settings** tab.
5. Click **Enable ListItem Specific Security** to enable security at the list item level.
6. Check **Resource Mapping** to change the mapping of the role or user, and then click Next.
7. Click **Next**. You will be redirected to the **Security Group Mapping** page.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

8. On the action bar, click **New** to add a new resource mapping. The **Mapping: Role to a Security Group** page appears.
9. Follow the steps explained in Create a Resource Mapping for mapping role and user.

Edit a Resource Mapping

Edit a Resource Mapping

To edit a existing resource mapping at List item level

1. Log on as the administrator user.
2. From the **Enterprise Console** menu, select any list.
3. Right-click any items in the list, and the select **Security Settings** to set security at the list item level.
4. Click the **Advanced Settings** tab.
5. Click **Enable ListItem Specific Security** to enable security at the list item level.
6. Check **Resource Mapping** to change the mapping of the role or user, and then click Next.
7. Click **Next**. You will be redirected to the **Security Group Mapping** page.
8. Select the required user or role to be modified, and click Edit on the action bar. The **Mapping: Role to a Security Group** page appears.
   Alternatively, select and right-click a listitem, and then click Edit from the context menu to modify an existing resource mapping.
9. Select the user 'Daniel', and then click **Edit**.
10. Select a security group from the drop-down list to change the security group for the selected user. For example, 'Daniel' is a Reader. Now map him to the Contributor security group.
11. Click **Save** to apply the security settings only to this workflow at the list item level. This mapping will not apply to the other lists. The confirmation message to save the resource mapping appears.
12. Click **OK**. Now security is enabled at the Workflow list item level.

Delete a Resource Mapping

Delete a Resource Mapping

To delete an existing resource mapping at the List item level

1. Log on as the administrator user.
2. From the **Enterprise Console** menu, select any list.
3. Right-click any items in the list and select **Security Settings** to set security at the list item level.
4. Click the **Advanced Settings** tab.
5. Click **Enable ListItem Specific Security** to enable security at the list item level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Check **Resource Mapping** to change the mapping of the role or the user, and then click **Next**.
7. Click **Next**. You will be redirected to the **Security Group Mapping** page.
8. Select the resource mapping to be deleted, and click **Delete**. For example, select 'Jim' and click the **Delete** option available in the ribbon bar. Alternatively, select and right-click the resource mapping, and then select Delete from the context menu.
9. Click **OK** to confirm deletion.
10. Click **Yes** in the **Delete Resource Mapping** window.
11. Click **OK** in the confirmation window. A message about successfully deleting the user appears.

Enable Security at List Item Level by Customizing the Security Group Definition

ListItem Level Customization

The security group definition available at the Repository level and List level can be customized at the List Item level by copying all the security groups available at the immediate parent level to the list item level.

You can change the definition available for a security group and customize the security group definitions at the list item level.

Note: Do not update the security definition at levels if it is explicitly customized at that levels.

Security Group Customization

Example: This example shows how to customize the Contributor security group definition at the list item level. The Contributor security group at the repository level is customized at the List level and List Item level. (Here the Workflow listitem is considered). The contributor user can design and publish workflows at the repository level but cannot create workflow associations at the repository level.

The intention here is to allow association at the workflow list level for the contributor user and restrict the contributor user to publishing the item at item level.

Considered that a user called 'Kevin', who belongs to the contributor security group, exists in the repository.

To accomplish the scenario described above, perform the following steps:

1. Log on as the administrator.
2. In the **Enterprise Console**, point to admin, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Standard Settings** tab, click **View Security Mapping** at **Repository Level**. The **View Security Mapping** at **Repository Level** page appears. This page shows the mapping at the repository level before enabling security at the list level. The following image shows that the user 'Kevin' belongs to the contributor security group:
4. Click the Contributor security group to view the rights mapped to this security group at the repository level. The following image shows the contributor user behavior at the repository level. At the repository level, the contributor user can design and publish workflows, but cannot associate the workflows at the repository level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Login as 'Kevin' who belongs to the Contributor security group.
6. Go to **Workflows**.
7. Before the security settings are changed, the contributor can design and publish the workflow. If this user tries to associate an event with the workflow, the following warning message appears: You do not have the rights to perform this action.
8. To enable security at the **Workflow** list level, log on as the administrator.
9. Go to Workflows.
10. Click **Security** to set security at the List level. By default, the **Standard Settings** tab is selected.
11. In the Advanced Settings, click **Enable List Specific Security** to enable security at the **Workflow** list level, and click **Next**.
12. Click **Next**. The **Security Group** page appears.
13. Select the Contributor security group, and then click **Edit** on the ribbon bar, or right-click and then click **Edit** from the context menu to customize the security definition of the group. The Modify a Security Group page appears.
14. Check the **Can Create Association** right, and then click **Save**. The Synchronizing the Security Group dialog box appears. Click **Yes** to synchronize the changes to its below level, or click **Cancel**. The following message appears: Security Group has been modified successfully.
15. Click **OK**.
16. After changing the security settings, click the **Navigation** icon to check what is the behavior of this user.
17. Click **View Security Mapping** at List Level. The mapping at list level after security is enabled at the list level.
18. Click the Contributor security group to view the rights mapped to this group at the list level. At the list level, the contributor user can design workflow and publish. After changing the security settings, the Contributor user can also associate events with workflow.
19. To check, log on as 'Kevin' who belongs to the Contributor security group.
20. Go to **Workflows**.
21. After security settings at the list level, the contributor can design a workflow, publish the workflow and also associate the workflow with events.
22. Before enabling security at the list item level, the same security settings will be available for the list item too. If no security is enabled at the list item level, it will copy the security settings available at the immediate parent level. To enable security at the Workflow list item level, login as the administrator user.
23. Go to Workflows.
24. Select any workflow.
25. In the **Advanced Settings** tab, click **Enable ListItem Specific Security**. The following Enabling Security Wizard page appears.
26. Select the Security Group Customization check box, and then click **Next**.
27. Click **Next**. Select the Contributor security group, and click **Edit** on the ribbon bar, or right-click the security group and click **Edit** from the context menu to customize its security definition.
28. Disallow the Can Publish Item right for the Value for the Rights column, and the Follower specific value for the Rights column.
29. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
30. Click **Yes** to synchronize the changes to its below level, or click **Cancel**. The following message appears: *Security Group has been modified successfully*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

31. Click **OK**.
32. After changing the security settings, click the Navigation icon to check the behavior of this user.
33. Click **View Security Mapping at List Item Level**.
34. Click the Contributor security group to view the rights mapped to the security group at list level. At the list level, the contributor user can design workflow and associate events with workflow but will not be able to publish the workflow.
35. To check, log on as 'Kevin' who belongs to the Contributor security group.
36. Go to **Workflows**.
37. Double-click a workflow. For example, Design a workflow called 'TestWF'. Associate this workflow with events, and then publish this workflow. The following warning message appears: *You do not have permission to perform this* action.

After the security settings at the list item level, the contributor will not be able to publish the workflow. The contributor user can still design and publish a workflow and associate an event with other workflow list items.

Here, the Workflows list has its own security and other list levels like Users List, SOA Folder, File Watcher list etc do not have security enabled at their list level. Whatever the security defined at the repository level will apply to these lists.

Note: You can enable Resource Mapping and customize the security group at the same time. The LastUpdatedDateTime column in the Security Group list displays the modified date and time of a particular security group, only when it is updated at the Repository level.

Remove Mapping Available at ListItem Level

To remove the mapping available at list item level

1. Log on as an administrator user.
2. Select **Workflows** from the **Enterprise Console** menu.
3. Right-click the workflow where security is enabled at the item level and select **Security Settings** from the context menu. By default, the **Standard Settings** tab is selected.
4. Click the **Advanced Settings** tab.
5. Click **Remove Security from this ListItem** to delete the mapping from the immediate above level. The following **Remove Security Wizard** page appears.
6. Check **Resource Mapping**, and then click **Next**.
7. Click **Next** to delete the resources mapped at this item level.
8. To check, click the **Advanced Settings** tab. The Advanced Settings tab contains only the **Enable ListItem specific Security**.

Remove Customization Available at ListItem Level

To remove the customization available at the list item level

1. Log on as an administrator user.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Select **Workflows** from **Enterprise Console** pull-down menu.
3. Select the Workflow for which you want to remove customization.
4. Click **Security**. By default, **Standard Settings** tab is selected.
5. Click **Advanced Settings** tab.
6. Select the **Security Setting** you want to remove.
7. Click **Remove Security** from this ListItem to delete the mapping from the immediate above level. The following **Remove Security Wizard** page appears.
8. Check **Security Group Customization**, and then click **Next**.
9. Click **Next** to delete the customization at this item level.
10. To check, click **Advanced Settings** tab. The **Advanced Settings** tab contains only the **Enable ListItem specific Security**.

Effect on Resource Rights after Deleting the Mapping

The example in this section explains the effect on resource rights after deleting the resource mapping at the item level.

Consider a Workflow listitem. A user belongs to the Contributor security group at the repository level, and to the Limited Access security group at the ListItem level. This example shows the resource rights, if the user deletes the mapping done at the listitem level.

Assume that there are four users in a repository with the following resource mapping at the repository level:

Users	**Security Group Mapping**
John	Administrator
Kevin	Contributor
Bob	Reader
Mac	Limited Access

Here John, Kevin, and Bob are mapped to the Administrator, Contributor, and Reader security groups respectively.

1. Log on as the Administrator user to map the Contributor user 'Kevin' to the Limited Access security group at the workflow listitem level.
2. Go to **Workflows**.
3. Right-click a workflow, for example, 'Sale Workflow', and the select **Security Settings** from the context menu.
4. Click **View Effective Security for User/Role**.
5. Select the **User Based** tab, and search for the user 'Kevin' to view mapping at the listitem level.
6. Before security is enabled at the listitem level, 'Kevin' has Contributor rights at the 'Sale Workflow' item level.
7. Click **Security Snapshots** to view the mapping details at the repository level.
8. Go to **Advanced Settings** tab, and click **Enable ListItem Specific Security** to enable the security at the List level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Check **Resource Mapping** to change the mapping of the user 'Kevin', and then click **Next**.
10. Click **Next**.
11. Select 'Kevin', and click **Edit** from the ribbon bar, or right-click the user and select Edit from the context menu.
12. The Mapping : User to a Security Group appears.
13. Map the user 'Kevin' to the Limited Access security group at the list level.
14. Click **Save**. The resource mapping saved successfully message appears.
15. Click **OK**. You will be redirected to the Security Group Mapping page.
16. Now the user 'Kevin' has Limited Access rights at the Workflow listitem level.
17. After changing the security settings, check the behavior of this user.
18. Click **View Effective Security for User/Role**.
19. Select the **User Based** tab and search for 'Kevin' to view the mapping at the listitem level.
20. After enabling security at the listitem level, the user 'Kevin' has Limited Access rights at the 'Sale Workflow' item level.
21. Click **Security Snapshots** to view mapping details.
22. Now, remove the mapping at this list item level. To do this, click **Advanced Settings** tab, and then click **Remove Security from this ListItem**.
23. Select **Resource Mapping**, and click **Next** to remove mapping at this level.
24. Click **Next**.
25. Click **View Effective Security for User/Role** to view after deleting mapping at the 'Sale Workflow' item level.
26. Select the **User Based** tab and search for 'Kevin' to view the mapping at the listitem level.
27. After removing security at the listitem level, the user 'Kevin' has Contributor rights at the 'Sale Workflow' item .
28. Click **Security Snapshots** to view mapping details.

Synchronization

Synchronization is the option that lets the user synchronize the security group at all the levels below the current level to propagate the changes wherever the Sync (synchronization flag) is ON, in a hierarchical order.

As soon as the Sync flag is found OFF at a particular level, it will not move further to synchronize the Security Group definition.

You can implement security at the following levels in a repository:

1. List
2. ListItem -This level can have n number of levels, parent item > child item > child item

Scenario 1: The user makes changes (edits) in a security group at the Repository level

The user should be allowed to synchronize the security group at all available lists and the listitems wherever security is enabled and the sync flag is ON for that security group.

If the security is not enabled at the list level then it should search for its items to find whether security is enabled at any item. If it is there and the sync flag is ON then it should propagate the changes at that level and its child items in a hierarchical order. Once the sync flag is found OFF, it will not move further to synchronize.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scenario 2: The user makes changes in a security group at the List (Workflow) level

The user has the option to propagate the changes to all the listitems wherever security is enabled and the sync flag is ON in a hierarchical order. Once the sync flag is found to be OFF, it will stop moving further to synchronize the below items.

Scenario 3: The user makes changes in a security group at the List Item level

The user has the option to propagate the changes to all the child items wherever security is enabled and the sync flag is ON in a hierarchical order. Once the sync flag is found to be OFF, it will stop moving further to synchronize the child items.

If a new security group is created at the Repository level, it will synchronize its definition at all levels irrespective of the Sync (synchronize flag) option.

Secure Repository Using Security Framework

Using the AVEVA Work Tasks security framework, you can secure the repository by controlling the List and ListItems which are a part of that repository and features like Process Designer, Business Activity Monitoring (BAM), Work item, Calendar, and Queue.

See Securing List and ListItems and Securing Workflow Management Server Controls for more information.

Secure List and ListItems

We assume that you have basic knowledge about AVEVA Work Tasks list.

List

List security is required to control the access to list.

ListItem

The ListItem security is required to control the access to items inside the list. The security settings at this level can be applied only to that item. The ListItem features like create, delete, edit, publish, unpublish, disable, and enable can be secured.

Secure AVEVA Work Tasks Server Controls

AVEVA Work Tasks list has the following controls:

Process Designer

This control is used to design workflows. You can secure the Process Designer by controlling the access to categories, actions, properties and menus.

See Process Designer Rights for more information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Business Activity Monitoring (BAM)

This control is used to monitor reports. You can secure the BAM control by controlling the access to reports based on workflows scope like who initiated a workflow, who owns a workflow, and the workflows owned by others.

You can control the menus like Dashboard, Analyze, and Execution etc. based on user privileges.

You can also control the views like Analysts, Progress and Execution based on user privileges.

See BAM Security for more information.

Work Item

This control is used to view and act on all the work items assigned to the user. You can secure work items by controlling access to the work item features like forwarding work item, setting alternate actor, setting preference, viewing queue, viewing resource calendar, and viewing work item control.

See Securing Work Item Control for more information.

Calendar

This control is used to set the calendar details of the organization like working hours, holidays and etc. You can secure the Calendar by controlling the access to adding new calendars, deleting calendars, viewing calendars, viewing base calendars, and viewing calendar control.

See Securing Calendar Control for more information.

Queue

You can secure queue by controlling the access to adding new queues and viewing the queue control.

See Securing Queue Control for more information.

Secure Menus

This section provides information on how to manage a navigation list. Use the **Manage Navigation List** option to create a new navigation, Form, List, Dashboard, and other folder items in the Enterprise Console, and also to modify existing items.

Manage Navigation List

Use the **Manage Navigation List** option to create a new navigation (Left menu), Form, List, Dashboard, and folder items in the Enterprise Console, and also to modify existing items. Using this option, you can set security rights at the List and ListItem level.

Use the **Manage Navigation List** option to create a new Navigation link, Forms link, List link, Dashboard link, and folder. You can edit existing links and define security at the List and ListItem level.

To open the Manage Navigation List

1. Log on to the Enterprise Console.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Point to **admin**, and then click **Repository Settings**.

   The **Repository Settings** page appears.
3. Click the **Advanced Settings** tab.
4. Click **Manage Navigation List**.

   The **Navigation List** page appears.

Create Folder

Use the **New Folder** option to create a folder. You can create Navigation link, Form link, List link, and sub folders under the folder, and also can edit the Navigation list details. You can set the security definition at List and ListItem level.

To create a folder

1. On the action bar of the **Navigation List** page, click **New**, and then click **Folder**.

   The **New Folder** page appears.
2. In the **Title** box, type a name for the folder.

   A folder name should not have **:"\\""'<>;|*/{}?** special characters.
3. In the **Description** box, type a description for the folder.
4. In the **Image Name** box, type a name for the image.

   The image should be available in the path *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**BPMUITemplates**Default**Repository**Site**images**MenuIcons*.
5. In the **Menu Order** box, type a number to position the folder in the **Enterprise Console** menu.
   - By default, the **Menu Order** number will be one more than the highest **Menu Order** number entered for the links in the **Navigation List**.
   - You can enter either a positive or a negative integer. Negative **Menu Order** numbers take precedence over positive **Menu Order** numbers.
   - If you clear the **Menu Order** number, then the system will set the **Menu Order** to **0**.
   - The link with the lowest **Menu Order** number appears first in the menu followed by other links in the ascending order of their **Menu Order** number.
   - In case of links with the same **Menu Order** number, the first saved link takes precedence and appears before the other links with the same **Menu Order** number.
   - You must log out and log on to the **Enterprise Console** to see changes in the order of the menu items.
   - If you manually add any menu item in the system defined menus, then after upgrade, the menu order will not be retained as defined. You must again reset the menu order based on your preference.
6. Click **Save**.

   The folder is created.

   Note: When you create a new folder or item in AVEVA Work Tasks, the Item Status will be Published. However, this is not applicable for Forms and Workflows folders and items.

Create Navigation Link

Use the **Navigation** option to create a new menu item in the Enterprise Console menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To create a new navigation list

1. On the **Action** bar of the **Navigation List** page, click **New**, and then click **Navigation**.

   The New Navigation Link page appears.
2. In the **Title** box, type a unique name for the navigation link, and in the **Description** box, type a description for the folder.
3. In the **Navigate To Page** box, enter the path of the aspx file name.

   The aspx file must be available in the following path:

   ```
   ..\AVEVA\Work Tasks\Web\BPMUITemplates\Default\Repository\Site
   ```

   If you save this aspx file in some other location, mention the path with the aspx file name. For example, if the aspx file is in the Repository folder, then mention the path in the **Navigate To Page** box as follows:

   ```
   \Repository\<file name.aspx>
   ```
4. In the **Image Name** box, enter the image name.

   The image must be available in the following path:

   ```
   [AVEVA Work Tasks Installation Path]\Web\BPMUITemplates\Default\Repository\Site\
   images\MenuIcons
   ```

   If you save this image file in some other location, mention the path with the image file name. For example, if the image is in the Repository folder, then mention the path in the **Image Name** box as follows:

   ```
   [AVEVA Work Tasks Installation Path]\Web\BPMUITemplates\Default\Repository\<image
   file name>
   ```

   If you have applied default theme to the repository, then the image must be available in the following path:

   ```
   [AVEVA Work Tasks Installation Path]\Web\BPMUITemplates\Default\Themes\Default\
   Repository\Site\Images\MenuIcons
   ```

   If the new theme is applied to the repository, then the image must be available in the following path:

   ```
   [AVEVA Work Tasks Installation Path]\Web\BPMUITemplates\Default\Themes\<New
   Theme>\Repository\Site\Images\MenuIcons
   ```
5. In the **Menu Order** box, type a number to position the link in the **Enterprise Console** menu.
   - By default, the **Menu Order** number will be one more than the highest **Menu Order** number entered for the links in the **Navigation List**.
   - You can enter either a positive or a negative integer. Negative **Menu Order** numbers take precedence over positive **Menu Order** numbers.
   - If you clear the **Menu Order** number, then the system will set the **Menu Order** to **0**.
   - If you create a new List using **Tools** > **Manage List**, you must set the menu order for that list from the **Repository Settings** > **Advanced Settings** > **Manage Navigation List** page. If you do not set the menu order for the list manually, the menu order of the list will be considered as 0, and the list will appear at the top of the Enterprise Console menu.
   - The link with the lowest **Menu Order** number appears first in the menu followed by other links in the ascending order of their **Menu Order** number.
   - In case of links with the same **Menu Order** number, the first saved link takes precedence and appears before the other links with the same **Menu Order** number.
   - You must log out and log on to the **Enterprise Console** to see changes in the order of the menu items.
   - If you manually add any menu item in the system defined menus, then after upgrade, the menu order will not be retained as defined. You must again reset the menu order based on your preference.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Click **Submit**.

   The navigation link is created.

7. Click **OK** to create the new left menu item.

Examples

1. Create two aspx pages. Create one aspx page with the main page and another aspx page without the main page.
   - BPM.aspx(With Main Page)

```
<%@ Page Language="C#" MasterPageFile="~/BPMUITemplates/Default/Repository/
Site/MasterPage.main" AutoEventWireup="true"
Inherits="Skelta.Repository.Web.CodeBehind.WorkFlowList" Title="Untitled Page"
%>
<asp:Content ID="Content1" ContentPlaceHolderID="Cont" Runat="Server">
<div>
<font color="purple"> <center><b>Welcome to AVEVA Work Tasks
</b></center></font>
</div>
</asp:Content>
```

   - BPMWithoutMaster.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="BPMWithoutMaster.aspx.cs" Inherits="BPMWithoutMaster" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
</head>
<body>
<form id="form1" runat="server">
<div>
<font color="purple"> <center> <b>Welcome to AVEVA Work Tasks </b>
</center></font>
</div>
</form>
</body>
</html>
```

2. Save these two aspx pages in the following path:

```
..\AVEVA\AVEVA Work Tasks\Web\BPMUITemplates\Default\Repository\Site
```

3. On the action bar of the **Navigation List** page, click **New**, and then click **Navigation**.

   The New Navigation Link page appears.

4. Enter the **Title** as **AVEVA Work Tasks Menu**.
5. Enter a description.
6. In the **ImageName** box, type image name as **External-Service.jpg**.

   This image must be available in the following path:

```
..\AVEVA\AVEVA Work Tasks\Web\BPMUITemplates\Default\Repository\Site\images\MenuIcons
```

7. Click **Submit**, and then click **OK** on the confirmation window.
8. Double-click **AVEVA Work Tasks Menu**, and click **Navigation** to create a sub-menu item called **BPM Without Main Page**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

a. Enter the sub-menu name **BPM Without Main Page** in the **Title** box.
b. Enter a description.
c. Enter the navigation link name **BPMWithoutMaster.aspx** in the **NavigateToPage** box.

   This file is available in the following path:

   ```
   ..\AVEVA\AVEVA Work Tasks\Web\BPMUITemplates\Default\Repository\Site
   ```

d. Enter the image name, **icon-manage-forms.gif** in the **ImageName** box.

   This file is available in the following path:

   ```
   ..\AVEVA\Wonderware\AVEVA Work Tasks\Web\BPMUITemplates\Default\Repository\
   Site\images\MenuIcons
   ```

e. Click **Submit**, and then click **OK** on the confirmation window.
f. Now the sub-menu item, **BPM Without Main Page** is created under **AVEVA Work Tasks Menu**.

9. Similarly, create another sub-menu item called **BPM With Main Page**.
10. Log out from the Enterprise Console, and then log on again.

    The newly created menus get reflected in the **Enterprise Console** menu.
11. Click **AVEVA Work Tasks Menu**, and then click **BPM Home With Main Page**.

    This redirects you to the **BPM Home With Main Page**.
12. In the **Enterprise Console** menu, select **BPM Without Main Page**.

    Here you will not be able to view the **Enterprise Console** menu, **Repository, Provider**, and **Logged In** information.
13. To go back to the **Enterprise Console** menu, close this window, and then log on to the **Enterprise Console** again.

    Use the following code in all the user created aspx pages to **avoid logging in** to the **Enterprise Console** again:

    ```
    <%@ Page Language="C#" MasterPageFile="~/BPMUITemplates/Default/Repository/Site/
    MasterPage.main" AutoEventWireup="true"
    Inherits="Skelta.Repository.Web.CodeBehind.WorkFlowList" Title="Untitled Page" %>
    ```

Note: Ensure the size of the image (icon) is 25x25 pixels.

Create Form Link

You can create form links in the **Enterprise Console** menu for the forms created in the **Forms** section. This option is especially useful when the number of forms in **Forms** is very high. Use the **Form Link** to easily navigate to the required form using links in the **Enterprise Console** menu.

To create a form link

1. On the **Action** bar of the **Navigation List** page, click **New**, and then click **Form Link**.

   The **New Form Link** page appears.

   You can also create a folder here using the folder option and create the form links under it for better readability.
2. In the **Title** box, type a unique name for the form link, and in the **Description** box, type a description for the form link.
3. In the **Select a Form** box, select the form to be linked from the available list.
4. In the **Image Name** box, type the image name.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If the image is not mentioned, a default image will be inserted. The default image is located in the following path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\BPMUITemplates\Default\Repository\Site\images\MenuIcons

You must save your image files in this path.

5. If the **Window Mode** for the form link is set to **Pop-up**, the form appears as a pop-up in the **Enterprise Console**.
   If the **Window Mode** for the form link is set to **Inline**, the forms appears inline in the **Enterprise Console**.
6. Under **Show in Navigation List**, select as follows:
   - Select **Web** to show form link in the navigation menu of the **Enterprise Console**.
   - Select **Mobile** to show form link in the navigation menu of the **AVEVA Work Tasks Pro** app.

   Note:
   - The **Show in Navigation List** property is applicable only when the **Window Mode** property is set to **Inline**.
   - By default, the **Show in Navigation List** is set to **Web**.
   - The **Form Link** navigation items created under **Folder** navigation item will not be available in AVEVA Work Tasks Pro app.
7. In the **Menu Order** box, type a number to position the link in the **Enterprise Console** menu.
   - By default, the **Menu Order** number will be one more than the highest **Menu Order** number entered for the links in the **Navigation List**.
   - You can enter either a positive or a negative integer. Negative **Menu Order** numbers take precedence over positive **Menu Order** numbers.
   - If you clear the **Menu Order** number, then the system will set the **Menu Order** to **0**.
   - The link with the lowest **Menu Order** number appears first in the menu followed by other links in the ascending order of their **Menu Order** number.
   - In case of links with the same **Menu Order** number, the first saved link takes precedence and appears before the other links with the same **Menu Order** number.
   - You must log out and log on to the **Enterprise Console** to see changes in the order of the menu items.
   - If you manually add any menu item in the system defined menus, then after upgrade, the menu order will not be retained as defined. You must again reset the menu order based on your preference.
8. Click **Save**.

   The form link is created.
9. Log out of the **Enterprise Console**.
10. Log on to the **Enterprise Console** again, and view the form link created in the **Enterprise Console** menu.

    Before accessing the form link, you must log out and then log on to the Enterprise Console again to refresh the Enterprise Console menu.

Note: While designing non workflow initiator forms, you can hide the **Submit** button and the title bar using the **Show Submit button** and **Show top level title** properties of the form.

For more information, see *Forms and List User Guide*.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create List Link

Use this option to create a new list link.

To create a new list link

1. On the **Action** bar of the **Navigation List** page, click **New**, and then click **List Link**.

   The **New List Link** page appears.
2. In the **Title** box, type a unique name for the list link, and in the **Description** box, type a description for the list link.
3. In the **Select a List Name** drop-down, select the required list.
4. In the **Image Name** box, type the image name.

   If the image is not mentioned, a default image will be inserted. The default image is located in the following path:

   *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**BPMUITemplates**Default**Repository**Site**images**MenuIcons*

   You must save your image files in this path.
5. In the **Menu Order** box, type a number to position the link in the **Enterprise Console** menu.
   - By default, the **Menu Order** number will be one more than the highest **Menu Order** number entered for the links in the **Navigation List**.
   - You can enter either a positive or a negative integer. Negative **Menu Order** numbers take precedence over positive **Menu Order** numbers.
   - If you clear the **Menu Order** number, then the system will set the **Menu Order** to **0**.
   - If you create a new List using **Tools** > **Manage List**, you must set the menu order for that list from the **Repository Settings** > **Advanced Settings** > **Manage Navigation List** page. If you do not set the menu order for the list manually, the menu order of the list will be considered as 0, and the list will appear at the top of the Enterprise Console menu.
   - The link with the lowest **Menu Order** number appears first in the menu followed by other links in the ascending order of their **Menu Order** number.
   - In case of links with the same **Menu Order** number, the last saved link takes precedence and appears before the other links with the same **Menu Order** number.
   - You must log out and log on to the **Enterprise Console** to see changes in the order of the menu items.
   - If you manually add any menu item in the system defined menus, then after upgrade, the menu order will not be retained as defined. You must again reset the menu order based on your preference.
6. Click **Save**.

   The link list is created.
7. Click **OK**.

   You will be redirected to the linked list page.

   The newly created form link appears in the **Enterprise Console** menu. If you click the link list, it redirects you to the link list page.
8. Click **New** to insert list items.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create Dashboard Link

Create a dashboard link to add a tabbed page as a link in the **Enterprise Console** menu. This page can have one or more forms (except list forms) with each form appearing in its own tab. You can also set this tabbed page as the home page.

To create a dashboard link

1. On the **Action** bar of the **Navigation List** page, click **New**, and then click **Dashboard**.

   The **New Dashboard Link** dialog box appears.
2. In the **Title** box, type a name for the dashboard link, and in the **Description** box, type a description for the dashboard link.
3. In the **Menu Order** box, type a number to position the link in the **Enterprise Console** menu.

   Note:
   - By default, the menu order number will be one more than the highest menu order number entered for the links in the **Navigation List**.
   - The menu order number should be **0** or a positive integer.
   - The link with the lowest menu order number appears first in the menu followed by other links in the ascending order of their menu order number.
   - In case of links with the same menu order number, the last saved link takes precedence and appears before the other links with the same menu order number.
   - You must log out and log on to the **Enterprise Console** to see changes in the order of the menu items.
4. In the **Tab Order** box, type a number to position the tab in the tabbed page.
5. In the **Type** list, select **Forms**.
6. In the **Title** box, type a name for the tab.
7. In the **Navigation** list, select a form.
8. To select form parameters, if required, click the **Form Parameters** icon.

   The **Form Parameters** dialog box appears.
9. In the **Value** box against each parameter, type the values for the parameters, and then click **Finish**.
10. To add another tab, click **Add Record**.

    A new record appears.

    Note:
    - A minimum of 1 tab is required for the tabbed page.
    - A maximum of 10 tabs are supported by the tabbed page.
    - To delete a tab, click the **Delete** icon.
11. Repeat **Steps 4 to 9** for each added tab
12. Click **Save**.
13. The dashboard link is saved and an information message appears.
14. Click the **Close** button.

    The information message and the **New Dashboard Link** dialog box are closed.

For more information on **Forms** and **Form Parameters**, see *Forms and Lists User Guide*.

Note: Ensure forms added to tabs are available. If a form is deleted, then the tab containing the deleted form must be updated with an available form.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To set a dashboard link as the home page

1. In the **Enterprise Console**, Click the **User** icon, and then click **Repository Settings**.

   The **Repository Settings** page appears.
2. Click the **Advanced Settings** tab, and then click **Manage Security Groups**.

   The **Security Group** page appears.
3. Right-click on a security group, and then click **Edit** (or click on a security group, and then on the **Action** bar, click **Edit**).

   The **Modify a Security Group** dialog box appears.
4. In the **Default Home Page** list, click a dashboard link.
5. Click **Save**.

   The security group settings are saved, the **Modify a Security Group** dialog box is closed, and the **Synchronization** dialog box appears.
6. Under **Do you want to Synchronize it?**, click **Yes**, and then click **Submit**.

   The synchronization settings are saved and an information message appears.
7. Click the **Close** button.

   The information message and the **Synchronization** dialog box are closed.

Edit Navigation List Items

To edit a navigation list item

1. In the **Navigation List** page, select a navigation list item.
2. Click **Edit** on the ribbon.

   The **Edit <navigation list item>** appears.

   Alternatively, select and right-click a navigation list item, and then click **Edit**.
3. Modify the desired field.
4. Save the modifications.

Change Owner Information

You can change the owner of a list item. You might need this feature to inherit another user's items to make it own item, or change owner of that item. Only an administrator can change the owner of a list item.

To change owner of the navigation list

1. In the **Navigation List** page, select a navigation list item.
2. Click **Edit** on the ribbon.

   The **Change List Item Owner** page appears.

   Alternatively, select and right-click a navigation list item, and then click **Edit**.
3. In the **Select a User** box, enter the user name using the **User Lookup** window.
4. Click **Save**.

   The owner information is changed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

5. Click **OK** in the confirmation window.

Organize Navigation List

As new folders, navigation links, form links, list links get added under the Enterprise Console menu, there is a need to organize them. The **Organize** option in the Navigation List ribbon bar enables you to organize and categorize the Navigation List items. For example, you can create folders and group the list items into separate folders using the Organize option.

Consider a scenario, where we have four form links, Leave Approval, Leave Summary, Mark Attendance, and On Duty Request under the Enterprise Console menu as shown in the image given below. As all these menu items are related to Leave Management, we would like it to group it under Leave Management section. The steps given below explain how to organize the menu items using Organize feature of the Navigation List.

To group navigation list items within folders:

1. In the **Navigation List** page, click **Organize** on the ribbon.

   The **Navigation Details** page appears.

   Alternatively, select and right-click a navigation list item, and then click **Edit**.

   The Organize Navigation Details window arranges all the folders and links in a tree format. You can modify the list items organization in two ways, either Drag and Drop or use the context menu.
2. You can drag and drop the navigation list items to organize.
3. Click **Close**.

   You can see that the Enterprise Console menu changes when you log on the next time.

Secure Event Providers

The Event Provider is the component of Event Framework which handles a specific events.

Manage Event Providers by editing, changing owner information, and securing the Event Provider.

Manage Event Providers

Use the **Manage Event Providers** option to manage the event providers such as, creating a new event provider, editing an existing event provider, changing owner information, or securing event provider.

To open Manage Event Providers

1. Log on to the **Enterprise Console**.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. Click the **Advanced Settings** tab.
4. Click **Manage Event Providers**. The **Event Provider** page appears.

Create an Event Provider

Use the New option to create an event provider.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To create a new event provider

1. On the Action bar of the **Event Provider** page, click **New**. The **New Event Provider** page appears.
2. In the **Title** box, type a name for the event provider, and in the **Description** box, type a description for the event provider.
3. In the **Provider Type** box, enter the type of provider. This field is mandatory.
4. In the **Display Priority** box, enter the priority. This field is mandatory. As the name indicates, the event providers are displayed in the **Association** page based on the priority specified for the event providers. We recommend you to specify the display priority from 11 onwards because AVEVA Work Tasks might introduce new event providers which might be assigned priority from six to 10.
5. Click **Save & Continue**. The event provider is created.
6. Click **OK**.

The following table shows the order in which the AVEVA Work Tasks event providers are displayed in the **Association** page. You can change the priority order.

Event Provider	Display Priority
Enterprise Console	2
BAM	4
File Watcher	5

Note:
When you create a new event provider, an entry is created in the **SKEEventProviders** table. However, the entry does not appear in the Association page. You must manually add the event name to the SKAddInProviders table.

If the event provider is already assigned a priority, no error message appears at the time of creation. However, there regarding which event provider to be displayed first. You might get a page crash while opening the association page.

To display the newly created event provider, you have to write an Event Provider class and the functionality for it.

Edit an Event Provider

The **Edit** option provides detailed steps for editing an existing event provider.

To edit an Event Provider

1. Select the event provider to be modified.
2. On the Action bar, click **Edit**. The **Edit Event Provider Details** page appears.
   Alternatively, right-click and select **Edit** from the context menu. You can modify the title, description, provider type, and display priority of the selected event provider.
3. Click **Save & Close**. The modifications are saved, and a confirmation message appears.
4. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Change Owner Information

The **Change Owner** option helps the user to change the owner of an event provider based on user requirements. This may be needed when the user wants to inherit someone else's items or change the owner of items.

The Change Owner capability is restricted to the Administrator user.

To change the owner of the navigation list

1. Select the event provider in which the owner information has to be changed.
2. On the Action bar, click **Change Owner**. The **Changing Owner** page appears.
3. Alternatively, select and right-click an event provider, and then select **Change Owner** from the context menu.
4. Select the user from the **User Lookup** window.
5. Right-click and select **Change Owner** from the context menu or click **Change Owner** on the ribbon bar.
6. Click **Save**. The owner information is saved.

Secure Event Provider

Use the **Security** option to prevent the event providers from being displayed in the **Association** window for users of certain security groups.

Example:

The scenario describes how a user who has Contributor rights should not be allowed to view the Enterprise Console option in the Association screen.

Procedure

1. Select the event provider in which the owner information has to be changed.
2. Select and right-click an event provider, and then select **Security Settings** from the context menu. The security settings page appears.
3. In the **Advanced Settings**, click **Customize Security Group for this List Item**.
4. Select a **Security Group Name**. For example, **Contributor**.
5. On the Action bar, click **Edit**. The **Modify a Security Group** page appears. Alternatively, right-click **Contributor,** and then select **Edit** from the context menu.
6. The **Can Consume Item** is checked at both the levels. Uncheck this so that the Contributor security group cannot view the **Enterprise Console** list.
7. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
8. Click **Yes** to synchronize all the listitems. A message about successfully modifying the security group appears.
9. Click **OK**.
10. Log on as a user who has **Contributor** rights. For example, AVEVA Work Tasks provides a default user, George with contributor rights.
11. Go to **Workflows**.
12. Select any workflow, and then click **Associate** on the action bar. The **Association** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

13. The contributor user cannot view the Enterprise Console in the **Association** page.
    Ensure that the **Can Create Association** right is set to **True** to perform this action.
14. Log on as the administrator user.
15. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
16. In the **Advanced Settings** tab, click **Manage Navigation List**.
17. Select a **Navigation Link**, and then click **Security**.
18. In the **Advanced Settings** tab, click **Remove Security from this ListItem** to remove security from this list item.

    Ensure that the **Can View Item** right is checked for all the levels. If this right is unchecked, then this option will be removed from the navigation list.

Secure Queue Control

You can use the **Queue Management** window to define the security of a queue. Only administrators can define the security for a queue.

To open the Queue Management window

In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Queue Management**. The **Queue Management** page appears.

The Queue Management window lists all the queues that have already been added in a grid. The **Queue Management** window can be controlled using **Queue Control Right**.

The following table summarizes the Queue Control rights available for a repository:

Queue Control Rights	
Can Add New Queue	This right controls the access to add a new queue.
Can View Queue Control	This right controls the access to view queue control.
Manage Queues	This right controls the access to modify/delete queue.

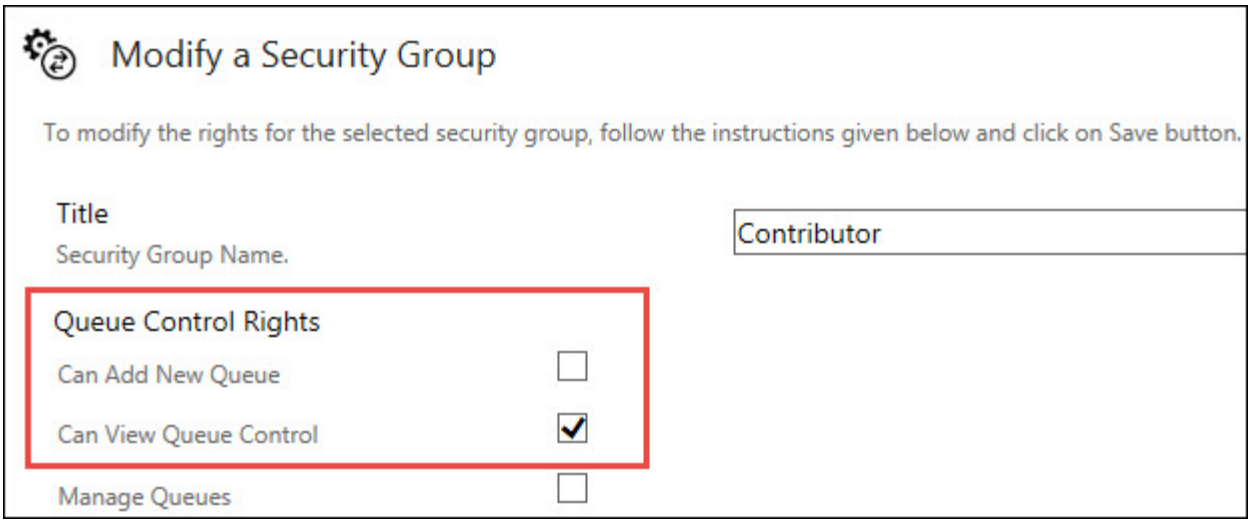
Control New Queue Addition

A contributor user cannot add a new queue. Only the Administrator user can add a new queue. The **New Queue** option is not available for the Contributor user.

The following example demonstrates how the **Queue Control** rights can be used to allow a contributor user to add a new queue.

1. To control the access to add a new queue for a **Contributor** user, log on as an **Administrator** user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Select a **Security Group Name**, and then click **Edit**. The **Modify a Security Group** page appears. For example, select the Contributor security group.
   The following image shows that the **Contributor** user cannot add a new queue even though this user can view the **Queue Management** window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



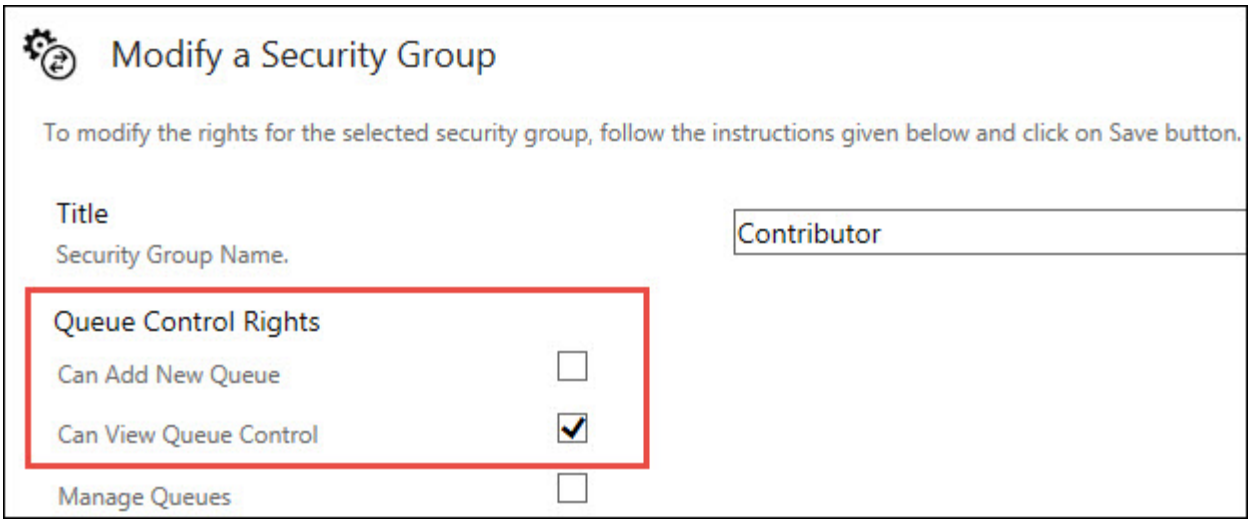
5. Check the **Can Add New Queue** to provide permissions to the **Contributor group** to **New Queue**.
6. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
7. Click **Yes**, if you want to synchronize the changes to its below levels, or else click **Submit**.
8. Click **OK** on the confirmation window.
9. Log on as a user who belongs to the Contributor security group.
10. Go to the **Queue Management** page. Now, the contributor user will be able to view the button to add a queue in the **Queue Management** window.

Control Queue View

The following example demonstrates how to prevent a contributor user from viewing the Queue Management window.

1. Log on as an administrator user to control the **Queue Management** window for the contributor user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Select a **Security Group Name**, and then click **Edit**. The **Modify a Security Group** page appears. For example, select the **Contributor** security group.
   The following image shows that the **Contributor** user cannot add a new queue even though this user can view the **Queue Management** window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

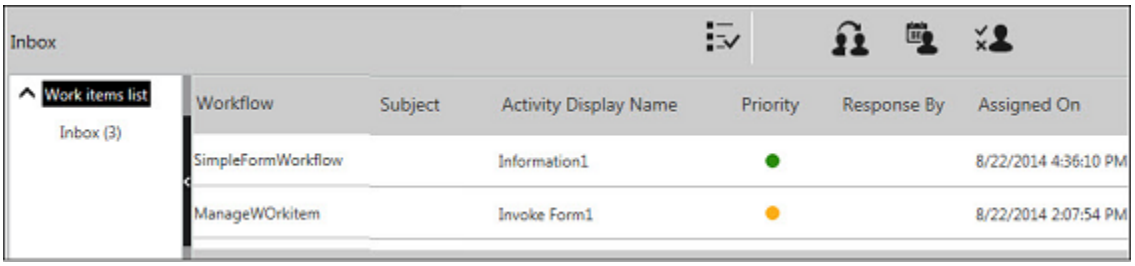


5. Uncheck the **Can View Queue Control** right to control to the access to view the Queue control.
6. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
7. Click **Yes** if you want to synchronize the changes to the below levels, or else click **Submit**.
8. Click **OK** on the confirmation window.
9. Now log on as a user who belongs to the Contributor security group.
10. Click **Queue Management**. A warning message about no permissions to view the Queue control appears.
11. Click **OK**.

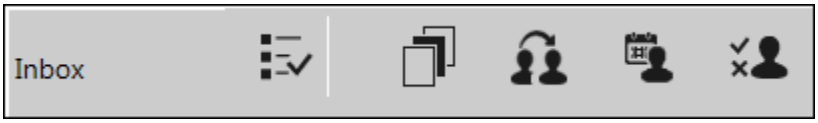
Secure Work Item Control

The work item control has options that enable users to manage how work items are delivered, when they are delivered, and to assign work items to alternate actors in their absence. An additional option enables queue owners to access the settings to their queues.

The following figure shows the Work items list window:



The following image shows the work item control toolbar where the user can secure the features of work item controls

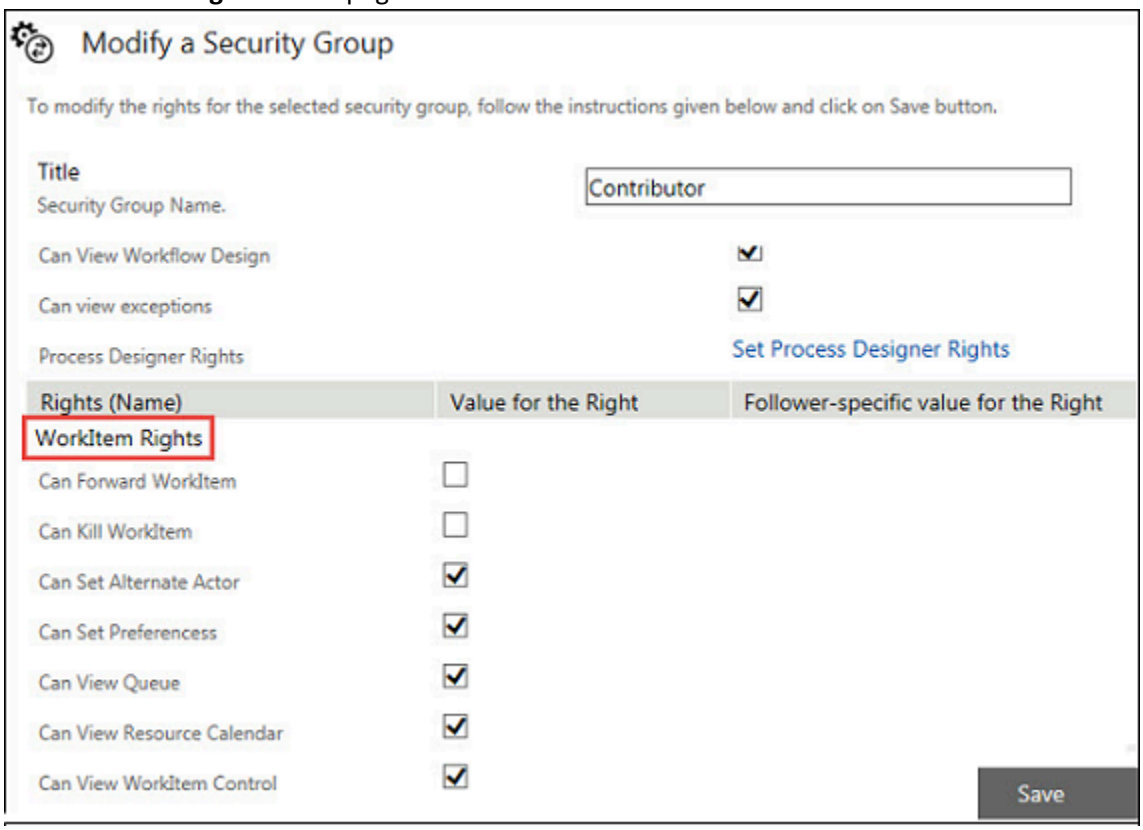


To view the work item control rights

1. Log on as the Administrator user.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Point to admin, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Select any **Security Group**, and then click **Edit**. The **Modify a Security Group** page appears. You can see the **Work Item Rights** in this page.



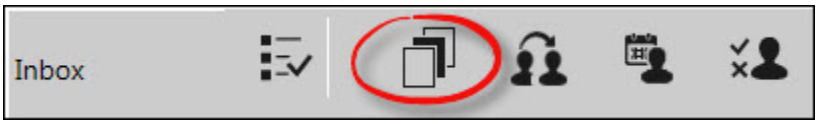
The following table summarizes the Rights available for a repository:

Work Item Rights Work Item	
Can Forward Work Item	This right controls access to forward the work item to another actor which is assigned. The Reassign and Delegate option will be displayed only when reassigned allowed is 'Yes' for that particular activity.
Can Kill Work Item	This right controls the access to kill work items.
Can Set Alternate Actor	This right controls the access to set alternate actor.
Can Set Preferences	This right controls the access to set preferences.
Can View Queue	This right controls the access to queue control.
Can View Resource Calendar	This right controls the access to view the resource calendar.
Can View Work Item Control	This right controls the access to view the work item control itself.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Secure Work Item Control - Show Queue

You can secure the **Show Queue** work item control in the Work items list window.

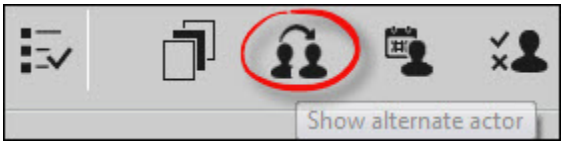


To secure the Show Queue control for the Contributor user

1. Log on as the Administrator user.
2. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Users**. The **Users** page appears.
3. On the action bar, click **New**. The **Users wizard** appears.
4. Create a user. For example, create a user called Tom.
5. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
6. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Groups** page appears.
7. On the Action bar, click **New** to map '*Tom*' to the **Contributor** security group. To know more about mapping users, see Create a Resource Mapping.
8. Log on as 'Tom' who is a contributor user now. Tom can now view the **Show Queue** control.
9. To secure the **Show Queue** control for the contributor user, log on to the Enterprise Console as the Administrator user.
10. Point to admin, and then click **Repository Settings**. The **Repository Settings** page appears.
11. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
12. Select any **Security Group**, and then click **Edit**. For example, select Contributor security group. The Modify a Security Group page appears. You can see the Work Item Rights in this page.
13. Uncheck the Can View Queue right to control the **Show Queue** control.
14. Click **Save**. The Synchronizing the **Security Group** dialog box appears.
15. Click **Submit**. As the work item control rights cannot be customized other than at the repository level, you do not need to synchronize.
16. Click **OK** in the confirmation window.
17. Log on as '*Tom*' user. Now, Tom cannot view the **Show Queue** control.

Secure Work Item Control - Show Alternate Actor

You can secure the **Show Alternate Actor** work item control in the Work items list window



To secure Show Alternate Actor control for the Contributor user

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears. Right-click the **Contributor** security group, and then click **Edit**. The **Modify a Security Group** page appears.
4. Uncheck the Can Set Alternate Actor right.
5. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
6. Click **Submit**. As the work item control rights cannot be customized other than at the repository level, there is no need to synchronize.
7. Click **OK** in the confirmation window.
8. Log on as a user who belongs to the Contributor security group. This user is now cannot view the **Show Alternate Actor** control.

Secure Work Item Control - Show Resource Calendar

You can secure the Show Resource Calendar work item control in the Work items list window.

To secure the Show Resource Calendar control for the Contributor user

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the **Contributor** security group, and then click **Edit**. The **Modify a Security Group** page appears.
5. Uncheck the **Can View Resource Calendar** right.
6. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
7. Click **Submit**. As the work item control rights cannot be customized other than at the repository level, there is no need to synchronize.
8. Click **OK** in the confirmation window.
9. Log on as a user who belongs to the Contributor security group. This user is now unable to view the **Show Resource Calendar** control.

Secure Work Item Control - Set Preferences

You can secure the **Set Preferences** work item control in the Work items list window

To secure the Set Preferences control for the Contributor user

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the **Contributor** security group, and then click **Edit**. The **Modify a Security Group** page appears.
5. Uncheck the **Can Set Preferences** right.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
7. Click **Submit**. As the work item control rights cannot be customized other than at repository level, there is no need to synchronize.
8. Click **OK** in the confirmation window.
9. Log on as a user who belongs to Contributor security group. This user is now unable to view the **Set Preferences** control.

Secure Work Item Control - View

To secure work item control for the Contributor user

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the **Contributor** security group, and then click Edit. The **Modify a Security Group** page appears.
5. Uncheck the **Can View Work Item Control** right.
6. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
7. Click **Submit**. As the work item control rights can not be customized other than at the repository level, there is no need to synchronize.
8. Click **OK** in the confirmation window.
9. Log on as the user who belongs to the Contributor security group.
10. In the **Enterprise Console**, click menu icon, select **Inbox**. You will get no permission related warning message.

Secure Work Item Control - Forward

To secure the re-assigning work item task to another actor for a Administrator user

1. Log on as the administrator user.
2. Assume that the 'SupportWF' workflow is designed and the **Re-Assign Allowed** property of the **Approval** Activity is set as **Yes**. Associate this workflow with an event. Once the associated event is triggered, the work item will be displayed in the **Inbox**.
3. Right-click on the 'SupportWF' task. Now the administrator user will be able to view the **Reassign Work Item** and **Delegate Work Item** options in the context menu.
4. To control these options, click **Repository Settings**, click **Advanced Settings** tab, click **Manage Security Groups**, and then select **Administrator** security group.
5. Click **Edit**. The **Modifying a Security Group** page appears.
6. Uncheck the **Can Forward Work Item** right to control the **Reassign Work Item** and **Delegate Work Item** options in the Work items list window.
7. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
8. Click **Cancel**. As the work item control rights cannot be customized other than at the repository level, there is no need to synchronize.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

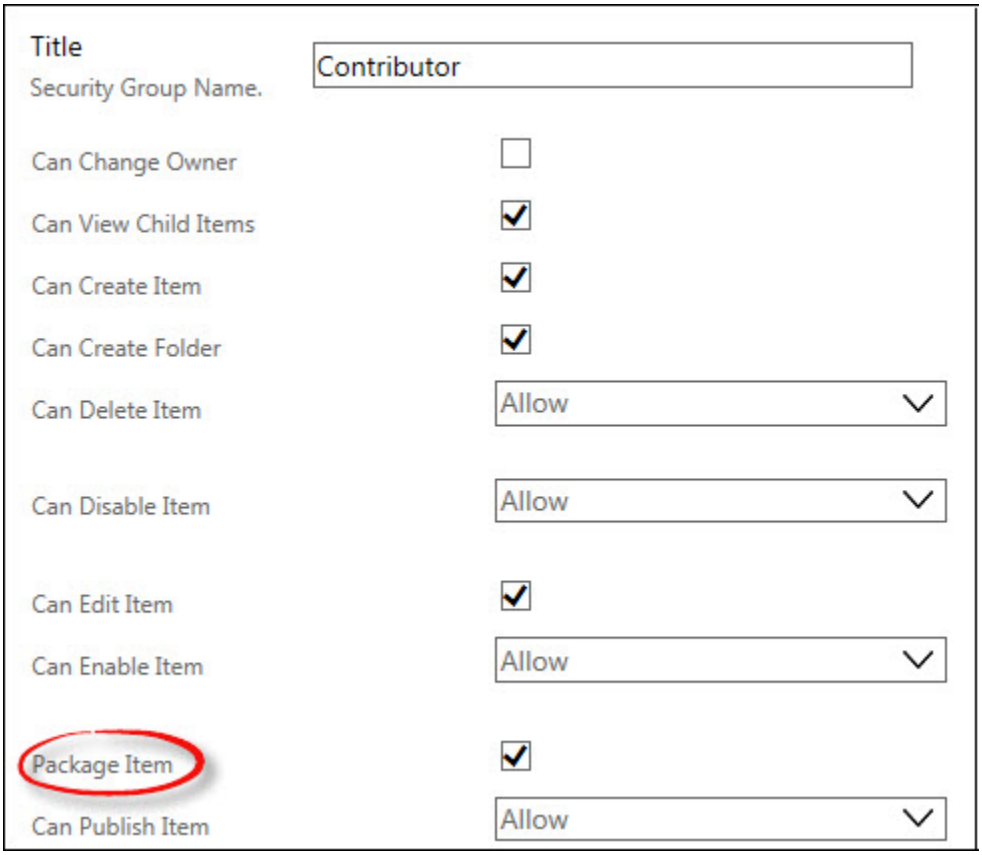
9. Click **OK** in the confirmation window.
10. In the **Enterprise Console**, click menu icon, and then select **Inbox**.
11. Right-click on the grid. The user will not get the **Reassign Work Item** and **Delegate Work Item** context menu options. The **Work Item Detail View** window of the selected workflow appears.

Secure Package

Use the Package feature to convert the tedious and complex task of deploying one or more artifacts (Database Lookup, List Lookup, and Database Connection) along with its dependencies from one environment/repository to another, easy and simple.

In addition to the rights specific mentioned in Secure Package Template List and Secure Actions in Manage Package, the Package Item right specific to the Package is available in the Generic List Actions rights.

Package Item: Controls the list items like workflow and forms from being added to the Package Template.



Secure Actions in Package Template

In the **Package Template** list, you can perform various actions such as, create, publish, generate, and export a Package Template.

- In the **Enterprise Console**, click menu icon, click **Package**, and then click **Package Template**.

  The **Package Template** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Using the Package Template List Rights, you can control the user from exporting, generating, and marking the package template as shipped.

The following table summarizes the Package Template List Rights:

Package Template List Rights	
Export	This right controls the access to export the package template.
Generate	This right controls the access to generate the package template.
Mark as Shipped	This right controls the access to mark the package template as shipped.

Control Generation of Package Template

In the Package Template list, you can control the user from generating the package template.

To generate a package template

1. Log on as a user who belongs to the **Contributor** security group.
2. Open **Package Template** page.
3. Select a published package template, and click **Generate**. A message about successful generation of a package template appears.
4. Click **OK**.

To control the user from generating the package template

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the **Contributor** security group, and then click **Edit**. The **Modify a Security Group** page appears.
5. In the **Package Template List Rights** section, uncheck the Generate right to restrict the contributor from generating the package template.
6. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
7. Click **Yes** if you want to synchronize the changes to the below levels, or else click **Submit**.
8. Click **OK** in the confirmation window.
9. To verify if the a user who belongs to the Contributor security group can generate a package template or not, log on as a user who belongs to the Contributor security group.
10. Open Package Template page.
11. Select a published package template, and click Generate. The following message appears:
You do not have rights to perform this action.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Control Package Template Export

In the Package Template list, you can control the user from exporting the package template.

To export a package template

1. Log on as a user who belongs to the Contributor security group.
2. Open **Package Template** page.
3. Select a generated package template, and click **Export**.

   The **Export** dialog box appears.
4. Select **Save**.

   A message about successful export of the package template appears.
5. Click **OK**.

To control the user from exporting the package template

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**.

   The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**.

   The **Security Group** page appears.
4. Right-click the **Contributor** security group, and then click **Edit**.

   The **Modify a Security Group** page appears.
5. In the **Package Template List Rights** section, uncheck the **Export** right to restrict the contributor from exporting the package template.
6. Click **Save**.

   The **Synchronizing the Security Group** dialog box appears.
7. Click **Yes** if you want to synchronize the changes to the below levels, else click **Submit**.
8. Click **OK** on the confirmation window.
9. To verify if the a user who belongs to the **Contributor** security group can export a package template or not, log on as a user who belongs to the Contributor security group.
10. Open the **Package Template** page.
11. Select a generated package template, and click **Export**.

    The following message appears:

    You do not have rights to perform this action.

Control Package Template Marking

In the Package Template list, you can control the user from marking a package template as shipped.

To mark a package template as shipped

1. Log on as a user who belongs to the **Contributor** security group.
2. Open **Package Template** page.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Select and right-click an exported package template, and click **Mark as Shipped**.
   The **Mark as Shipped** dialog box appears.
4. Click **Yes** to confirm, else click **No**.

To control the user from marking the package template as shipped

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**.
   The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**.
   The **Security Group** page appears.
4. Right-click the **Contributor** security group, and then click **Edit**.
   The **Modify a Security Group** page appears.
5. In the **Package Template List Rights** section, uncheck the **Mark As Shipped** right to restrict the contributor from marking the package template.
6. Click **Save**.
   The **Synchronizing the Security Group** dialog box appears.
7. Click **Yes** if you want to synchronize the changes to the below levels, else click **Submit**.
8. Click **OK** on the confirmation window.
9. To verify if the a user who belongs to the **Contributor** security group can mark a package template or not, log on as a user who belongs to the Contributor security group.
10. Open the **Package Template** page.
11. Select and right-click a published package template.
    The **Mark As Shipped** option is not available in the shortcut menu as the user has no rights.

Secure Actions in Manage Package

In the **Manage Package List** in the target environment, you can import, edit, run, install, update, and uninstall a package.

- In the **Enterprise Console**, click menu icon, click **Package**, and then click **Manage Package**.
  The **Manage Packages** page appears.

Using the Manage Package List rights, you can control the user from installing and uninstalling the package.

The following table summarizes the Manage Package List Rights:

Manage Package List Rights	
Run	This right controls the access to install the package.
Uninstall	This right controls the access to uninstall the package.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Control Package Install

In the Manage Package list, you can control the user from installing a package.

To install a package

1. Log on as a user who belongs to the **Contributor** security group.
2. Open **Manage Package** page.
3. Select a package, and click **Install**. The **Package Configurations** page appears.
4. Specify the required configurations.
5. Click **Install Package**. The **Package History** page appears, and then package installed successfully message appears.
6. Click **OK**.

To control the user from installing the package

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the **Contributor** security group, and then click **Edit**. The **Modify a Security Group** page appears.
5. In the **Manage Packages List Rights** section, uncheck the Install right to restrict the contributor from installing the package.
6. Click **Save**. The Synchronizing the Security Group dialog box appears.
7. Click **Yes**, if you want to synchronize the changes to the below levels, else click Submit.
8. Click **OK** on the confirmation window.
9. To verify if the a user who belongs to the Contributor security group can install a package or not, log on as a user who belongs to the Contributor security group.
10. Open **Manage Packages** page.
11. Select a package and click Install. The following message appears: *You do not have rights to perform this action.*

Control Package Uninstall

In the Manage Package list, you can control the user from uninstalling a package.

To uninstall a package

1. Log on as a user who belongs to the **Contributor** security group.
2. Open **Manage Package** page.
3. Select a package, and click **Uninstall**. The **Uninstall** page appears.
4. Specify the required configurations.
5. Click **Uninstall**. The **Package History** page appears, and then package uninstalled successfully message appears.
6. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To control the user from uninstalling the package

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the Contributor security group, and then click Edit. The **Modify a Security Group** page appears.
5. In the **Manage Packages List Rights** section, uncheck the **Uninstall** right to restrict the contributor from uninstalling the package.
6. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
7. Click **Yes**, if you want to synchronize the changes to the below levels, else click Submit.
8. Click **OK** on the confirmation window.
9. To verify if the a user who belongs to the Contributor security group can install a package or not, log on as a user who belongs to the Contributor security group.
10. Open **Manage Packages** page.
11. Select a package and click **Uninstall**. The following message appears:
You do not have rights to perform this action.

Secure Calendar Control

Calendars enable creation of multiple, multi-level, hierarchical calendars. In addition, you can define a number of lower-level calendars that inherit from one of the top-level calendars. As a result, an organization with a number of geographical locations can define an organization-wide main calendar, which will broadly define the work hours across the organization. It can then allow each location to define a separate calendar with its own work hours to accommodate the unique work hour requirements of that region. This calendar will inherit the settings of the main calendar and the custom settings can be added on top of them.

Additional levels of customized calendars can be configured by inheriting from the corresponding higher-level calendar. The lowest level of calendars that can be defined by extending this process are the Resource calendars.

To open Calendar Control

In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Calendar**. The **Calendar** page appears.

The **Calendar Control** user interface opens with a window that displays all the available calendars. If no calendar has been configured then this window shows only the Add New Calendar icon.

Using **Calendar Control** rights, the user can control the access to add a new calendar, delete existing calendar, view all the calendars, view base calendar and view the calendar control.

The following table summarizes the Queue Control rights available for a Repository:

Calendar Control Rights	
Can Add Calendar	This right controls the access to add a new calendar.
Can Delete Calendar	This right controls the access to delete existing calendar.
Can View All Calendars	This right controls the access to view all the calendars.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Calendar Control Rights	
Can View Base Calendar	This right controls the access to view the base calendar.
Can View Calendar Control	This right controls the access to view the calendar control.

Control Adding a New Calendar

The following example demonstrates how to secure adding a new calendar by using **Calendar Control** rights for a contributor user.

To control the access to add a new calendar for a contributor user

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the **Contributor** security group, and then click **Edit**. The **Modify a Security Group** page appears.
5. In the **Calendar Control Rights** section, uncheck the **Can Add Calendar** right to restrict the contributor from adding a calendar.
6. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
7. Click **Yes** if you want to synchronize the changes to the below levels, or else click Submit.
8. Click **OK** in the confirmation window.
9. To verify if the a user who belongs to the Contributor security group cannot add a new calendar, log on as a user who belongs to the Contributor security group.
10. Open **Calendar** page. The contributor user cannot view the **Add New Calendar** icon in the **Calendar Control** page.

Control Deleting an Existing Calendar

The following example demonstrates how to secure the deleting an existing calendar option using Calendar Control rights for a contributor user.

To control the access to delete a calendar for a contributor user

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the **Contributor** security group, and then click **Edit**. The **Modify a Security Group** page appears.
5. In the **Calendar Control Rights** section, uncheck the **Can Delete Calendar** right to restrict the contributor from deleting a calendar.
6. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
7. Click **Yes** if you want to synchronize the changes to the below levels, or else click Submit.
8. Click **OK** in the confirmation window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. To verify if the a user who belongs to the Contributor security group cannot delete a calendar, log on as a user who belongs to the Contributor security group.
10. Open **Calendar** page.
11. Click any existing calendar. The contributor user cannot view the **Delete** option in the **Calendar Control** page.

Control Viewing All Calendars

The following example demonstrates how to secure viewing all calendars using the Calendar Control rights for a contributor user.

To control the access to view all calendars for a contributor user

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the **Contributor** security group, and then click **Edit**. The **Modify a Security Group** page appears.
5. In the **Calendar Control Rights** section, uncheck the **Can View All Calendars** right to restrict the contributor from deleting a calendar.
6. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
7. Click **Yes** if you want to synchronize the changes to the below levels, or else click **Submit**.
8. Click **OK** in the confirmation window.
9. To verify if the a user who belongs to the Contributor security group cannot view all calendars, log on as a user who belongs to the Contributor security group.
10. Open **Calendar** page. The contributor user cannot view any calendar in the **Calendar Control** page.

Control Viewing Base Calendar

The following example demonstrates how to secure viewing the base calendar using Calendar Control rights for a contributor user.

To control the access to view the base calendar for a contributor user

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the Contributor security group, and then click **Edit**. The **Modify a Security Group** page appears.
5. In the **Calendar Control Rights** section, uncheck the **Can View Base Calendar** right to restrict the contributor from deleting a calendar.
6. Click **Save**. The Synchronizing the **Security Group** dialog box appears.
7. Click **Yes** if you want to synchronize the changes to the below levels, or else click Submit.
8. Click **OK** in the confirmation window.
9. To verify if the a user who belongs to the Contributor security group cannot view all calendars, log on as a user who belongs to the Contributor security group.
10. Open **Calendar** page.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

11. Click any existing calendar. The Contributor user cannot see the **View** button in the **Calendar Control** window. The user is unable to view the details of the base calendar.

Control Viewing Calendar Control

The following example demonstrates how to secure viewing the Calendar Control for a contributor user.

To control the access to view the base calendar for a contributor user

1. Log on as the Administrator user.
2. Point to **admin**, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the **Contributor** security group, and then click **Edit**. The **Modify a Security Group** page appears.
5. In the **Calendar Control Rights** section, uncheck the **Can View Calendar Control** right to restrict the contributor from deleting a calendar.
6. Click **Save**. The **Synchronizing the Security Group** dialog box appears.
7. Click **Yes** if you want to synchronize the changes to the below levels, or else click Submit.
8. Click **OK** in the confirmation window.
9. To verify if the a user who belongs to the Contributor security group cannot view all calendars, log on as a user who belongs to the Contributor security group.
10. In the **Enterprise Console**, click menu icon, click **Settings**, and then click **Calendar**. The following message appears: *You do not have permission for this control*

Secure BAM Control

You can implement security for the Business Activity Monitoring (BAM) control.

The BAM security has the following rights:

- BAM Report Rights
- Can View Global Reports
- Workflowscope OwnedWorkflow (Can View Owned Reports)
- Workflowscope_InitiatedWorkflow (Can View Triggered Reports)
- Workflowscope_Any (Can View Others Report)
- Can View BAM Control

Use BAM Security

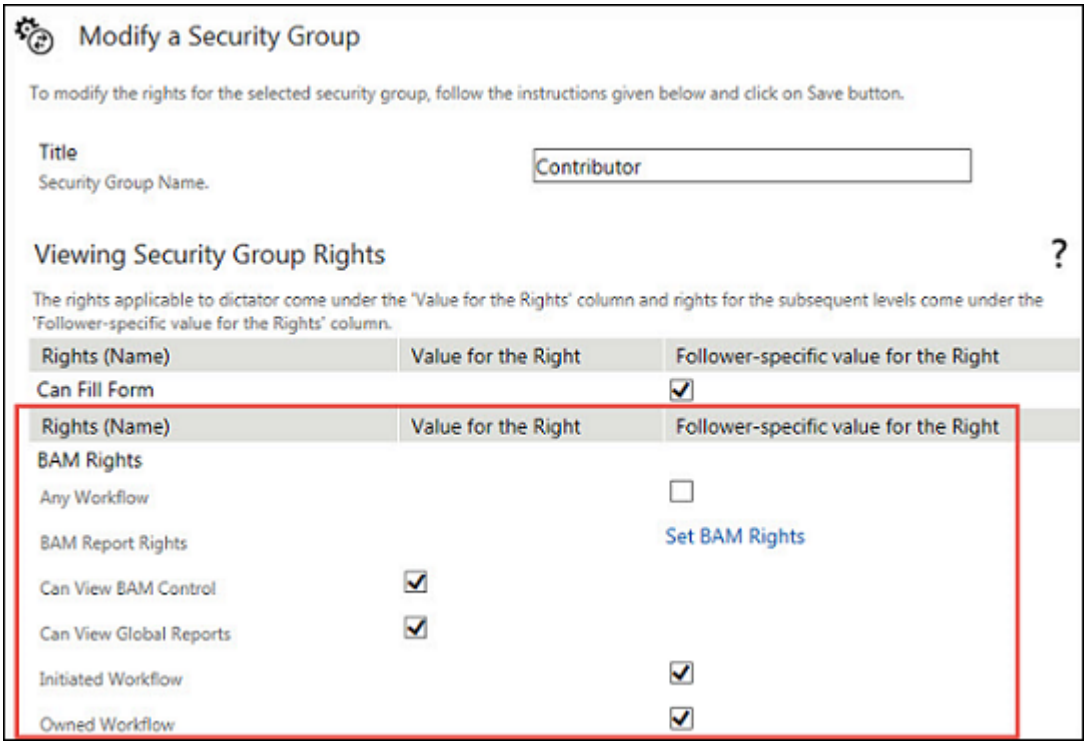
Only the administrator can grant or deny permissions.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To use the BAM Security

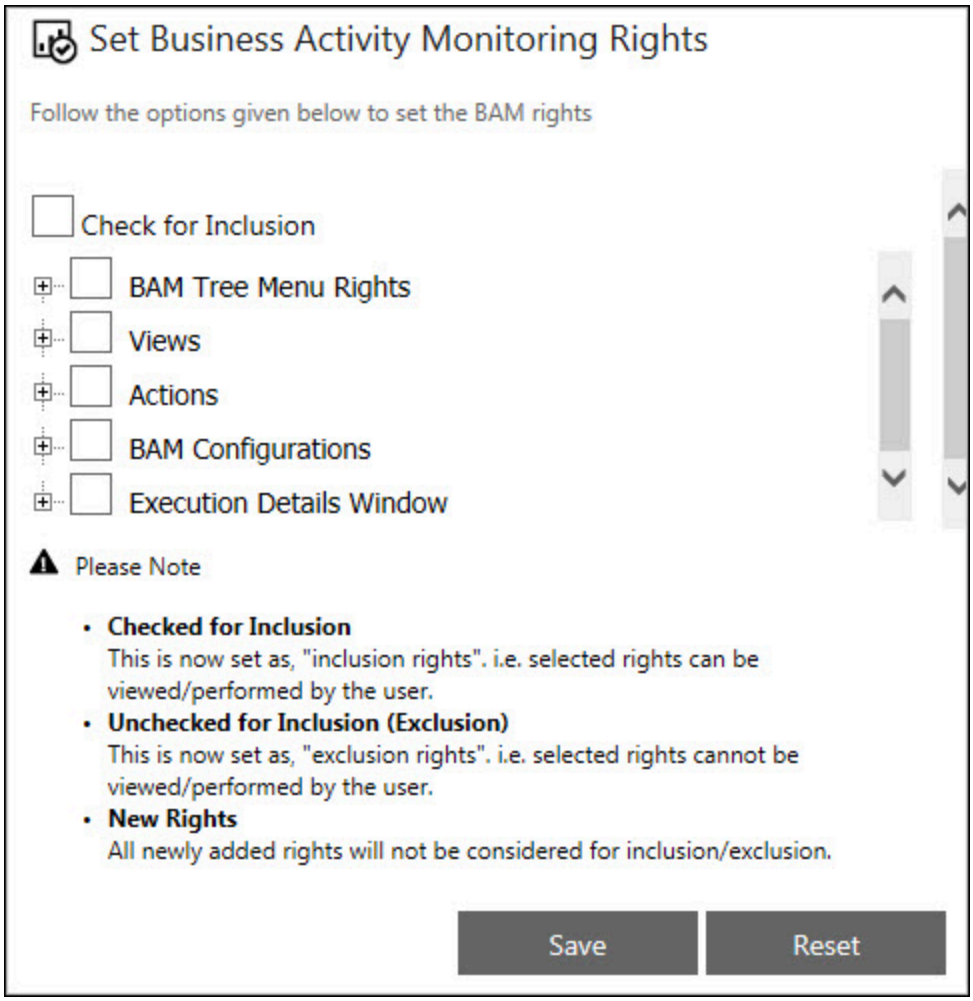
To set permissions at the Repository Level

1. Log on as the Administrator user.
2. Point to admin, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the security group to be customized, and then click **Edit**. The **Modify a Security Group** page appears. Modify the BAM rights.



5. Click the **Set BAM Rights** link to set the BAM rights. The **Set Business Activity Monitoring R**ights dialog box appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



6. Check **BAM Tree Manu Rights**, **Views**, **Actions**, and **BAM Configuration**. Note that the **Check for Inclusion** check box is available on the top. If you select the Check for Inclusion check box, the selected **BAM Tree Menu Rights**, **Views**, Actions, and BAM Configurations options are included in the rights value and saved in the database. If the Check for Inclusion check box is not selected, the selected BAM Tree Menu Rights, Views, Actions, and BAM Configurations options will be excluded from the rights value and all the unselected options will be saved in the database.
7. Click **Save**. The Set **Business Activity Monitoring Rights** dialog box closes.
8. To set the **Any Workflow**, **Can View BAM Control**, **Can View Global Reports**, **Initiated Workflow**, and **Owned Workflow rights**, select the corresponding check boxes, and then click **Save**.

To set BAM security at the List Item level

1. Go to the **Workflows** list.
2. Select an item and click **Security**. The security settings page appears.
3. Alternatively, select and right-click an item, and then click Security Settings from the shortcut menu.
4. In the **Advanced Settings** tab, click **Customize Security Group for this List**.
5. Right-click the security group to be customized, and then click **Edit**. The **Modify a Security Group** page appears. Modify the BAM rights.
6. Follow Step 5 to Step 8 to set BAM security at the List Item level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Default Rights for Different Security Groups

- **Administrator** - A user who belongs to the Administrator group can view the “Global Reports”, “Owned Reports”, “Triggered Reports” and “Others Report”.
  Only an “Administrator” can set the BAM Security either at the Repository level or List Item level.
- **Contributor** - A user who belongs to the Contributor group can view the “Global Reports”, “Owned Reports” and “Triggered Reports”.
- **Reader** - A user who belongs to the Reader group, can view only the “Triggered Reports”.
- **Limited Access** - A user who belongs to the Limited Access group, cannot view the BAM Configuration.

Default Rights for Different Security Groups

The following table lists default rights for different security groups.

Security Group Name	Can View Global Reports	Can View Owned Reports	Can View Triggered Reports	Can View Others Report	Can View BAM Control
Administrator	Yes	Yes	Yes	Yes	Yes
Contributor	Yes	Yes	Yes	No	Yes
Reader	No	No	Yes	No	Yes
Limited access	No	No	No	No	No

BAM Report Rights

The BAM Report Rights, specifies the rights for the following:

- BAM Tree Menu Rights
- Views
- Actions
- BAM Configuration

To set view BAM Report rights

1. Log on as the Administrator user.
2. Point to admin, and then click **Repository Settings**. The **Repository Settings** page appears.
3. In the **Advanced Settings** tab, click **Manage Security Groups**. The **Security Group** page appears.
4. Right-click the security group to be customized, and then click **Edit**. The **Modify a Security Group** page appears. Modify the BAM rights.
5. Click the **Set BAM Rights** link to set the BAM rights.

See BAM Tree Menu Rights, Views, Actions and BAM Configuration for more details.

The following table lists the rights and their scope to implement security for BAM:

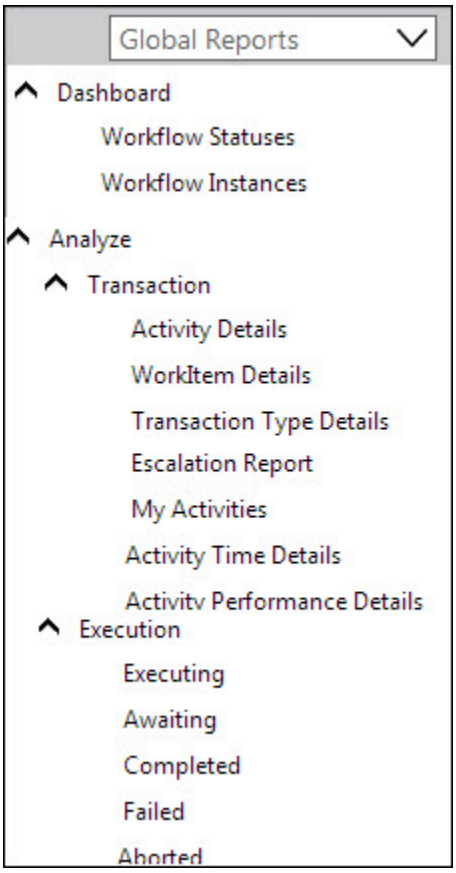
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Right Name	Scope
BAM Report Rights	ListSpecificForListItemLevel
Can View Global Reports	Repository
Workflow_OwnedWorkflow	ListSpecificForListItemLevel
Workflowscope_InitiatedWorkflow	ListSpecificForListItemLevel
Workflow_Any(Can View Others Report)	ListSpecificForListItemLevel
Can View BAM Control	Repository

BAM Tree Menu Rights

What is BAM Tree Menu Rights?

The BAM Tree Menu rights specify the nodes of the tree structure in the BAM Report. The tree structure is shown below.



AVEVA Work Tasks security provides the BAM Report Rights to control or restrict the access to these tree nodes (either privilege or restrict) in the BAM Report.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

How to Access BAM Tree Menu Rights?

To access or set the BAM Tree Menu Rights, the user must have administrative privileges.

To view the BAM Tree Menu rights, follow the steps given below:

1. Login as an Administrator user.
2. Select **Reports** from the **Enterprise Console** menu.
   The BAM Tree Menu Rights appears.

Purpose

AVEVA Work Tasks provides the **BAM Tree Menu Rights** to control the tree nodes (either privilege or restrict) in the BAM Report. In the BAM Tree Menu Rights, only those nodes for which a user has permission will be displayed in the BAM Report while the user is accessing the BAM Report.

Scenario

Scenario Description: Restricting the user from viewing some nodes in the tree structure of the BAM control.

Business Scenario: This scenario describes how to prevent the contributor user from viewing some nodes in the tree structure of the BAM control. Here, the Administrator user can restrict the contributor user from viewing the Workflow Statuses and Workflow Instances nodes of the tree structure in the BAM Control.

Roles: To understand this scenario effectively, assumed that there are two users in a repository. The first user belongs to the Administrator security group and the other user belongs to the Contributor security group.

1. admin - Will have Administrator access rights.
2. UserC - Will have Contributor access rights.

The following table depicts the users and their security groups:

User Name	Security Group
admin	Administrator
UserC	Contributor

Overview

Both the admin and UserC (Contributor) are able to see all the nodes in the tree structure of BAM Control.

The admin (Administrator) wants to restrict UserC (Contributor) to viewing some nodes in the tree structure of BAM Control. The same is depicted in the following table:

User Name	Report Menu Item	BAM Tree Menu Rights
admin	Yes	Yes
UserC	Yes	No

Scenario Solution:

To achieve the same, perform the following steps:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Login as the Contributor user. Here, login as 'UserC'.

   Before the security settings, the Contributor user can see all the nodes of the tree structure in the BAM Control. As mentioned above, the user must be an administrator to configure the security settings. As the admin user belongs to Administrator security group, this user can set the security settings to restrict the UserC to viewing some of the nodes in the tree structure in the BAM Control.
2. To restrict the UserC to viewing some of the nodes in the tree structure in BAM Control, login as an administrator user. Here, login as 'admin'.
3. Click **Repository Settings > Advanced Settings > Manage Security Groups**. You will be redirected to the Security Group list.
4. As the 'admin' user wants to restrict the contributor user, either select 'Contributor' and click **Actions > Edit** or right click 'Contributor' and then select Edit from the context menu.
5. A pop-up window appears with the assigned security rights. Click the **Set BAM Rights** link. Another pop-up window appears to set the BAM Rights.
6. Expand the **BAM Tree Menu Rights**.
7. The administrator user wants to prevent the other user from viewing the Workflow Statuses and Workflow Instances nodes under the Dashboard node. Unselect the **Check for Inclusion** check box. It means the selected rights should be excluded (restricted). So select the Workflow Statuses and Workflow Instances nodes under the Dashboard node.
8. After selecting the nodes to restrict, click **Save**. The window will be closed and you will be redirected to the security rights page. Click **Save**.
9. The **Synchronization** dialog box is displayed. Click **Yes** to synchronize the changes to the below level.
10. A message stating that “Security Group has been modified successfully” will appear. Click **OK**. Now the admin user has restricted the UserC (contributor) from viewing Workflow Statuses and Workflow Instances nodes of the tree structure in the BAM Control.
11. To check the same, login as a user who belongs to the contributor security group. Here, login as 'UserC'.
12. Select **Reports** from the **Enterprise Console** menu to access the BAM control. Observe that the Workflow Statuses and Workflow Instances nodes are not visible in the tree structure of the BAM control.

Views

The following are the three kinds of BAM available:

- Analysts View
- Progress View
- Execution View

AVEVA Work Tasks security provides the BAM Rights to control the access to or restrict these views (either privilege or restrict) in the BAM Report.

To open Process execution view, follow the step given below:

1. Login as the Administrator user.
2. Select **Reports** from the **Enterprise Console** menu.
3. In right panel, double-click any workflow to open the **Process execution view** window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Example: The following example demonstrates how to prevent a contributor user from seeing the **Execution View** in the **Process execution view** window.

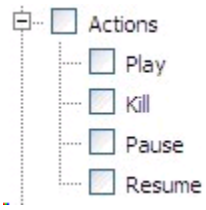
To achieve the same, follow the steps given below:

1. Login as a user who belongs to the Contributor security group.
2. Select **Reports** from the **Enterprise Console** menu.
3. In right panel, double-click any workflow to open the **Process execution view** window. Before the security settings, the contributor user is able to see all the views in the **Process execution view** window.
4. To prevent the contributor user from seeing the **Execution View** in the **Process execution view** window, login as the administrator user.
5. Click **Repository Settings > Advanced Settings > Manage Security Groups**.
6. Select the Contributor security group and click **Edit** or right-click Contributor and select **Edit** from the context menu.
7. Click **Set BAM Rights**.
8. Unselect the **Check for Inclusion** check box to exclude.
9. Select **Execution View**.
10. Click **Save**. The Reset button is used to reset the previous settings.
11. Click **Save** in the Modify Existing Security Group window.
12. Click **Yes** in the Synchronization window if you want to apply these settings to the below levels or else click **Submit**.
13. Click **OK** in the confirmation window.
14. Login as the contributor user to check.
15. Select **Reports** from the **Enterprise Console** menu.
16. Double-click any workflow instance in the right panel. The Process execution view window is displayed. Now this user is not able to see the execution view.

Actions

In BAM, there are actions to control the workflow execution. These actions are provided in the form of image buttons in the Process Execution window.

AVEVA Work Tasks provided BAM rights to control these actions.



See BAM Process Execution Viewfor more details.

Example: The following example demonstrates how to prevent a contributor user from viewing the Kill action option.

To do this, follow the steps given below:

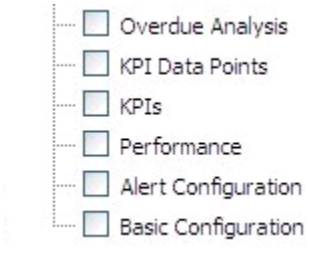
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Login as a user who belongs to the Contributor security group.
2. Select **Reports** from the **Enterprise Console** menu.
3. In the right panel, double-click any workflows to open the **Process execution view** window. Before the security settings, the contributor user is able to see all the actions in the **Process execution view** window.
4. To prevent the contributor user for viewing the **Kill** action in the **Process execution view** window, login as an administrator user.
5. Click **Repository Settings > Advanced Settings > Manage Security Groups**.
6. Select the Contributor security group and click **Edit** or right-click Contributor and select **Edit** from the context menu.
7. Click **Set BAM Rights**.
8. Unselected the **Check for Inclusion** check box to exclude.
9. Click **Set BAM Rights**.
10. Select **Kill** action.
11. Click **Save**.
12. The Reset button is used to reset the previous settings.
13. Click **Save** from the **Modify Existing Security Group** window.
14. Click **Yes** in the Synchronization window if you want to apply these settings to the below levels or else click **Submit**.
15. Click **OK** in the confirmation window.
16. Login as a contributor user to check.
17. Select **Reports** from the **Enterprise Console** menu.
18. Double-click any workflow instance in the right panel. The **Process execution view** window is displayed. Now this user is not able to see the **Kill** option in the tool bar.

BAM Configuration

In BAM, there are some configuration settings which allow the user to set Overdue Analysis, KPI(s), Performance, Alerts, and Time settings to schedule the BAM reports.

AVEVA Work Tasks provided BAM rights to control these configurations:



To open BAM configuration window, follow the step given below:

1. Login as an **Administrator** user.
2. Select **Workflows** from the **Enterprise Console** menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. Select any workflow and click **Design** on the ribbon bar or right-click and select **Design Workflow** from the context menu. This will redirect you to the **Process Designer** window.
4. Select **BAM Configuration** from the **Settings** menu. Before the security settings, the contributor user is able to see all the BAM configuration tabs in the **Process Designer** window.

The **BAM Configuration** window for the selected workflow is displayed.

Example: The following example demonstrates how to prevent a contributor user from viewing the Performance and Basic Configuration tabs in the BAM Configuration window.

To achieve the same, follow the steps given below:

1. Login as a user who belongs to the Contributor security group.
2. Select **Workflows** from the **Enterprise Console** menu.
3. Select any workflow and click **Design** from the ribbon bar or right-click and select **Design Workflow** from the context menu. This redirects you to the **Process Designer** window.
4. Select **BAM Configuration** from the **Settings** menu.
5. To prevent the contributor user from viewing the **Performance** and **Basic Configuration** tabs in the BAM Configuration window, login as the administrator user.
6. Click **Repository Settings > Advanced Settings > Manage Security Groups**.
7. Select the Contributor security group and click Edit or right-click Contributor and select Edit from the context menu.
8. Click **Set BAM Rights**.
9. Unselect the **Check for Inclusion** check box to exclude.
10. Click **Set BAM Rights**.
11. Select **Performance** and **Basic Configuration**.
12. Click **Save**. The Reset button is used to reset the previous settings.
13. Click **Save** in **Modify Existing Security Group** window.
14. Click **Yes** in the Synchronization window if you want to apply these settings to the below levels or else click **Submit**.
15. Click **OK** in the confirmation window.
16. Login as a contributor user.
17. Select **Workflows** from the **Enterprise Console** menu.
18. Right-click any workflow and select **Design Workflow** from the context menu. The **Process Designer** will be displayed.
19. From the **Settings** menu, select **BAM Configuration**. Now the logged in user is not able to view the Performance and Basic Configuration tabs in the BAM Configuration window.

Execution Details Window

What is Execution Details Window?

Execution Details Window displays the selected activity details like Step details, Variables, Contents and Content.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

How to Access Execution Details Window?

To access the **Execution Details Window**, click any activity in the **Process execution view** window.

The **Process execution view** window can be accessed in two ways:

From the **Enterprise Console** menu, select **Reports**. Double-click any instance of a workflow in the Execution section of the BAM Report. This will open the **Process Execution View** window.

or

From the **Enterprise Console** menu, select **Workflows**. Right-click any published workflow in the Workflows list and select **Publish and Test Run** from the context menu. This will open the **Process execution view** window.

Note: If you are not able to see the Reports menu item in the application menu, it means that the Can Consume right is set as false for whatever security group you belong to at the repository level.

Purpose

AVEVA Work Tasks implements security for this Execution Details Window. This window is accessible only if the user has the permission to view this page. Otherwise, the system will prompt the user stating “You do not have rights to view this page”.

This section explains the how to set or configure the security for the Execution Details Window.

Predefined Security Groups - Permissions

The default value of this right for the different predefined security groups is as follows:

Security Group Name	Have Permission to access execution details window?
Administrator	Yes
Contributor	Yes
Reader	No
Limited Access	No

To implement the security for the **Execution Details Window**, the user has to select/unselect the Execution Details Window check box in the BAM Rights section.

Inclusion & Exclusion Scenarios

If the “Check for Inclusion” check box not selected, the selected rights are excluded. Similarly, if the “Check for Inclusion” check box is selected then the selected rights are included. In the Inclusion scenario (When the "Check for Inclusion" check box is selected), if the “Execution Details Window” check box is selected then the user can view the Execution Details Window.

In Exclusion scenario (When the "Check for Inclusion" check box not selected), if the “Execution Details Window” check box is not selected then the user can view the Execution Details Window.

Note: As the **Execution Details Window** is accessible from the **Process execution view** window, the user must have at least one view selected in the Views of the BAM Rights section.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Users who belong to either the administrator or the contributor security groups are able to view the execution details. Users who belong to either the reader or limited access security group are not able to view the execution details. See Predefined Security Groups - Permissions

Scenario

Scenario Name: Restricting the user from viewing the Execution Details Window.

Business Scenario: This scenario describes how to prevent the contributor user from viewing the Execution Details Window in the Process execution view window.

Roles: To understand this scenario effectively, assumed that there are two users in a repository. One user belongs to the Administrator security group and the other user belongs to the Contributor security group.

1. **UserA** - Has Administrator access rights.
2. **UserC** - Has Contributor access rights.

The following table depicts the users and their security groups:

User Name	Security Group
UserA	Administrator
UserC	Contributor

Overview

As mentioned in Predefined Security Groups - Permissions, the users who belong to either the Administrator or Contributor security groups have the rights to view the execution details by default. So Both UserA & UserC have the right to view the execution details window.

Here UserA (Administrator) wants to prevent UserC (Contributor) from viewing the execution details window. The same is depicted in the following table:

User Name	Can View Execution Details
UserA	Yes
UserC	No

Scenario Solution:

To achieve the same, perform the following steps:

1. Login as a Contributor user. Here, login as 'UserC'.
2. From the **Enterprise Console** menu, select **Reports**. Before the setting security, UserC can view the Execution Details Window option in the Process execution view window.
3. To prevent the Contributor user from accessing the Execution Details Window, login as the Administrator user. Here login as 'UserA'.
4. Click **Repository Settings > Advanced Settings > Manage Security Groups**. You will be redirected to the Security Group list.
5. Select the **Contributor** security group and click **Edit** on the ribbon bar or right-click the security group and select Edit from the context menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. A pop-up window will appear with the assigned security rights. Click **Set BAM Rights**.
7. The BAM Rights window opens. Select the Execution Details Window check box and click Save. The BAM Rights window will be closed. Click **Save** in the Modify Existing Security Group window.
8. The **Synchronization** dialog box is displayed. Click **Yes** to synchronize the changes to the below level.
9. You will be prompted with a message stating that “Security Group has been modified successfully”. Click OK.
10. To check, login as a user who belongs to the contributor security group i.e 'UserC'.
11. From the **Enterprise Console** menu, select **Reports**. As UserC does not have the rights to view the execution details window.

Can View Global Reports

What is Global Reports?

In the BAM control, the Global Reports option in the drop-down is used to monitor all triggered workflow details like executing instances, awaiting instances, completed instances, failed instances and aborted instances at a time, instead of monitoring individual workflow details.

How to Access Global Reports?

To access Global Reports, click **Reports** from the **Enterprise Console** (application) menu. BAM Control will be displayed. If the user has the right to view **Global Reports**, the Global Reports option will selected by default in the drop-down.

Note: If you are not able to see the **Reports** menu item in the application menu, it means the **Can Consume** right is set as false for whatever security group you belong to at the repository level.

Purpose

AVEVA Work Tasks implemented a security right named **Can View Global Reports** at the repository level to control the Global Reports. This right can have either the true or false value (check box).

Predefined Security Groups - Right Value

The default value of this right for different the predefined security groups is as follows:

Security Group Name	Right Name	Right Value
Administrator	Can View Global Reports	True
Contributor	Can View Global Reports	True
Reader	Can View Global Reports	False
Limited Access	Can View Global Reports	False

If this right is set to **True** for a specific user only then will the user be able to see the **Global Reports** item in the drop-down of BAM control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For example, the users who belong to the either the administrator or the contributor security group are able to see the Global Reports item in the drop-down of BAM control.

If this right is set to **False** for a specific user, the user will not be able to see the Global Reports item in the drop-down of BAM control.

For example, the users who belong to either the Reader or Limited Access security group are not able to see the Global Reports item in the drop-down of BAM Control.

Note: Even though the **Can View Global Reports** right is set as **False**, the user will be able to view the **Reports** menu item in the application menu (Enterprise Console). This right will not restrict the user from viewing this menu but if the user tries to access the BAM Control, the user will be redirected to the BAM control page where the **Global Reports** option will not be available in the drop-down.

Scenario

Scenario Name: Preventing the user from accessing the Global Reports item in the drop-down of the BAM control.

Business Scenario: This scenario describes how to prevent the contributor user from accessing the Global Reports item in the drop down of the BAM control.

Roles: To understand this scenario properly, assumed that there are two users in a repository. One of the user belongs to the **Administrator** security group and the other user belongs to the **Contributor** security group.

1. UserA - Has Administrator access rights.
2. UserC - Has Contributor access rights.

The following table depicts the users and their security groups:

User Name	Security Group
UserA	Administrator
UserC	Contributor

Overview

As mentioned in Predefined Security Groups - Right Value, the users who belong to either the Administrator or Contributor security groups have the rights to view the BAM control by default. So Both UserA and UserC have the right to view the BAM control.

Here UserA (Administrator) wants to prevent the UserC (Contributor) from viewing the Global Reports item in the drop-down of the BAM control. The same is depicted in the following table:

User Name	Report Menu Item	Can View Global Reports
UserA	Yes	Yes
UserC	Yes	No

It means that UserC is able to see the **Reports** menu in the application menu (Enterprise Console) after logging in. But when UserC clicks the **Reports** menu item, the BAM Control page will be displayed without the **Global Reports** item in the drop-down.

Scenario Solution:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To arrive at a solution for this scenario, perform the following steps:

1. Login as a user who belongs to the Contributor security group. Here, login as 'UserC'.
2. From the **Enterprise Console** menu, select **Reports**. Before setting security, UserC can see the BAM Control and the Global Reports item in the drop-down.
3. To prevent the contributor user from viewing the Global Reports item in the drop-down of the BAM control, login as the administrator user. Here login as 'UserA'.
4. Click **Repository Settings > Advanced Settings > Manage Security Groups**. You will be redirected to the Security Group list.
5. Select the **Contributor** security group and click **Edit** on the ribbon bar or right-click the security group and select **Edit** from the context menu.
6. A pop-up window appears with the assigned security rights. Uncheck the **Can View Global Reports** right and click **Save**.
7. Click **Save**.
8. The **Synchronization** dialog box is displayed. Click Yes to synchronize the changes to the below level.
9. You will be prompted with a message stating that “Security Group has been modified successfully”. Click **OK**.
10. To check, login as a user who belongs to the contributor security group i.e 'UserC'.
11. From the **Enterprise Console** menu, select **Reports**. The user is redirected to the BAM Control page but the **Global Reports** item is not available in the drop-down.
12. The 'UserC' (Contributor security group user) does not have any owned workflow or workflows that are triggered by 'UserC'. The user will get the screen shown below while accessing BAM Control (**Reports** menu item in the application menu) since this user does not have the permission to view the workflows owned and triggered by others.

Observe that 'UserC' (Contributor security group user) does not have the permission to view others' workflows (Workflowscope_Any is false).

As per the above screen, the rights of “UseC” (Contributor security group) are as follows:

Right Name	**Right Value**
Workflowscope_Any (Others workflows)	False
Can View BAM Control	True
Can View Global Reports	False
Workflowscope_InitiatedWorkflow (Triggered workflows)	True
Workflowscopre_OwnedWrokflow (Owned workflows)	True

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Workflowscope_InitiatedWorkflow

What is Workflowscope_InitiatedWorkflow?

In the BAM control, apart from Global Reports, the logged in user can see the reports of the workflows that the user triggered.

Example: Consider that the user "John" triggered "Workflow1". Even though "John" is not the owner of this workflow, they will be able to see the details of "Workflow1" in the BAM Report because they triggered it.

You can prevent "John" from viewing "Workflow1" details in the BAM Report, even though they triggered it, using the right called **Workflowscope_InitiatedWorkflow**.

How to Access Initiated Workflows?

To access the initiated workflow reports, the logged in user must have the right to view triggered workflows i.e. the **Workflowscope_InitiatedWorkflow** right value must be **True** at the item level.

Note: If you are not able to see the **Reports** menu item in the application menu, it means the **Can Consume** right is set as false for whatever security group you belong to at the repository level.

Purpose

AVEVA Work Tasks implemented a security right named **Workflowscope_InitiatedWorkflow** at ListSpecificForListItemLevel to control the initiated (triggered) workflows. This right can have either the True or False (check box) value.

Predefined Security Groups - Right Value

The default value of this right for the different predefined security groups is as follows:

Security Group Name	Right Name	Right Value
Administrator	Workflowscope_InitiatedWorkflow	True
Contributor	Workflowscope_InitiatedWorkflow	True
Reader	Workflowscope_InitiatedWorkflow	True
Limited Access	Workflowscope_InitiatedWorkflow	False

If this right is set to **True** for a specific workflow, then the user who triggered any instance of the workflow can view the report of the workflow in the BAM Report.

For example, users who belong to either the administrator, contributor or reader security group are able to see their initiated (triggered) workflows in the BAM Report by default.

If the right is set to **False** for a workflow and the user who triggered it logs in, the user will not be able to view the report of the workflow in the BAM Report section.

For example, the users who belong to the Limited Access security group are not able to see their initiated (triggered) workflows in the BAM Report by default.

Note: Even though the **Workflowscope_InitiatedWorkflow** right is set as **False**, the user can view the **Reports** menu item in the application menu (Enterprise Console).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scenario

Scenario Name: Preventing the user from viewing the initiated workflows in the BAM control.

Business Scenario: This scenario describes how to prevent the Reader user from viewing the initiated workflows in the BAM control.

Roles: To understand this scenario clearly, assumed that there are two users in a repository. One user belongs to the Administrator security group and the other user belongs to the Reader security group.

1. admin - Has Administrator access rights.
2. UserR - Has Reader access rights.

The following table depicts the users and their security groups:

User Name	Security Group
admin	Administrator
UserR	Reader

Overview

As mentioned in Predefined Security Groups - Right Value, the users who belong to either the Administrator, Contributor or Reader security group have the rights to view initiated workflows in the BAM control by default. So both, 'admin' and 'UserR' have the right to view initiated workflows in the BAM control.

The users who belong to the Limited Access security group does not have the right to view initiated workflows in the BAM control by default.

Assume that there are two workflows named adminWF and readerWF (both created by admin – owner is admin) in the repository.

Assume that the 'adminWF' is triggered by the admin user and 'readerWF' is triggered by UserR (Reader security group).

After logging in, the admin user can see the details of both the workflows (adminWF and readerWF) apart from the Global Reports in the BAM Report. This is because the admin user has the rights to view both owned workflows and initiated workflows and also the right to view the Global Reports.

After logging in, UserR cannot see the Global Reports option in the drop-down. This is because UserR is a part of the Reader security group and does not have the permission to view Global Reports. UserR can see only the 'readerWF' in the BAM Report as this user has the right to view initiated workflow, by default.

Here admin (Administrator) wants to prevent the UserR (Reader) from viewing UserR's initiated workflow details in the BAM control. The same is depicted in the following table:

User Name	Report Menu Item	Workflowscope_InitiatedWorkflow
admin	Yes	Yes
UserR	Yes	No

It means that when UserR (Reader) logs in, UserR will be able to see the Reports item in the application menu (Enterprise Console). If UserR clicks the Reports menu item, the user will be redirected to the BAM page and will

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

not be able to see the initiated workflow in the BAM Report.

Scenario Solution:

To prevent the reader from viewing initiated workflow in the BAM Report, perform the following steps:

1. Login as a user who belongs to the Reader security group. Here, login as 'UserR'.
2. From the **Enterprise Console** menu, select **Reports**.
3. Here, 'UserR' is not the owner of readerWF even though UserR triggered readerWF.
4. To prevent 'UserR' from viewing readerWF details in the BAM Report, login as an administrator user. Here, login as 'admin'.
5. From the **Enterprise Console** menu, select **Workflows**. You will be redirecting to the **Workflows** list page.
6. As the 'admin' user wants to restrict access to the "readerWF" details in the BAM control (because readerWF is triggered by UserR), right click **'readerWF**' and select **Security Settings** from the context menu.
7. You will be redirected to the ListItem security settings page. Click the **Advanced Settings** tab.
8. Click **Enable List Item Specific Security**.
9. Select the **Security Group Customization** option in the Enable Security Wizard and click **Next**.
10. Click **Next**.
11. The user will be redirected to the **Security Group** customization at list item level page.
12. As 'admin' wants to restrict 'UserR' (Reader Security Group), select the Reader security group and click **Actions > Edit** or right click the **Reader security** group and select Edit from the context menu.
13. A pop-up window is displayed with assigned security rights. Uncheck the Workflowscope_InitiatedWorkflow right and click Save.
14. Before clicking **Save**, ensure that the required BAM tree menu rights are set.
15. The **Synchronization** dialog box is displayed. Click **Yes** to synchronize the changes to the below levels.
16. You will be prompted with a message stating that "*Security Group has been modified successfully*." Click **OK**.
17. To check the same, login as a user who belongs the **Reader** security group. Here, login as 'UserR'.
18. To access the BAM control, select Reports from the **Enterprise Console** menu. The 'UserR' will be redirected to the BAM control page and will not be able to view the 'readerWF' details in the BAM control.

Workflowscope_OwnedWorkflow

What is Workflowscope_OwnedWorkflow?

In BAM control, apart from Global Reports, the logged in user is able to see the reports of the workflows that the logged in user triggered.

Example: Consider that user "Tim" is the owner of the "workflow1" and the user "John" logged-in and triggered the "workflow1". The user "John" only can view the report of the "workflow1" in the BAM Report. Even though "Tim" is the owner of "workflow1", if they log in, they cannot view the report of "workflow1" in the BAM Report.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

How to Access Owned Workflow's?

To access the owned workflow reports, the logged in user must have the right to view owned workflow i.e. the **Workflowscope_OwnedWorkflow** right value must be **True** at the item level.

Note: If you are not able to see the **Reports** menu item in the application menu, it means the **Can Consume** right is set as false for whatever security group you belong to at the repository level.

Purpose

AVEVA Work Tasks implemented a security right named **Workflowscope_OwnedWorkflow** at **ListSpecificForListItemLevel** to control the owned workflows. This right can have either the True or False (check box) value.

Predefined Security Groups - Right Value

The default value of this right for the different predefined security groups is as follows:

Security Group Name	Right Name	Right Value
Administrator	Workflowscope_OwnedWorkflow	True
Contributor	Workflowscope_OwnedWorkflow	True
Reader	Workflowscope_OwnedWorkflow	False
Limited Access	Workflowscope_OwnedWorkflow	False

If this right is set to **True** for a workflow for a specific user and the user is the owner of the workflow and has not triggered any instances of the workflow, the user can see the workflow details in the BAM Report.

For example, users who belong to either the administrator or contributor security group are able to see their owned workflows in the BAM Report by default.

If this right set to **False** for a workflow for a specific user and the user is the owner of the workflow and has not triggered any instance of the workflow, the user will not be able to see details of the workflow in the BAM Report (despite being the owner of the workflow).

For example, users who belong to either the Reader or Limited Access security group are not able to see their owned workflows in the BAM Report by default.

Note: Even though the **Workflowscope_OwnedWorkflow** right is set as **False**, the user can view the **Reports** menu item in the application menu (Enterprise Console).

Scenario

Scenario Name: Privileging the user to view owned workflows in the BAM control.

Business Scenario: This scenario describes how to provide access to the Reader user to view owned workflows in the BAM control.

Roles: To understand this scenario clearly, assume that there are two users in a repository. One user belongs to the Administrator security group and the other user belongs to the Reader security group.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. admin - Has Administrator access rights.
2. UserR - Has Reader access rights.

The following table depicts the users and their security groups:

User Name	Security Group
admin	Administrator
UserR	Reader

Overview

As mentioned in Predefined Security Groups - Right Value, the users who belong to either the Administrator or Contributor security group have the rights to view owned workflows in the BAM control by default. So the 'admin' user has the right to view owned workflow in the BAM control.

The users who belong to either the Reader or Limited Access security group do not have the right to view owned workflows in the BAM control by default. So 'UserR' does not have the right to view owned workflow in the BAM control.

Assume that there are two workflows named adminWF (created by admin – owner is admin) and readerWF (created by UserR (Reader) – owner is UserR) in the repository.

Assume that both the workflows are triggered by 'admin' user.

After logging in, 'admin' can see the both workflow names and the Global Reports item in the drop-down of the BAM control.

After logging in, 'UserR' cannot see the Global Reports option in the drop-down (because UserR is a Reader and does not have permission to view the Global Reports) and readerWF details in the BAM Control (as UserR is the owner of readerWF).

Here, admin (Administrator) wants to privilege UserR (Reader) to view UserR's owned workflow details in the BAM control. The same is depicted in the following table:

User Name	Report Menu Item	Workflowscope_OwnedWorkflow
admin	Yes	Yes
UserR	Yes	Yes

It means that after logging in, UserR (Reader) is able to see the **Reports** item in the application menu (Enterprise Console). If UserR clicks the Reports menu item, the user will be redirected to the BAM page and be able to see owned workflow in the BAM Report.

Scenario Solution:

To privilege the reader to view owned workflow in the BAM Report, perform the following steps:

1. Login as a user who belongs to the Administrator security group. Here, login as 'admin'.
2. From the **Enterprise Console** menu, select **Workflows**. You will be redirected to the Workflows list page.
3. As admin wants to give the privilege for the 'readerWF' (UserR is the owner of this workflow), right-click

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

'readerWF' and select **Security Settings** from the context menu.

4. You will be redirected to the ListItem security settings page. Click the **Advanced Settings** tab.
5. Click **Enable List Item Specific Security**.
6. Select the **Security Group Customization** option in the **Enable Security Wizard** and click **Next**.
7. Click **Next**.
8. The user will be redirected to the **Security Group** customization at list item level page.
9. To give privilege to 'UserR' (Reader security group), select the **Reader** security group and click **Actions > Edit** or right-click the **Reader** security group and select **Edit** from the context menu.
10. A pop-up window is displayed with the assigned security rights. Check the **Workflowscope_OwnedWorkflow** right and click **Save**.
11. Before clicking **Save**, ensure that the required **BAM** tree menu rights are set.
12. The **Synchronization** dialog box is displayed. Click **Yes** to synchronize the changes to the below level.
13. You will be prompted with a message stating that "**Security Group has been modified successfully**." Click **OK**.
14. To check the same, login as a user who belongs the **Reader** security group. Here, login as 'UserR'.
15. To access the **BAM** control, select **Reports** from the **Enterprise Console** menu. Now the 'UserR' will get the BAM control page and will be able to view owned workflow.

Workflowscope_Any

What is Workflowscope_Any?

In BAM control, apart from the Global Reports, the logged in user can see the reports of the workflows that the logged in user triggered and owns.

Example: Consider that user "Tim" is not the owner of the "workflow1" and has not triggered "workflow1". But we want "Tim" to view "workflow1" details in the BAM Report.

How to Access Others Workflows?

To access others' workflow reports, the logged in user must have the right to view others workflows i.e. the **Workflowscope_Any** right value must be true at the item level.

Note: If you are not able to see the **Reports** menu item in the application menu, it means the **Can Consume** right is set as false for whatever security group you belong to at the repository level.

Purpose

AVEVA Work Tasks implemented a security right named **Workflowscope_Any** at "ListSpecificForListItemLevel" to control others owned or triggered workflows. This right can have either the True or False (check box) value.

Predefined Security Groups - Right Value

The default value of this right for the different predefined security groups is as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Security Group Name	Right Name	Right Value
Administrator	Workflowscope_Any	True
Contributor	Workflowscope_Any	False
Reader	Workflowscope_Any	False
Limited Access	Workflowscope_Any	False

If this right is set to **True** for a specific workflow for a user (who is not the owner or has not triggered any instances of the workflow), then that user is able to see the report of the workflow in the BAM Report after logging in.

For example, the users who belong to the administrator security group are able to see others owned or triggered workflows in the BAM Report by default.

If this right is set to **False** for specific workflow for a user (who is not the owner or has not triggered any instances of the workflow), then that user cannot see the report of the workflow in the BAM Report after logging in.

For example, users who belong to either the contributor, reader or limited access security group are not able to see others owned or triggered workflows in the BAM Report by default.

Note: Even though the **Workflowscope_Any** right is set as **False**, the user is able to view the **Reports** menu item in the application menu (Enterprise Console).

Scenario

Scenario Name: Privileging the user to view others owned workflows in the BAM control.

Business Scenario: This scenario describes how to privilege the Reader user to view the workflows owned by the contributor user in the BAM control.

Roles: To understand this scenario clearly, assumed that there are three users in a repository. The first user belongs to the **Administrator** security group, the second user belongs to the **Reader** security group, and the third user belongs to the **Contributor** security group.

1. admin - Has Administrator access rights.
2. UserR - Has Reader access rights.
3. UserC - Has Contributor access rights.

The following table depicts the users and their security groups:

User Name	Security Group
admin	Administrator
UserR	Reader
UserC	Contributor

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Overview

As mentioned in the Predefined Security Groups - Right Value, the users who belong to the Administrator security group have the rights to view others owned and triggered workflows in the BAM control by default. So 'admin' has the right to view others owned and triggered workflows in the BAM control.

As mentioned in Predefined Security Groups - Right Value, the users who belong to either the Contributor, Reader or Limited Access security group do not have the right to view others owned and triggered workflow in the BAM control by default. So both UserR (Reader security group) and UserC (Contributor security group) cannot view others owned or triggered workflows in the BAM control.

Assume that a workflow 'GenericWF' is created by UserC (Contributor security group) and triggered by UserC.

The Admin user can see the 'GenericWF' details in the BAM Report by default. As mentioned in Predefined Security Groups - Right Value, users who belong to the Administrator security group have the right to view others owned and triggered workflows in the BAM control by default.

UserC (Contributor) can see the 'GenericWF' details in the BAM Report by default because UserC is the owner of that workflow and triggered it. Users who belong to the Contributor security group have the right to view the owned workflows (Workflowscope_OwnedWorkflow) and initiated workflows (Workflowscope_InitiatedWorkflow).

UserR (Reader) is not able to see the 'GenericWF' details in the BAM Report by default. As mentioned in Predefined Security Groups - Right Value, users who belong to either the Contributor, Reader or Limited access security groups do not have the rights to view others owned and triggered workflows in the BAM Report by default.

User Name	Report Menu Item	Workflowscope_Any
admin	Yes	Yes
UserC	Yes	Yes
UserR	Yes	Yes

It means after logging in, UserR (Reader) will be able to see the Reports item in the application menu (Enterprise Console). On clicking Reports, UserR will be redirected to the BAM page and will be able to see others owned workflow in the BAM Report.

Scenario Solution

To achieve the same, perform the following steps:

1. Login as an Administrator user. Here, login as 'admin'. Before the security settings, the 'admin' user can see the 'GenericWF' details in the BAM Report.
2. Login as a Contributor user. Here, login as 'UserC'. Before the security settings, the Contributor user see the 'GenericWF' details in the BAM Report.
3. Login as a Reader user. Here login as 'UserR'. Before the security settings, the Reader user cannot view the 'GenericWF' details in the BAM Report. The screen given below will be displayed as UserR (Reader) does not have the right to view Global Reports and there are no workflows owned and triggered by UserR.
4. As 'admin' wants to privilege 'UserR' to view the GenericWF in the BAM Report, login as 'admin'.
5. From the **Enterprise Console** menu, select **Workflows**. You will be redirected to the Workflows list page.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. As the 'admin' user wants to give the privilege to 'UserR' to view 'GenericWF' details in the BAM control, right-click '**GenericWF'** and select **Security Settings** from the context menu.
7. You will be redirected to the ListItem security settings page. Click the **Advanced Settings** tab.
8. Click **Enable List Item Specific Security**.
9. Select the **Security Group Customization** option in the **Enable Security Wizard** and click **Next**.
10. Click **Next**.
11. You will be redirected to the **Security Group** customization at list item level page.
12. To give privilege to 'UserR' (Reader Security Group), select the **Reader** security group and click **Actions > Edit** or right-click the **Reader** security group and select **Edit** from the context menu.
13. A pop-up window is displayed with the assigned security rights. Check the **Workflowscope_Any** right and click **Save**.
14. Before clicking **Save**, ensure that the required BAM tree menu rights are set.
15. The **Synchronization** dialog box is displayed. Click **Yes** to synchronize the changes to its below level.
16. You will be prompted with a message stating that *"Security Group has been modified successfully."* Click **OK**.
17. To check the same, login as a user who belongs the Reader security group. Here, login as 'UserR'.
18. To view "GenericWF" details in the BAM control, select **Reports** from the **Enterprise Console** menu. Now the 'UserR' will get the BAM control page and will be able to view owned workflow.

Can View BAM Control

What is BAM Control?

BAM control is used to monitor all triggered workflows details like executing instances, awaiting instances, completed instances, failed instances and aborted instances and so on.

How to Access BAM Control?

To access BAM control, click Reports from the Enterprise Console (application) menu.

Note: If you not able to see the Reports menu item in the application menu, it means the Can Consume right is set as false for the security group you belong to at the repository level.

Purpose

AVEVA Work Tasks has implemented a security right named Can View BAM Control at the repository level to control the BAM Control. This right can have either the true or false (check box) value.

Predefined Security Groups - Right Value

The default value of this right for the different predefined security groups is as follows:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Security Group Name	Right Name	Right Value
Administrator	Can View BAM Control	True
Contributor	Can View BAM Control	True
Reader	Can View BAM Control	True
Limited Access	Can View BAM Control	False

If this right is set to **True** for a specific user, only then that user can see the BAM control. For example, users who belong to either the Administrator, Contributor or Reader security groups are able to see the BAM control.

If this right set to **False** for a specific user, the user is not able to see the BAM control. For example, users who belong to the Limited Access security group are not able to see the BAM control.

Note: Even though the Can View BAM Control right is set as False, the user can view the Reports menu item in the application menu (Enterprise Console). This right will not restrict the user from viewing this menu but if the user tries to access the BAM Control, the user will be prompted with a message stating that “You do not have rights to view BAM Control”.

Scenario

Scenario Name: Preventing the user from accessing the BAM Control.

Business Scenario: This scenario describes how to prevent the contributor user from accessing the BAM control.

Roles: To understand this scenario clearly, assumed that there are two users in a repository. One user belongs to the Administrator security group and the other user belongs to the Contributor security group.

1. UserA - Has Administrator access rights.
2. UserC - Has Contributor access rights.

The following table depicts the users and their security groups:

User Name	Security Group
UserA	Administrator
UserC	Contributor

Overview

As mentioned in Predefined Security Groups - Right Value, users who belong to either the Administrator, Contributor or Reader security groups have the rights to view the BAM control by default. So both UserA and UserC have the right to view the BAM control.

Here, UserA (Administrator) wants to prevent the UserC (Contributor) from viewing the BAM control. The same is depicted in the following table:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

User Name	Report Menu Item	Can View BAM Control
UserA	Yes	Yes
UserC	Yes	No

It means that after logging in, UserC (Contributor) is able to see the Reports menu in the application menu (Enterprise Console). This user has to be prevented from accessing the BAM Control when the user clicks the Reports menu item.

Scenario Solution:

To arrive at a solution for this scenario, perform the following steps:

1. Login as a user who belongs to the Contributor security group. Here, login as 'UserC'.
2. From the **Enterprise Console** menu, select **Reports**.
3. Before setting security, UserC can see the BAM Control.
4. To prevent the Contributor user from accessing the BAM Control, login as an Administrator user. Here, login as 'UserA'.
5. Click **Repository Settings > Advanced Settings > Manage Security Groups**. You will be redirected to the Security Group list.
6. Select the **Contributor** security group and click **Edit** from the ribbon bar or right-click the Contributor security group and select **Edit** from the context menu.
7. A pop-up window will appear with assigned security rights. Uncheck the **Can View BAM Control** right and click Save.
8. Click **Save**.
9. The **Synchronization** dialog box is displayed. Click **Yes** to synchronize the changes to the below level.
10. You will be prompted with a message stating that “Security Group has been modified successfully.” Click **OK**.
11. To check, login as a user who belongs to the Contributor security group i.e 'UserC'.
12. From the **Enterprise Console** menu, select **Reports** menu. The following message appears: *You do not have rights to view the BAM control.*

Secure Process Designer

AVEVA Work Tasks security allows the user to secure the Process Designer. The security mechanism provides access to the Process Designer, rights to users and informs the user about who can access the Process Designer rights.

- Users can avail the security feature to secure the Process Designer in their business process management. The Process Designer can be secured in different ways. Users can control all most all features in the Process Designer.
- You can provide restricted access to a set of users to view, new folder, delete, design, publish, unpublish, associate and security settings features in the Process Designer.
- You can provide restricted access to a set of users to Save As different versions, Disable/Enable Item, View Associations features in the Process Designer.
- You can provide restricted access to a set of users to access all activities category (Human, Integration,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Communication) and can restrict them to some activities and properties in the activities category.

- You can give control to a set of users to access menus (File, Edit, Publish, Tools, Settings, Wizard, Test Run and Help) and their sub-menus.

Process Designer Rights

To set Process Designer rights, perform the following steps:

1. Login to the **Enterprise Console**.
2. Click **Repository Settings**.
3. Click **Advance Settings** > **Manage Security Group**.
4. Click the required security group for customization and click **Actions** > **Edit** from the ribbon bar or right-click the security group and select Edit from the context menu.
5. For example, select Administrator security group and select **Edit** or right-click this group and select **Edit** from the context menu. The **Modify a Security Group** page appears.
6. Scroll down and click the **Set Process Designer Rights** link to set the Process Designer rights. Once you click the **Set Process Designer Rights** link, the **Set Process Designer Rights** page appears where you can set the rights for actions and menus. This page displays all the action categories and menus that are available for the repository.
7. Select the **Check for Inclusion** check box to include the selected categories, activities, properties and menu items or unselect the **Check for Inclusion** check box for excluding the selected categories, activities, properties and menu items for the **Process Designer**.
8. After selecting the **Categories**, **Activities**, **Properties** and Menus, click **Save** to include/exclude the rights for the Process Designer.
9. Click **Save**. This will redirect you to the main Security Group window.

   Note: The Reset button is used to reset the previous settings.
10. Click **Save** to reflect the changes made in **Set Process Designer Rights** window. The main window gets closed and the **Synchronization** window is displayed.
11. Click **Yes** to synchronize the changes to the below level or click the Submit button to not to synchronize.
12. An alert message will pop up saying *Security Group has been modified successfully*. Click **OK** to close this window.
13. Open the **Process Designer** window to view the effects of the security settings on the action and menu rights.

    Note: If the **Check for Inclusion** option is selected, the selected action rights and menu rights will be available in the **Process Designer** window. The user will have access to all the actions and menus which are selected. If the **Check for Inclusion** option is not selected, the selected action rights and menu rights **will not be available** in the **Process Designer** window. The User will not have access to the actions and menus which are selected. See Inclusion and Exclusion for more information.

The following are the Process Designer rights:

1. **Action Rights** - Rights for controlling the categories, actions and properties. The user can control the categories, actions, properties that are displayed in the Process Designer. See Action Rights for more details.
2. **Menu Rights** - Rights for controlling the Process Designer menus like can export, can import, can validate

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

definition, can set work items fields. See Menu Rights for more details.

Inclusion and Exclusion

Inclusion-Exclusion is the facility provided for the user to grant privileged access or restrict the access that users have, to the selected rights. It is provided in the form of a check box in the BAM Rights/Process Designer Rights window. It decides whether the selected rights have to be included (privileged) or excluded (restricted) for the security group.

Examples:

Inclusion:

Assume that the **Check for Inclusion** is selected. This means the selected rights are included (privileged) for the selected security group user.

In this example, categories like Human Activities, Integration Activities, Engine Activities & Scheduler Activities are included (privileged) and some activities from Human Activities category like Approval, Check-List, Choice and Information are included.

To achieve the same, perform the following steps

1. Login to the **Enterprise Console**.
2. Click **Repository Settings**.
3. Click **Advance Settings** > **Manage Security Group**.
4. Select **New** or **Edit** a security group. For example, select **Administrator** security group and select **Edit**.
5. Click **Set Process Designer Rights**.
6. Select the **Check for Inclusion** check box.
7. Select **Human Activities**, **Integration Activities**, **Engine Activities** & **Scheduler Activities** categories. Also select some **Human Activities** like **Approval**, **Check-List**, **Choice** and **Information**.
8. Click **Save**.
9. Click **Save** to reflect the changes made in the **Set Process Designer Rights** window.
10. Click **Yes** to synchronize the changes to the below level or click **Submit** if you do not want to synchronize.
11. An alert message will pop up saying *Security Group has been modified successfully*. Click **OK** to close this window.
12. To see the effect, open the **Process Designer** window. Only the selected categories and selected activities from the **Human Activities** are included in the **Activities** drop-down list.

Exclusion

Assume that the **Check for Inclusion** is box is not selected. This means the selected rights are excluded (restricted) for the selected security group user.

The this example, categories like Human Activities, Integration Activities, Engine Activities & Scheduler Activities are excluded (restricted) and some activities from the Human Activities category like Approval, Check-List, Choice and Information are excluded.

To achieve the same, perform the following steps:

1. Navigate to the security group definition and open Set Process Designer Rights window to set the rights.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

(Perform the steps 1 to 5 from the above 'Inclusion' example).

2. Unselect the **Check for Inclusion** check box.
3. Select **Human Activities**, **Integration Activities**, **Engine Activities & Scheduler Activities** categories. Also select some Human activities like **Approval**, **Check-List**, **Choice and Information**.
4. Perform the steps 8 to 11 from the above 'Inclusion' example.
5. To see the effect, open the **Process Designer** window. The selected categories and activities from the Human Activities are excluded from the **Activities** drop down list.

Note: If a new Category, Activity, Property and Menu is added after setting the **Process Designer Rights** then it will not be included and will not get affected.

Action Rights

This option enables the user to select the actions (Categories, Activities and Properties) to exclude from the Process Designer. The selected actions will not be rendered when the Process Designer is loaded.

Similar to the one for Categories, there are options to select the Activities and Properties to exclude from the Process Designer. The selected Activities and Properties will not be rendered when the Process Designer is loaded.

See Also:

- Securing Categories
- Securing Activities of Categories
- Securing Properties of Activities

Secure Categories

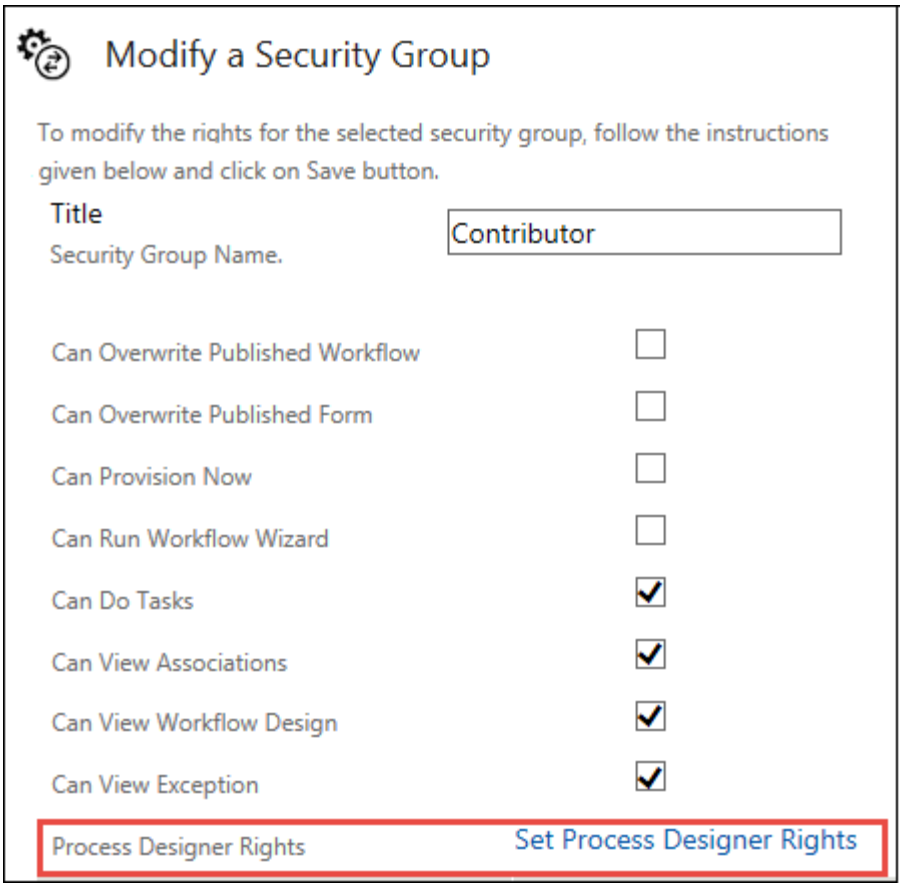
You can secure the categories to access and then you can select the required categories to include or exclude from the **Process Designer** window.

Example: The following example illustrates how to exclude the Custom Activities, BPMN Elements, Wizards and Other Actions categories for a Contributor user.

To achieve the same, perform the following steps:

1. Login as a user who belongs to the Contributor security group.
2. From the **Enterprise Console** menu, select **Workflows**.
3. To open the **Process Designer** window, select any workflow and click **Design** on the ribbon bar or right-click and select **Design Workflow** from the context menu.
4. Before the security changes are made, the **Contributor** user can access all categories.
5. As per the example, this user should not have access to **Custom Activities**, **BPMN Elements**, **Wizards** and **Other Actions** categories. To apply security for the categories in the **Process Designer** window, login to the application as an Administrator user.
6. Click **Repository Settings** > **Advanced Settings** > **Manage Security Groups**.
7. Select the Contributor security group and click **Edit**, or right-click **Contributor** and select **Edit** from the context menu.
8. Click **Set Process Designer Rights** to set action rights. The **Set Process Designer Rights** dialog box appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



9. Unselect the **Check for Inclusion** checkbox.
10. Select **Custom Activities**, **BPMN Elements**, **Wizards**, and **Other Actions**.
11. Click **Save**.
12. Click **Save** to save the **Security Group** definition and synchronize it, if required.
13. To see the effect of security provided to the categories for the **Contributor** user, login as a user who belongs to the **Contributor** security group.
14. Open the **Process Designer** window.
15. Click the **Activities** drop-down menu. Now the **Contributor** user is not able see the **Custom Activities**, **BPMN Elements**, **Wizard** and **Other Action** categories.
16. After security is applied, the **Contributor** user is able to view categories other than the selected categories.

Secure Activities of Categories

You can secure activities of categories, then expand the required category and select the required activities under it to include or exclude in the **Process Designer** window.

Example: The following example illustrates how to exclude the Task and Work activities from the Human Activities for a Contributor user.

To achieve the same, perform the following steps:

1. Login as a user who belongs to the Contributor security group.
2. From the **Enterprise Console** menu, select **Workflows**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. To open the **Process Designer** window, select any workflow and click **Design** on the ribbon bar or right-click the workflow and select **Design Workflow** from the context menu.
4. Before the security settings are changed, the **Contributor** user can access the **Task** and **Work** activities from **Human Activities**.
5. As per the example, this user should not have access to the **Task** and **Work** activities. To apply security for these activities in the **Process Designer** window, login to the application as an **Administrator** user.
6. Click **Repository Settings** > **Advanced Settings** > **Manage Security Groups**.
7. Select the **Contributor** security group and click **Edit** or right-click **Contributor** and select **Edit** from the context menu.
8. Click **Set Process Designer Rights** to set action rights.
9. Unselect the **Check for Inclusion** check box.
10. Select **Task** and **Work** activities to exclude from the **Process Designer**.
11. Click **Save**.
12. Click **Save** to save the **Security Group** definition and synchronize it, if required.
13. To see the effect of the security provided to the Activities for the Contributor user, login as a user who belongs to the **Contributor** security group.
14. Open the **Process Designer** window.
15. Select Human Activities from the **Activities** drop-down menu. Notice that the **Task** and **Work** actions are not displayed in **Human Activities**.
16. After security is applied, the **Contributor** user can view all activities other than the selected activities.

Secure Properties of Activities

You can secure the properties of activities, expand the required category and activity under that, and select the properties to include or exclude in the Process Designer window.

Example: The following example illustrates how to exclude the 'Re-Assign Allowed?' and 'Alternate Actor Allowed?' properties from the Approval actions for a Contributor user.

To achieve the same, perform the following steps:

1. Login as a user who belongs to the Contributor security group.
2. From the **Enterprise Console** menu, select **Workflows**.
3. To open the **Process Designer** window, select any workflow and select **Design** from the ribbon bar or right-click and select Design Workflow from the context menu.
4. Before security settings are applied for the properties of **Approval** activity, the Contributor user can access the **'Re-Assign Allowed?**' and **'Alternate Actor Allowed?**' properties of the Approval activity.
5. As per the example, this user should not have access to the 'Re-Assign Allowed?' and 'Alternate Actor Allowed?' properties of the Approval activity. To apply security for the properties in the Process Designer window, login to the application as an Administrator user.
6. Click **Repository Settings** > **Advanced Settings** > **Manage Security Groups**.
7. Select the **Contributor** security group and click **Edit** or right-click **Contributor** and select **Edit** from the context menu.
8. Click **Set Process Designer Rights** to set action rights.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

9. Unselect the **Check** for Inclusion check box.
10. Select the **Re-Assign Allowed?** and **Alternate Actor Allowed?** properties to exclude from the **Process Designer**. These two properties will come under the Approval activity which is under the **Human Activities** category.
11. Click **Save**.
12. Click **Save** to save the **Security Group** definition and synchronize it, if required.
13. To see the effect of the security provided on the properties for the contributor user, login as a user who belongs to the **Contributor security group**.
14. Open the **Process Designer** window.
15. Select **Human Activities** from **Activities** drop-down menu. Drag and drop an **Approval** activity if it is not there in the existing workflow and right-click the **Approval** activity and select **Activity Properties**.
16. You will notice that the **Re-Assign Allowed?** and **Alternate Actor Allowed?** properties are not displayed in the **Approval Activity Properties**.
17. After security is applied, the contributor user can view all properties other than the selected properties.

Menu Rights

This option enables the user to select the menus from the Process Designer. The selected menus and menu items will be rendered when the Process Designer is loaded.

Example: The following example illustrates how to restrict the menus for a Contributor user. Here, the Contributor user has privilege to access File, Edit, Publish, Help and Tools > Export menus.

To achieve the same, perform the following steps:

1. Login as a user who belongs to the Contributor security group.
2. From the **Enterprise Console** menu, select **Workflows**.
3. To open the **Process Designer** window, select any workflow and click **Design** on the ribbon bar or right-click and select **Design Workflow** from the context menu.
   Before the menus are controlled by applying security settings, the **Contributor** user can access all the menus.
4. Here the user is privileged for **File**, **Edit**, **Publish**, **Help** and **Tools** > **Export** menu items. To apply security for the properties in the **Process Designer** window, login to the application as an Administrator user.
5. Click **Repository Settings** > **Advanced Settings** > **Manage Security Groups**.
6. Select the **Contributor** security group and click **Edit** or right-click **Contributor** and select **Edit** from the context menu.
7. Click **Set Process Designer Rights** to set action rights.
8. Select the **Check for Inclusion** check box.
9. Expand the **Set Menu Rights** node and select the **File**, **Edit**, **Publish**, **Tools**, **Export**, and **Help** menus.
10. Click **Save**.
11. Click **Save** to save the **Security Group** definition and synchronize it, if required.
12. To see the effect of the security provided to the menus for the **Contributor** user, login as a user who belongs to the **Contributor security** group.
13. Open the **Process Designer** window.
14. The **Contributor** user is able to access **Settings**, **Wizards**, **Test Run**, **Tools** > **Import and Tools** > **Import from**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Visio.

15. After security is applied, the **Contributor** user is able to view only the selected menus and menu items.

Import Behavior for Restricted Activities and Activity Properties

If user has access rights for the activities of a workflow which has to be imported, then all the **Activities** and Activity Properties will be displayed with the respective activities for the user.

While importing the workflow, if some of the activities are restricted for the user then those activities will be replaced with **Dummy Actions**.

Example:

The following example illustrates how the user is prevented from accessing the activities and activity properties while importing a workflow.

The administrator user has access rights for all the activities and activity properties in the Process Designer. The contributor user cannot access activities like the Check-List and Information activities. These activities are under the Human Activities category.

What happens when a contributor user imports a workflow?

The 'Travel Desk' workflow is designed using the Start, Approval, Check-List and Information activities. If the administrator user imports the 'Travel Desk.xml' in the Process Designer window, this user will be able to see all the activities. When the contributor user imports the same xml, this user will not be able to see the Check-List and Information activities. These two activities will be replaced by Dummy actions.

What happens when a contributor user opens a workflow designed by an administrator user?

If the contributor user wants to open the existing 'Travel Desk' workflow which is designed by the administrator user in design mode then the contributor user is able to see the Check-List and Information activities but cannot set the Activity Properties for these activities. Since this user does not have access rights to these activities, the user will not be able to perform the actions.

To achieve the same, perform the following steps:

1. Login as a user who belongs to the Administrator security group.
2. In the Enterprise Console, click menu icon, and then click **Workflows**. The Workflows page appears.
3. On the action bar, click **New**, and then click **Workflow**. The **New Workflow** wizard appears.
4. Enter the title **Travel Desk** and a description for the workflow.
5. Click **Save & Design**. Select **Design the Workflow** now and click **Finish**.
6. The Process Designer window will be displayed. Select Tools > Import from the menubar.
7. The **Import AVEVA Work Tasks Process Definition** window appears. Locate the 'Travel Desk.xml' by clicking **Browse**.
8. Click **Import** to import the 'Travel Desk.xml' file. It will redirect you to the **Process Designer** window.
9. To restrict the activities like **Check-List** and **Information**, you need to customize the security definition of the contributor user.
10. Click **Repository Settings** > **Advanced Settings** > **Manage Security Group**.
11. Select the **Contributor** security group and click **Edit** on the ribbon bar or right-click and select **Edit** from the context menu.
12. In the **Modify Existing Security Group** window, click **Set Process Designer Rights** to restrict the activities

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Check-List and **Information**.

13. Click **Save** to save the changes.
14. Click **Save** to save the **Security Group** definition and synchronize it, if required.
15. Login as a user who belongs to the **Contributor** security group to see the effect of the security provided for the contributor user.
16. Open the **Process Designer** window. (Perform the steps 2 to 5 to open the Process Designer)
17. Import the 'Travel Desk.xml' file by clicking **Tools** > **Import**. The imported xml appears.
    Here, the Check-List and Information Activities are restricted for the Contributor user. Hence, these two activities will not be displayed in the Human Activities category list. These activities are replaced with Dummy Actions. If the user tries to edit their properties, a message saying “You do not have right for this activity” will be displayed and the Activity Properties will be empty.
18. Right-click Check-List activity and then select **Activity Properties**. The message "*You do not have right for this activity*" appears, and the *Activity Properties* will be empty.
19. Right-click Information activity and then select **Activity Properties**. The message "*You do not have right for this activity*" appears, and the *Activity Properties* will be empty.

Design Behavior for Restricted Activities and Activity Properties

While opening the existing workflow, if some of the activities are restricted for the user then those activities will be displayed but can not be able to set the Activity Properties.

Example:

The following example illustrates, how the user is restricted for the activities and activity properties while opening the existing workflow.

Here assumed that the administrator user has access rights for all the activities and activity properties in the Process Designer. Where as the contributor user has restricted to activities like Check-List and Information activities. These activities are under Human Activities category.

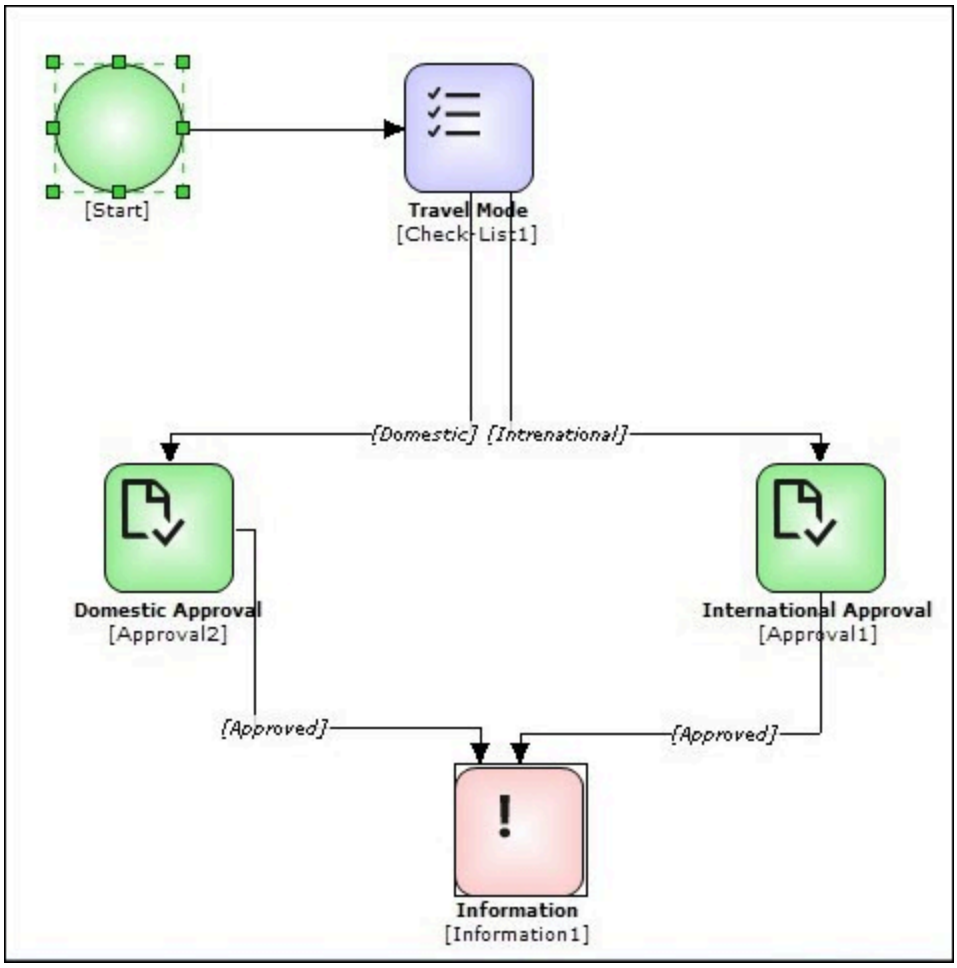
What happens when contributor user opens the Workflow designed by administrator user?

If the contributor user wants to open the existing 'Travel Desk' workflow which is designed by administrator user in design mode then the contributor user can able to see Check-List and Information activities but can not set the Activity Properties for these activities. Since this user does not have access rights to these activities and will not be able to perform the actions.

To achieve the same, perform the following steps:

1. Login as a user who belongs to the **Administrator** security group.
2. From the **Enterprise Console** menu, select **Workflows**.
3. Select the drafted **'Travel Desk'** workflow which is designed by administrator user by clicking on **Design** from the ribbon bar or **Design Workflow** from the right-click context menu.
4. The following screen is displayed. Here contributor user can able to view the **Check-List** and **Information** activities.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



5. Right-click **Check-List** activity and then select **Activity Properties**. The message "*You do not have right for this activity*" appears, and the **Activity Properties** will be empty.
6. Right-click Information activity and select Activity Properties. The message "*You do not have right for this activity*" appears, and the **Activity Properties** will be empty.

Note: You cannot re-design or open the published workflow. If you want to re-design or open the published workflow, the Can Overwrite Published Workflow right should be set as True for that security group.

See Can Overwrite Published Workflow right for more information.

Secure Process Designer Control at Workflow Level

The Process Designer Control can be secured at Repository, List, or ListItem (Workflow) level.

Example:

This example illustrates how the Repository level security will be applicable at listitem level and how to customize the security group definition at listitem level to meet the requirement of a particular workflow for a contributor user.

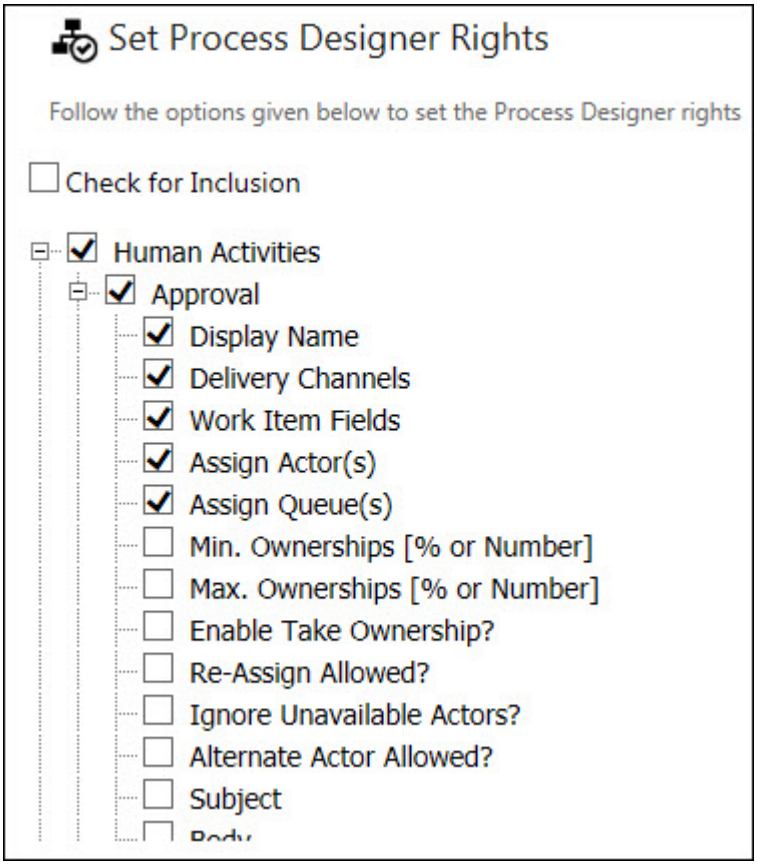
Consider three workflows 'Sales Domain', 'HR Domain' and 'Finance Domain'. Under the 'HR Domain' workflow, there are two child workflows called 'Employees Induction' and 'Leave Process' at the repository level.

The following steps explains how to secure the Process Designer control at Repository and ListItem level.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To set Process Designer rights at Repository Level

1. To customize the Contributor security group definition to set the Process Designer rights at repository level, login as a user who belongs to the Administrator security group.
2. Click **Repository Settings** > **Advance Settings** > **Manage Security Group**.
3. Select the **Contributor** security group and click **Edit** on the ribbon bar or right-click and select **Edit** from the context menu.
4. In the **Modify Existing Security Group** window, click **Set Process Designer Rights** to restrict to access the categories, activities and properties.
5. Select the **Check for Inclusion** check box.
6. Select the categories, activities and properties at the repository level.
   Here, only two categories – **Human Activities** & **Engine Activities** are privileged. The others are restricted for the contributor user.
   In the following screen a few properties are privileged and the others are restricted for the contributor user:



7. Click **Save** to save the changes.
8. Click **Save** to save the **Security Group** definition and synchronize it, if required.
9. Login as a user who belongs to the **Contributor** security group to see the effect of the security provided for the contributor user at the repository level.
10. From the **Enterprise Console** menu, select **Workflows**.
11. As there is no security is enabled at **Workflow** list and **Workflow** listitem level, the workflow list will inherit the security from the Repository level, and the Repository level security rights for the **Process Designer Control**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

12. Select '**HR Domain**' and click **Design** on the ribbon bar or select Design Workflow from the context menu. You will find the same categories, activities and properties for all the workflows, because none of them have security enabled. They inherit security from the repository level.
13. Select '**Finance Domain**' and click **Design** on the ribbon bar or select Design Workflow from the context menu. You will find the same categories, activities and properties for all the workflows, because none of them have security enabled. They inherit security from the repository level.
14. Select 'Sales Domain' and click Design on the ribbon bar or select Design Workflow from the context menu. You will find the same categories, activities and properties for all the workflows, because none of them have security enabled. They inherit security from the repository level.

Enabling ListItem Level Security for 'Sales Domain' Workflow

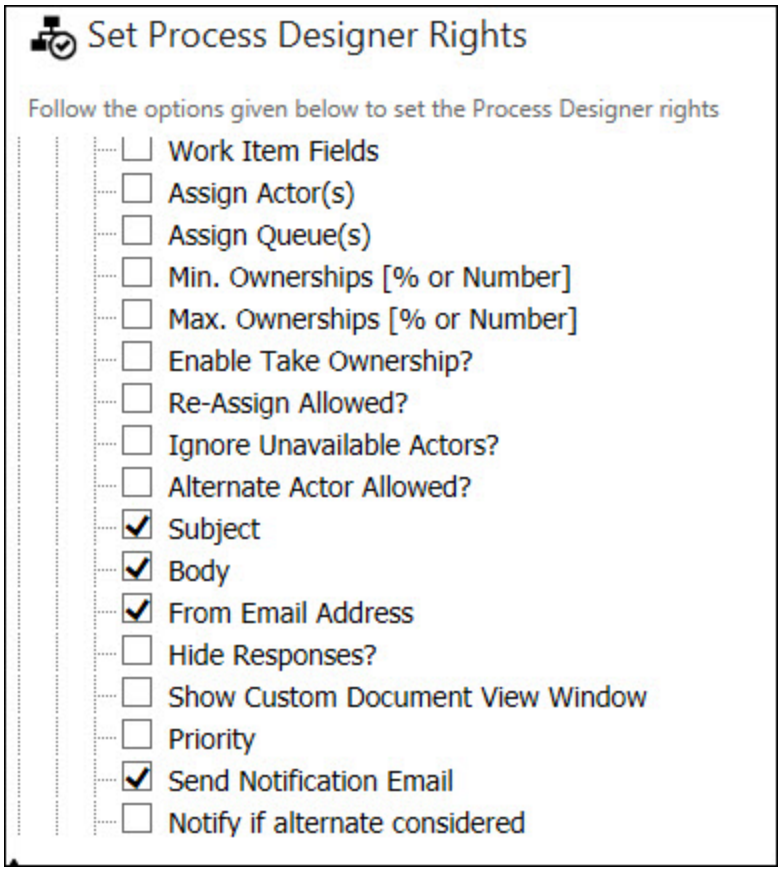
Here giving more privilege to access Integration activities (Category), Human Activities > Task (Activity), Human Activities > Approval > Subject, Body, From Email Address and Send Notification Email (Properties) along with above action rights at the 'Sales Domain' workflow listitem level.

For this, you need to customize the security definition at 'Sales Domain' workflow listitem level.

To achieve the above, perform the following steps:

1. Login as an **Administrator** user to customize the **Contributor** security group.
2. Select **Workflows** from the **Enterprise Console** menu.
3. Right-click the 'Sales Domain' workflow and then select Security Settings from the context menu.
4. Click the **Advanced Settings** tab to enable security at the list item level.
5. Click **Enable** ListItem specific Security.
6. Select **Security Group Customization**, click **Next** and again **Next**.
7. Select the **Contributor** security group and click **Edit** from the ribbon bar or select **Edit** from the context menu.
8. To open the **Process Designer Rights** window, use the **Set Process Designer Rights** link from value column.Note: Use the Set Process Designer Rights link from Follower specific value for the Rights column. It is for follower items not for the current item.
9. Select the **Check for Inclusion** check box.
10. Select the categories, activities and properties at item level as shown in the following figures.
11. Here only three categories – Human Activities, Human Activities > Approval, Check-List, Choice, Information and Task, Engine Activities and Integration are privileged the others are restricted for the contributor user.
12. In the following screen, four properties of Notification Contents of Approval activity are privileged along with the other properties. The others are restricted for the contributor user.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



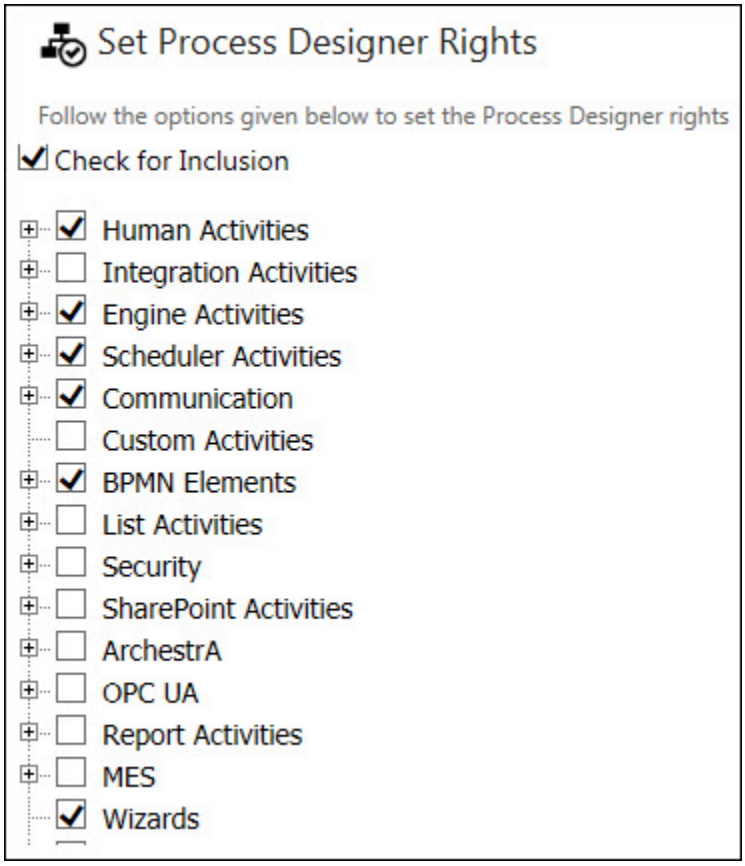
13. Click Save to save the changes.
14. Click Save to save the Security Group definition and synchronize it, if required.
15. Login as a user who belongs to the Contributor security group to see the effect of security provided for the contributor user at repository level.
16. From the Enterprise Console menu, select Workflows.
17. Select 'Sales Domain' and click Design on the ribbon bar or select Design Workflow from the context menu.

To enable ListItem Level Security for 'HR Domain' Workflow

1. Here, giving more privilege to access:
2. Scheduler Activities, Communication, BPMN Elements, and Wizard along with other categories mentioned above.
3. Human Activities > All Activities
4. Human **Activities > All Properties** along with above categories at the 'HR Domain' workflow listitem level.
5. For this, you need to customize the security definition at the 'HR Domain' workflow listitem level.
6. To achieve the above, perform the following steps:
7. Login as an **Administrator** user to customize the **Contributor** security group.
8. Select Workflows from the **Enterprise Console** menu.
9. Right-click 'HR Domain' workflow and select **Security Settings** from the context menu.
10. Click the **Advanced Settings** tab to enable the security at list item level.
11. Click **Enable ListItem** specific **Security**.

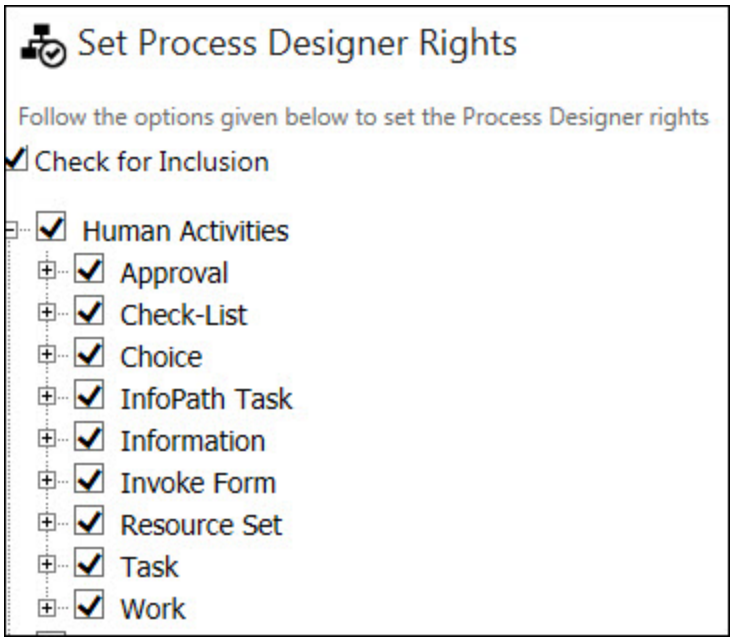
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

12. Select **Security Group Customization**, click **Next** and again **Next**.
13. Select the **Contributor** security group and click **Edit** on the ribbon bar or select **Edit** from the context menu.
14. To open the **Process Designer Rights** window, use the **Set Process Designer Rights** link from value column.
15. Select the **Check for Inclusion** check box.
16. Select the categories, activities and properties at the item level as shown in the following figures.
17. Here only six categories – **Human Activities**, **Engine Activities**, **Scheduler Activities**, **Communication**, **BPMN Elements** and **Wizards** are privileged. The others are restricted for the contributor user.

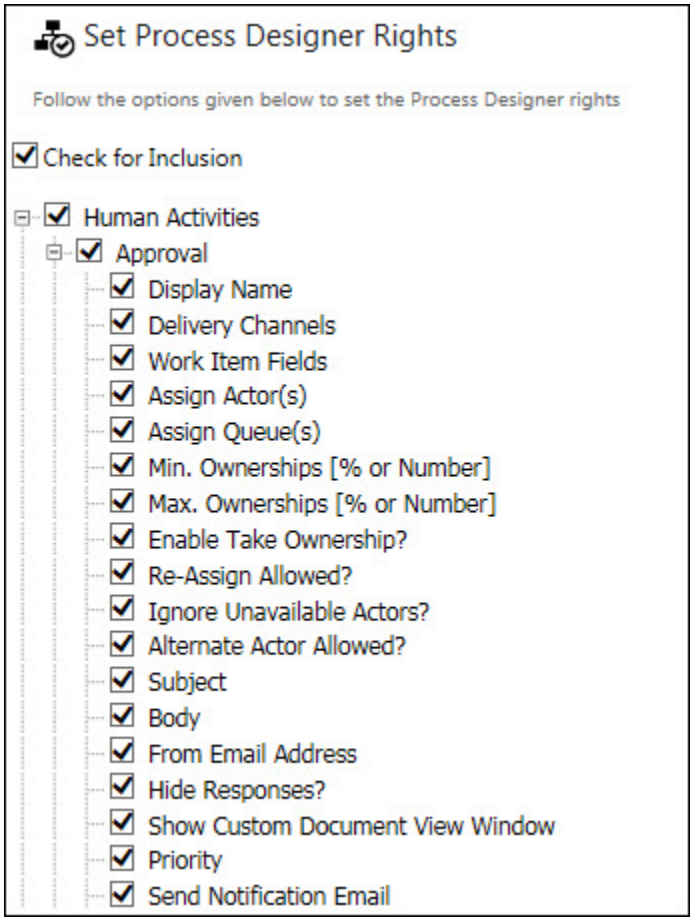


18. All activities of the Human Activities category are privileged. The others are restricted for the contributor user.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



19. All properties of the Approval Activity of Human Activities category are privileged. The others are restricted for the contributor user.



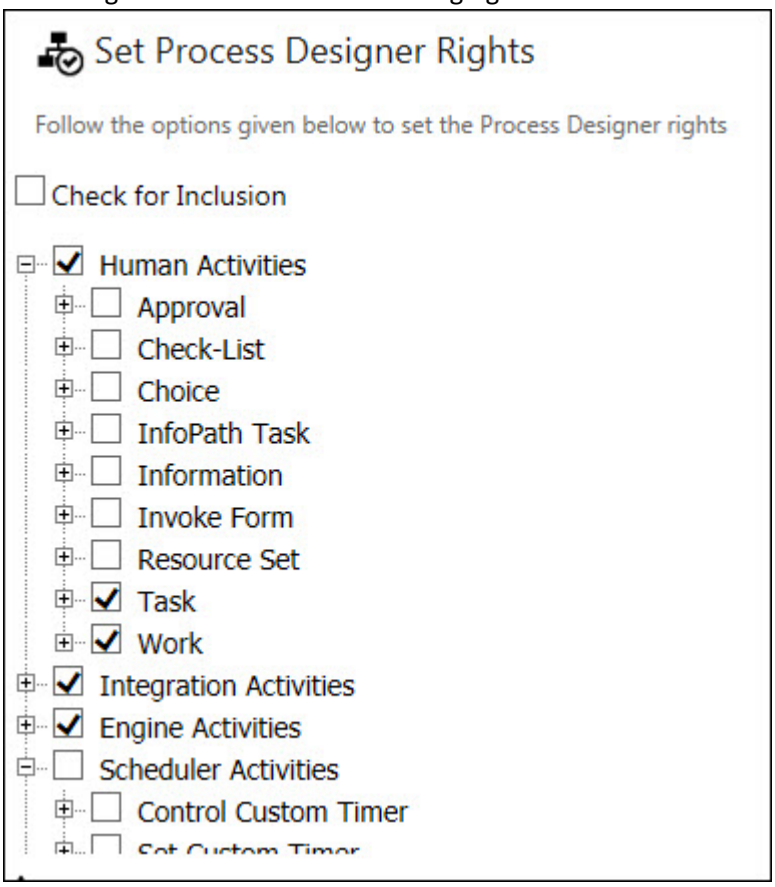
20. Click Save to save the changes.
21. Click Save to save the Security Group definition and synchronize it, if required.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

22. Login as a user who belongs to the Contributor security group to see the effect of security provided for the contributor user at repository level.
23. From the Enterprise Console menu, select Workflows.
24. Select 'HR Domain' and click Design from the ribbon bar or select Design Workflow from the context menu.
25. Here, you have set the Process Designer Rights for value, not for the follower specific value. So, if you have to navigate and view the Process Designer Control at child item then these changes will not be found there.
26. Select any child item of the 'HR Domain' workflow and click Design on the ribbon bar or select Design Workflow from the context menu. The changes are not applicable at its child item level.
27. This child item does not have any security enabled. It will inherit from its immediate Dictator. In this case, the parent item is 'HR Domain'.

To enable Security at Child Item Level for 'HR Domain' Workflow

1. To enable security at child item level, follow the steps given below:
2. Perform the steps 1 to 7 (See Enabling ListItem Level Security for 'HR Domain' Workflow steps).
3. To open the Process Designer Rights window, use the Set Process Designer Rights link from Follower specific value column.
4. Set the rights as shown in the following figure:



5. Click Save to save the changes.
6. Click Save to save the Security Group definition and synchronize it, if required.
7. Login as a user who belongs to the Contributor security group to see the effect of the security provided for

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the contributor user at the repository level.

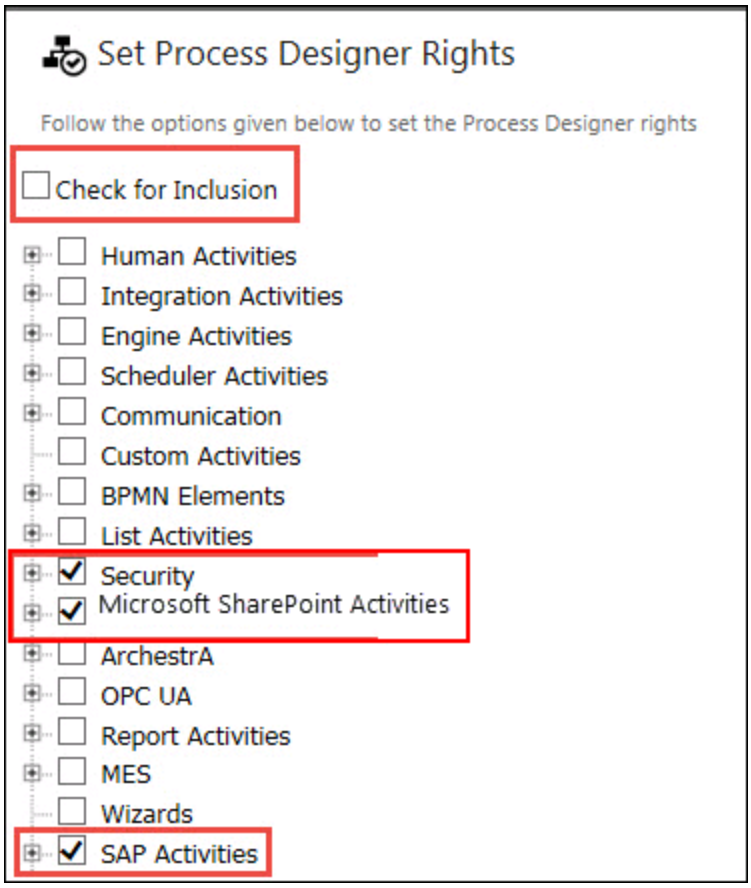
8. From the Enterprise Console menu, select Workflows.
9. Double-click 'HR Domain' and select any child item (Employee Induction or Leave Process).
10. Select Design from the ribbon bar or select Design Workflow from the context menu to see the effect.
11. Here, both the child workflows (Employees Induction and Leave Process) do not have the security enabled. Hence, they will inherit the security set by their immediate dictator. In this case, their parent workflow (HR Domain).

To enable Child Item Level Security for 'HR Domain' Workflow

The following example demonstrates that the security has been enabled at one child item i.e 'Employees Induction' and shows the effect for both the child items.

1. Go to the Employees Induction workflow and enable the listitem level security. To know more about how to enable security at the listitem level, click “Enabling Security at ListItem Level” and customize the Process Designer rights for the security group.
2. To enable security for child items, follow the steps given below:
3. Login as an Administrator user to customize the Contributor security group.
4. Select Workflows from the Enterprise Console menu.
5. Double-click the 'HR Domain' workflow and then select 'Employees Induction' workflow.
6. Right-click and select Security Settings from the context menu.
7. Click the Advanced Settings tab to enable the security at list item level.
8. Click Enable ListItem specific Security.
9. Select Security Group Customization, click Next and again Next.
10. Select the Contributor security group and click Edit from the ribbon bar or select Edit from the context menu.
11. To open the Process Designer Rights window, use the Set Process Designer Rights link from the Follower specific value column.
12. Select the Check for Inclusion check box.
13. Select the categories, activities and properties at item level.
14. The screen shows that the categories (Security and SAP Activities) are excluded/restricted.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



15. Click Save to save the changes.
16. Click Save to save the Security Group definition and synchronize it, if required.
17. Login as a user who belongs to the Contributor security group to see the effect of security provided for the contributor user at the repository level.
18. From the Enterprise Console pull-down menu, select Workflows.
19. Double-click 'HR Domain' and select any child item i.e. Employee Induction.
20. Select Design from the ribbon bar or select Design Workflow from the context menu to see the effect.

    The Employees Induction Process Designer will open with all the categories except the restricted ones.
21. For the 'Employees Induction' workflow, the administrator user has enabled and customized the security as mentioned above. For another child workflow called 'Leave Process', the administrator user has not enabled and customized the security. So, it will inherit from its immediate dictator. In this case, the 'HR Domain' workflow.
22. Right-click the 'Leave Process' workflow and select Design workflow and see the effect.

Can Overwrite Published Workflow Right

Can Overwrite Published Workflow right is introduced to prevent any changes to a published workflow that can intervene with any running instances of a workflow. This right can be assigned to the user's security group to allow the user to make modifications to a published workflow. By default, this right is restricted to all the security groups including the Administrator. This is because when this right is assigned, the user can make changes to a published workflow and thereby intervene with any running instances of the workflow. This will have an effect

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

on the new instances of the same workflow in the production environment. Hence, the user is recommended to create a minor version of the workflow, make the necessary modifications, and then publish it.

To be able to overwrite a published workflow, a user with **Manage Security** rights can enable this right either from Repository Settings or from the workflow specific Security Settings.

Behavior of a list item when Can Overwrite Published Workflow right is not assigned

1. Right-click the published workflow which needs to be edited. Since this workflow is published, the **Design Workflow** option is not available in the context menu.
2. If the user selects the **Design Workflow** option from the ribbon bar, the following message appears:
   You do not have rights to edit a published Workflow

To design this workflow, the **Can Overwrite Published Workflow** right needs to be assigned from the Repository Settings.

To assign 'Can Overwrite Published Workflow' right through Repository Settings

1. Point to **admin**, and then click **Repository Settings**. The Repository Settings page appears.
2. Click the **Advanced Settings** tab.
3. Click **Manage Security Groups** option. The Security Group screen displays all the security groups.
4. Select the required security group and click the **Edit** option on the ribbon bar, or right-click the group and select the **Edit** option from the context menu. The **Modify a Security Group** screen appears.
5. In the **Modify a Security Group** screen, scroll down to view the "**Can Overwrite Published Workflow**" under the **Workflow List Rights**.
6. Check the '**Can Overwrite Published Workflow**' option.
7. Click **Save** to assign the right to the security group. The Synchronization screen appears. Users can synchronize the changes made to the security group to the below levels as well. For more information, go to the Synchronization topic.
8. Check the **Yes** radio button to synchronize changes for the lists displayed.
9. To avoid synchronizing the changes, do not select the **Yes** option.
10. Click **Submit**. The confirmation message about successfully modifying the security group appears.
11. Click **OK**.

    Now, the user can edit any published workflows either through the context menu or the ribbon bar.

Behavior of the list item after the 'Can overwrite Published Workflow' right is assigned

Go to the Workflow list screen from the Enterprise Console menu and right-click the workflow to see the Design Workflow option enabled in the context menu.
If the user selects this option, the process designer with the workflow design opens and the user can edit the design of the workflow.

Note: It is recommended to set this right to false in the production environment to avoid any modifications to the design of the workflow.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Advanced Options

To assign 'Can Overwrite Published Workflow' right at the List item level

Prerequisite: The security group customization should be enabled for this list item.

1. Right-click a published workflow and select **Security Settings** from the context menu.
2. Click the **Advanced Settings** tab in the settings screen.
3. Click **Customize Security Group for this List Item** option. The **Security Group** screen appears.
4. Select the required security group and click the **Edit** option on the ribbon bar, or right-click the group and select the **Edit** option form the context menu. The **Modify a Security Group** screen appears.
5. In this screen, scroll down to the **Workflow List Rights** and check both the boxes against the **Can Overwrite Published Workflow** option. The option under **Value for the Right** column should be checked for the current workflow. The option under the **Follower-specific value for the Right** should be checked for the child workflows.
6. Click **Save** to assign the right to the user. The next few steps are similar to the ones given under the Repository Level section.
   Once the right is assigned, the user can edit any published workflows either through the context menu or the ribbon bar.

Advanced Security Options

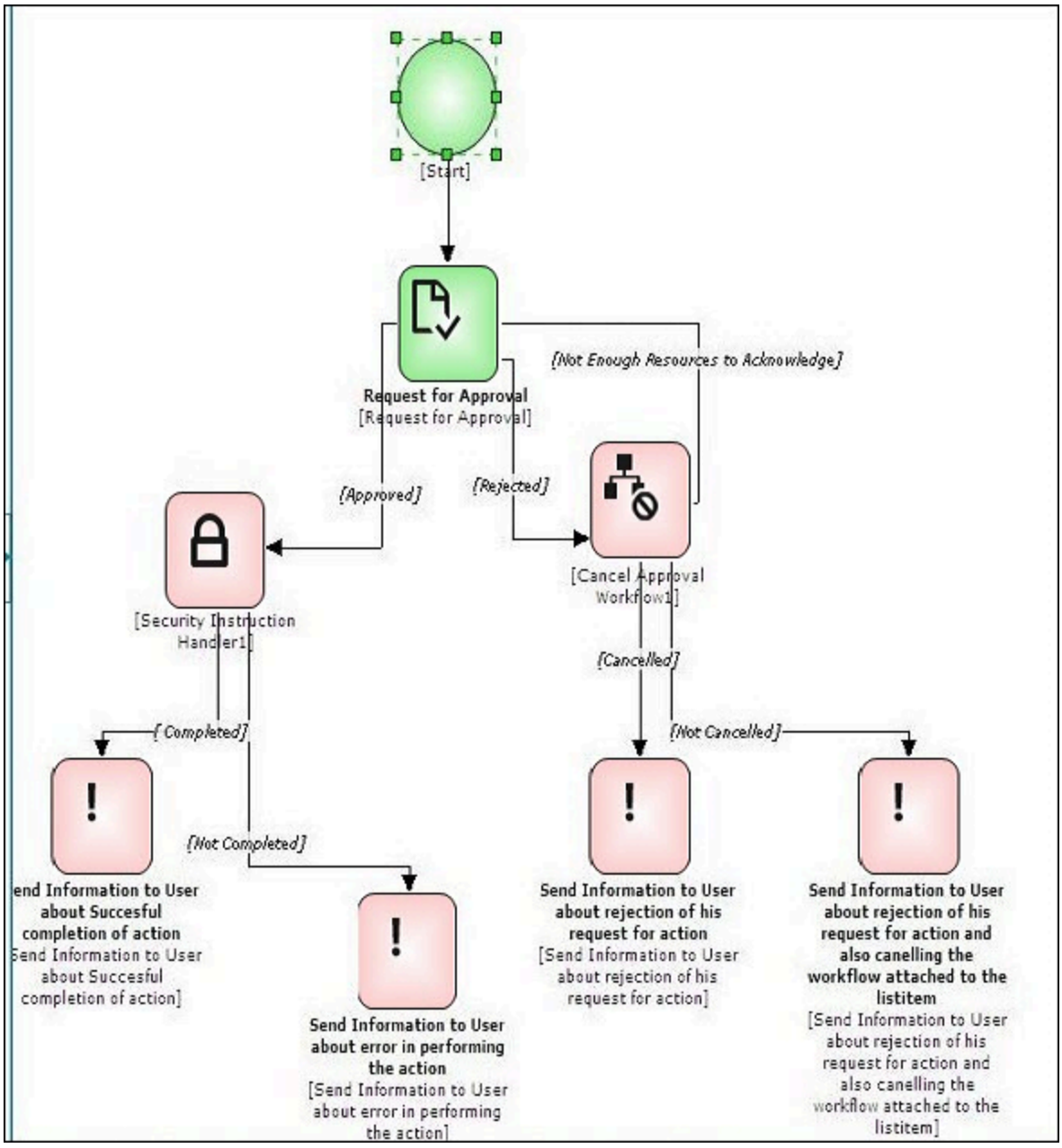
Set up security approval workflow in AVEVA Work Tasks.

Set up Security Approval Workflow

To setup security approval workflow

1. Login as an administrator user.
2. In the **Enterprise Console**, click menu icon, and then click **Workflows**. The **Workflows** page appears.
3. On the action bar, click **New**, and then click **Workflow**. The **New Workflow** wizard appears.
4. In the **Title** box, type a name for the workflow, and in the **Description** box, type a description for the workflow. For example, create **Security Workflow** workflow.
5. Click **Save & Continue**. The workflow is created, and the **Workflow Options** wizard appears.
6. Select **Design** the Workflow now, which is the default option, and then click **Finish**. The **Process Designer** appears.
7. Click **Tools** > **Import**.
8. Click **Browse** and then select the security approval workflow from *..\AVEVA\AVEVA Work Tasks\security_approval_workflow.xml*, as shown in the following figure:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



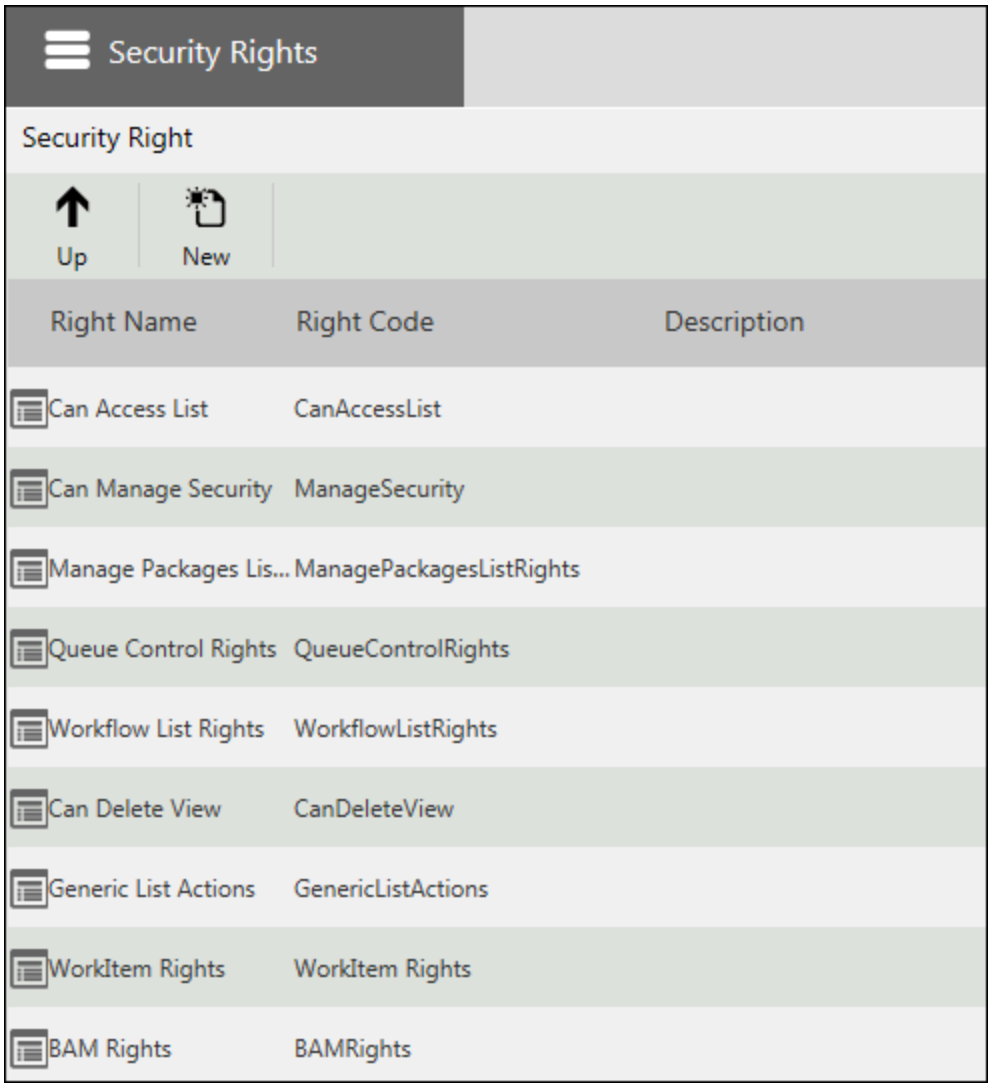
Manage Security Rights

This option allows user to Add, Modify, View and Delete Security Rights. Security Rights which are part of Security Groups are used to set for Lists and List Items. Users will have access rights as defined in the Security Group definition, that they belong to. The page lists all the standard rights that are provided.

To open Manage Security Rights

1. Login to the **Enterprise Console**.
2. Point to admin, and then click **Repository Settings**. The **Repository Settings** page appears.
3. Click the **Advance Settings** tab.
4. Click **Manage Security Rights**. The **Security Right** window displays all the security rights.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Note: If a user is assigned with additional privileges while the user is logged on to the Enterprise Console site, these privileges will not have an immediate effect. The user must log out of the Enterprise Console site and wait for five minutes for the changes to take effect.

See Also:

- Creating a Security Right
- Modifying a Security Right
- Deleting a Security Right
- Viewing a Security Right

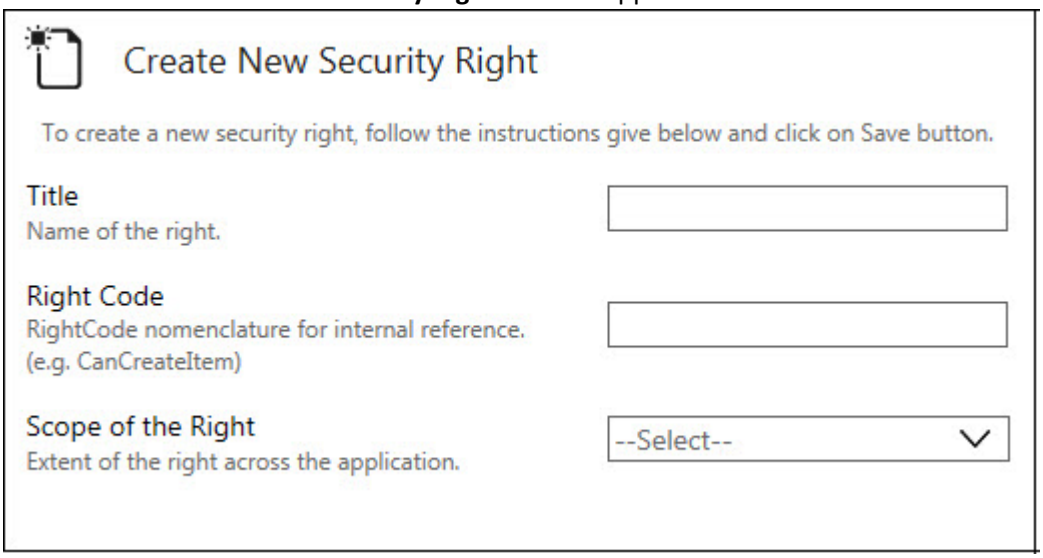
Create a Security Right

The new option can be used to create and add a new security right that can be later used within a security group.

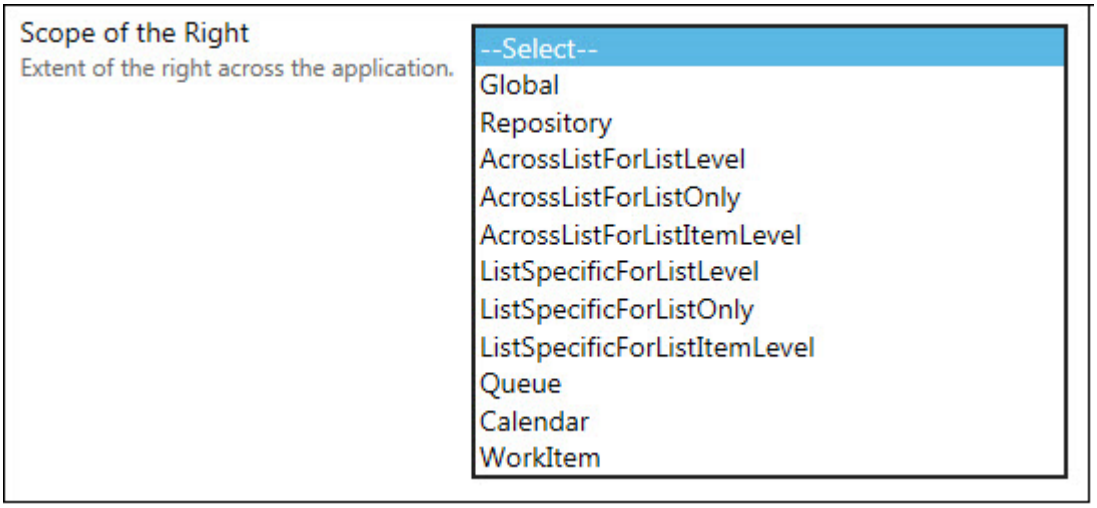
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To create a new security right, follow the steps given below

1. Log on to the **Enterprise Console**.
2. Point to admin, and then click **Repository Settings**. The **Repository Settings** page appears.
3. Click the **Advance Settings** tab.
4. Click **Manage Security Rights**. The **Security Right** window displays all the security rights.
5. Click **New**. The **Create New Security Right** window appears.

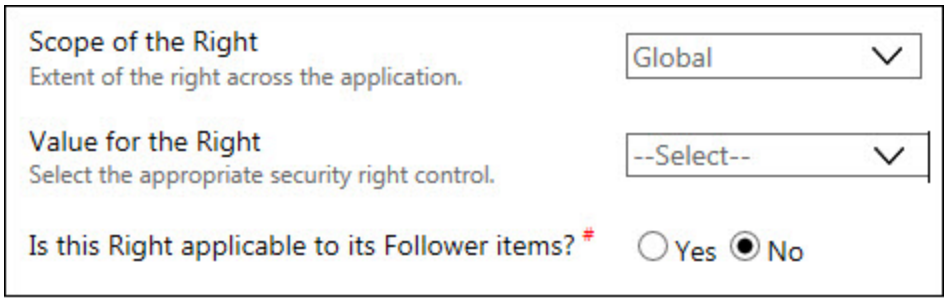


6. In the **Title** text box, enter a name for the right.
7. In the **Right Code** text box, enter a code name for the right.
8. In the **Scope** of the **Right** drop-down list, select the scope for the right created. Scope for the right can be **Global**, **Repository**, **List Specific**, **ListItem Specific**. Depending on the level selected, different fields will get displayed.

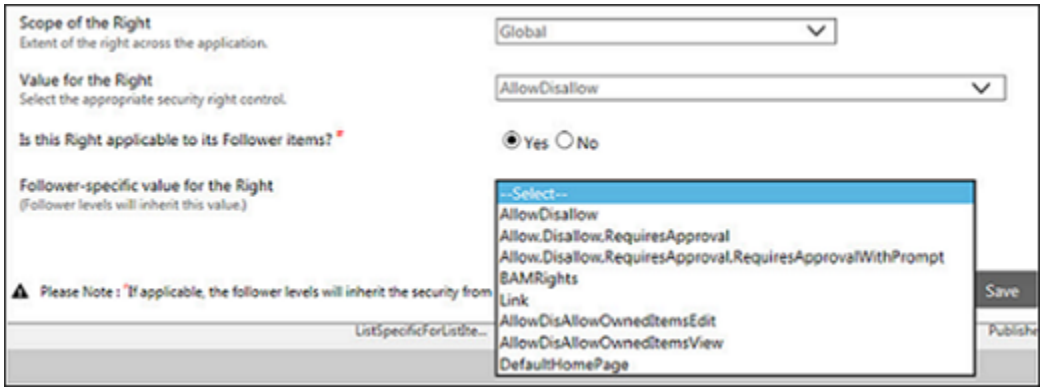


9. The following options appear if you select **Global**, **AcrossListForListLevel**, **AcrossListForListOnly**, or **AcrossListForListItemLevel**:

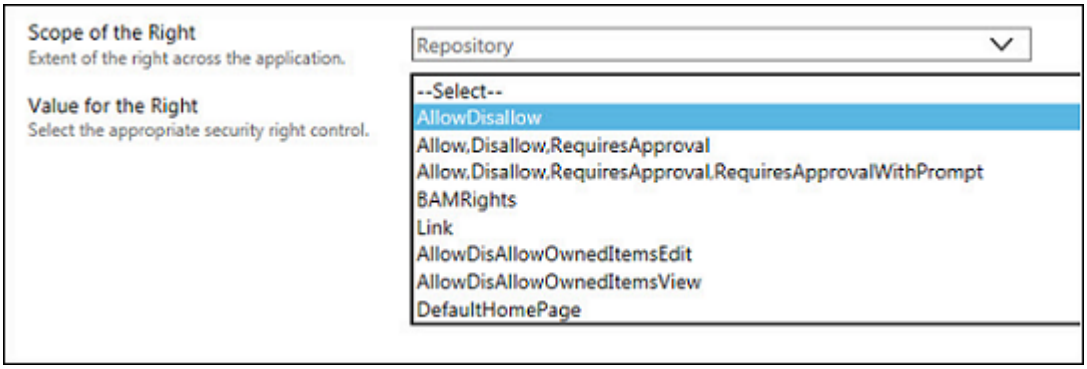
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



10. If you select the **Yes** radio button, the options shown below will get displayed. This allows you to set the values for the right created.



11. The option shown below will get displayed, if you select the scope for the right as **ListSpecificForListLevel**, **ListSpecificForListOnly** and **ListSpecificForListItemLevel**. This allows the user to select the **Lists** for which the new right can be applied. Value for the new right can also be set.
12. The following options will get displayed, if you select the scope for the right as **Repository**, **Queue**, **Calendar** or **Work Item**. This page opens the list of values that can be set for the right.



13. After selecting the relevant scope for the new security right and setting the values for it, click **Save** to finish and to save the new security right.

Modify a Security Right

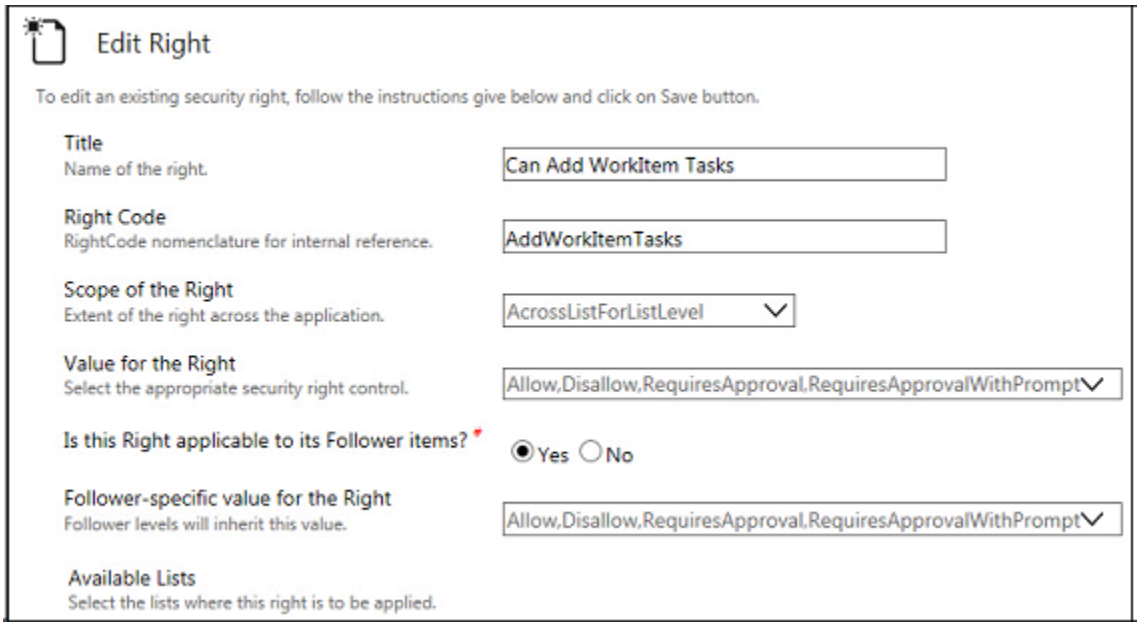
This option allows us to modify the scope, values and follower values for an existing security right.

To edit the security right

1. Log on to the **Enterprise Console**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Point to admin, and then click **Repository Settings**. The Repository Settings page appears.
3. Click the **Advance Settings** tab.
4. Click **Manage Security Rights**. The Security Right window displays all the security rights.
5. Select the security group to be modified from the list. For example, select the 'Can Add Work Item Tasks' right.
6. Click **Edit**. The following page is displayed:



7. Edit the required fields.
8. Click **Save** to update the changes.

Note: The user cannot modify the settings of Standard Rights. Only the settings of the user created security rights can be modified.

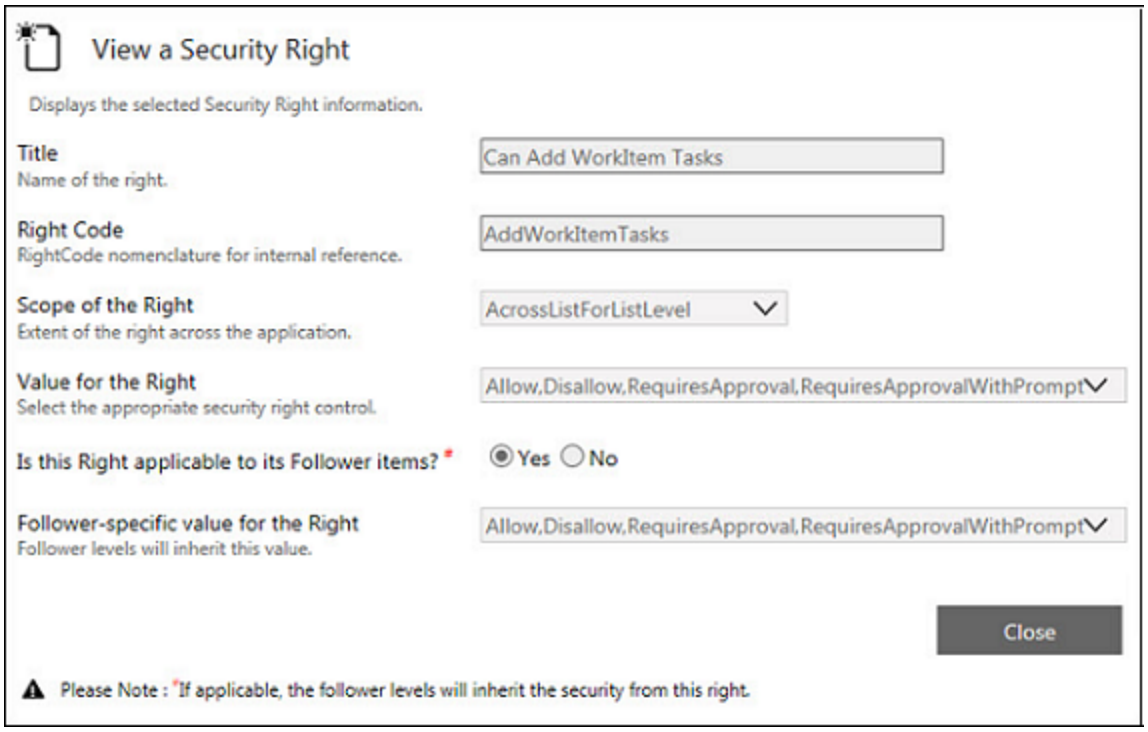
View Security Right

View option can be used to view the security right for the selected security right.

To view the security right

1. Log on to the **Enterprise Console**.
2. Point to admin, and then click **Repository Settings**. The **Repository Settings** page appears.
3. Click the **Advance Settings** tab.
4. Click **Manage Security Rights**. The **Security Right** window displays all the security rights.
5. Select the security group to be viewed.
6. Click **View**. The following window is displayed:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



7. Click **Close**.

Delete a Security Right

Delete option can be used to delete the selected security right.

To delete the security right

1. Log on to the **Enterprise Console**.
2. Point to admin, and then click Repository Settings. The Repository Settings page appears.
3. Click the **Advance Settings** tab.
4. Click **Manage Security Rights**. The Security Right window displays all the security rights.
5. Select the security group to be deleted.

   Note: It is not possible to delete the standard security rights. User defined rights can be deleted.
6. Click **Delete**.
7. Click **Yes** to delete the selected security right.

   Note: The user cannot delete the Standard Security Rights. Only the user defined security rights can be deleted.

Scope of Rights

Scope for Security Rights

Some rights will be applicable only at the repository level which are of scope:

- **Repository**- If it is a Repository scope, the rights will appear at the Repository level. For example: Calendar

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

rights, Work items list rights, Queue rights.

- **Queue** - If it is a Queue scope, the rights can be applied only at Queue.For example: Can Add New Queue and Can View Queue Control
- **Calendar** - If it is a Calendar scope, the rights can be applied only at Calendar.For example: Can Add Calendar, Can Delete Calendar, and Can View All Calendars.
- **Work Item** - If it is a Work Item scope, the rights can be applied only at Work Item.For example: Can Forward Work Item, Can Set Alternate Actor, and Can View Work Item Control.

Some rights will be applicable at only at the list level which are of scope:

- **AcrossListForListOnly** - If it is a **AcrossListForListOnly** scope, the rights can be applied only at the List level not at the ListItem level across all lists.
- **ListSpecificForListOnly** - If it is a **ListSpecificForListOnly** scope, the rights can be applied at only at the List level not at the ListItem level for a specific list.

Some rights are applicable only at the listitem level which are of scope:

- **AcrossListForListItemLevel** - If it is a **AcrossListForListItemLevel** scope, the rights can be applied only at the ListItem level across all lists. For example: Can Publish an Item
- **ListSpecificForListItemLevel** - If it is a **ListSpecificForListItemLevel** scope, the rights can be applied only at the ListItem level for a specific list. For example: Queue, Calendar and Work Item

Some rights are applicable both at the list level as well as the listitem level which are of scope:

- **AcrossListForListLevel** - If it is a **AcrossListForListLevel** scope, the rights can be applied at the List level and ListItem level across all lists. For example: The Can Add Item right can be applied at the List level and ListItem level.
- **ListSpecificForListLevel** - If it is a **ListSpecificForListLevel** scope, the rights can be applied at the List level and ListItem level for a specific list. For example: The Can Design Workflow right can be applied at the List level and ListItem level for list Workflow.

Some rights are applicable at all the levels which are of scope:

Global – If it is a Global scope, the rights will appear at all levels in the site. For example: Manage Security

Create User Defined Operations and Rights

AVEVA Work Tasks has set of custom operations and set of custom rights on Repository, List and ListItem level. The custom operations like add, edit, delete and security etc. AVEVA Work Tasks has provided an option to user to add own custom operation(menu item) in context menu and also can define own custom rights based on the requirements and can use these rights and operations in the application.

For example, user wants to create a 'order' operations in context menu in the enterprise console application to see order details by clicking on the 'order' operator from context menu.

Adding a New Custom Operations and Custom Rights

The following example demonstrates creating a new custom operation in a context menu and as well as adding new rights for the created custom operation to control this action.

Example: The user wants to add a new custom operation called 'View Orders' in the context menu of the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

'Product Categories' List.

The following steps explain the process to be followed:

1. Create a aspx page called 'Products' as shown below.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Products.aspx.cs"
Inherits="Orders" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html >
<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<table width="100%" cellpadding="0" cellspacing="5" border="1">
<tr>
<td> Sl.No </td>
<td> Product </td>
</tr>
<tr>
<td>1</td>
<td>Books</td>
</tr>
<tr>
<td>2</td>
<td>Computer and Peripherals</td>
</tr>
<tr>
<td>3</td>
<td>Consumer Electronics</td>
</tr>
<tr>
<td>4</td>
<td>Kitchen and Home Appliance</td>
</tr>
<tr>
<td>5</td>
<td>Memory cards, pen drives and Harddisk</td>
</tr>
<tr>
<td>6</td>
<td>Mobiles and Accessories</td>
</tr>
<tr>
<td>7</td>
<td>Musical Instruments</td>
</tr>
<tr>
<td>8</td>
<td>Tools and Hardware</td>
</tr>
<tr>
<td>9</td>
<td>Video and Computer Games</td>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</tr>
<tr>
<td>10</td>
<td>Services</td>
</tr>
</table>
</div>
</form>
</body>
</html>
Note: Place 'Products.aspx', and 'Products.aspx.cs' in the  [AVEVA Work Tasks
Installed Path]\AVEVA\AVEVA Work Tasks\Web\BPMUITemplates\Default\Repository\Site
folder.
```

2. Create one more aspx page called 'Custom Operations' as shown below.

   Implementation part is outlined below:

   a. First you have to create the list definition object by passing the Application Name (Repository name) and List Name to get the actions which are associated with the list definition. In this example, Application name set as 'ListSecurityRepo' and List name as 'NavigationDetails'.
   b. After creating the list definition object, you will get the existing actions associated with that list.
   c. Store all existing actions to new object.
   d. Create a new action and set the new action properties like name, type, pagepath, iconpath, DisplayValidRightValues, and ActionAllowValidRightValues.

   Name - Mention a name for the context menu. Here name is mentioned as 'View Products'.

   Type - Mention a type. Here type is mentioned as 'MenuItem'.

   PagePath - Mention a path of the aspx page to navigate by clicking on 'View Products' custom operation.

   IconPath - Mention a image path. Here icon path mentioned as 'Edit'. Which is a class under *[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\BPMUITemplates\Default\Repository\ListControl\Scripts\JsdoMenu\themes\office_xp*. Here edit image is used for newly created context menu.

   DisplayValidRightValues - Mention right code. Here Right code is mentioned as 'ActionOrders' and set as true value for Value for the Rights.

   ActionAllowValidRightValues - Mention right code. Here Right code is mentioned as 'ActionOrders' and set as true value for Follower specific value for the rights.

   a. Add new action to the object which is created.
   b. Finally call the save method on list definition object.
   c. Right-click on 'Custom Operation' aspx page and then click on View in Browser. You will get a message like 'New action added successfully'.

   See the Developer's Guide for more implementation about creating a new context menu.

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
```

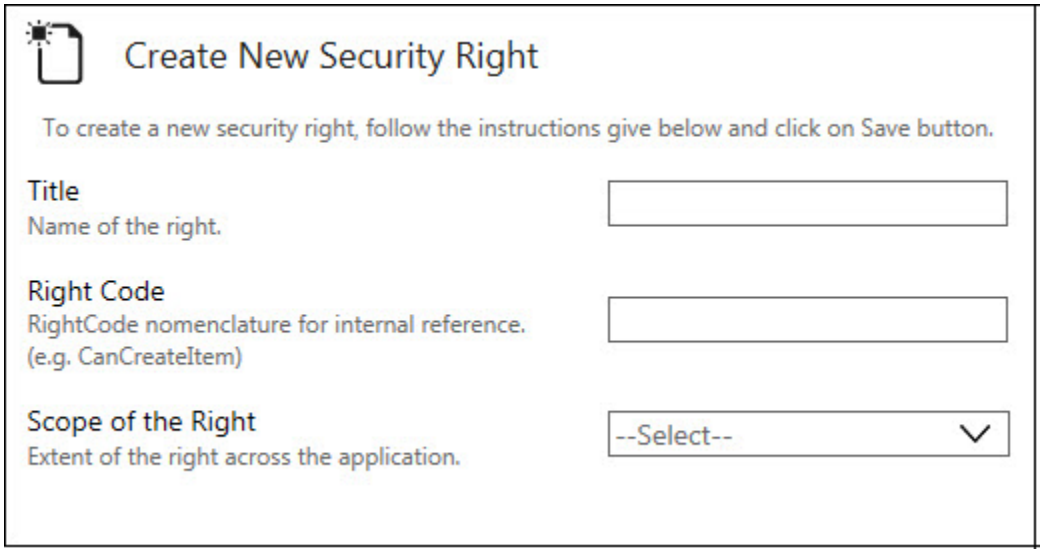
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
using Skelta.Repository.List;
using Skelta.Repository.List.Action;
using Skelta.Core;
using System.Collections.Generic;

public partial class CustomOperations : System.Web.UI.Page
{
  string ApplicationName = "ListSecurityRepo";
string ListName = "NavigationDetails";
ListDefinition _ListDef;
ListActions UpdatedActions;
protected void Page_Load(object sender, EventArgs e)
{

}

protected void btnCreate_Click(object sender, EventArgs e)
{

try
{
// Creating ListDefinition object
_ListDef = new ListDefinition(new ApplicationObject(ApplicationName), ListName);
// Getting Existing actions in List<ListAction> collection
List<ListAction> _ExistingActions = _ListDef.Operations.Actions;
//Declaring ListActions Object
UpdatedActions = new ListActions();
// Looping through Existing actions and add each action ListActions object
foreach (ListAction action in _ExistingActions)
{
UpdatedActions.Actions.Add(action);
}

// Creating New Action

ListASPXUIAction NewAction = new ListASPXUIAction();
// Setting up new action properties
NewAction.Id = "_sys_Orders";
  NewAction.Name = " View Products";
NewAction.Type = ListActionType.MenuItem;
NewAction.PagePath = "Products.aspx";
NewAction.IconPath = "Edit";
NewAction.DisplayValidRightValues = "ActionOrders<True>";
NewAction.ActionAllowValidRightValues = "ActionOrders<True>";
NewAction.ApprovalRequiredValidRightValues = "";
NewAction.ApprovalRequiredWithPromptValidRightValues = "";

// Adding new action
UpdatedActions.Actions.Add(NewAction);
// Setting opeartions of ListDefinition object
_ListDef.SetOperations(UpdatedActions);
// Finally calling Save method of ListDefinition object
_ListDef.Save();
Response.Write("New action added successfully");
}
catch (Exception ex)
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
{
Response.Write(ex.Message.ToString());
}
}
}
```

3. Login as an administrator user.
4. Click **Repository Settings (top-right corner)** > **Advanced Settings** > **Manage Navigation List**.
5. Right-click on any **Navigation** link and check the newly added context menu. 'View Products' is not added in the context menu.
6. Here, you need to define security settings at this level. Before defining security settings at this level, you need to create security rights for the created action to control.
7. To create a security right for this item, click **Repository Settings (top-right corner)** > **Advanced Settings** > **Manage Security Rights** > **New** . The following screen is displayed:



8. In the **Title** text box, enter right name as **Can View Products**.
9. In the **Rights Code** text box, enter right code as **Action Orders**.
10. From Scope for the Rights drop-down list, select scope as 'ListSpecificforListItemLevel'. See List Level Security Settings for more information on scope.
11. After selecting the relevant scope for the new security right and setting the values for the right. This allows the you to select the **Lists** to which the new right can be applied for and you can also set the value for the new right.
12. From **Value** for the Right drop-down list, select **AllowDisallow** from the list.
13. Select **Yes** for **Is this Right applicable to the Follower Rights?**
14. Select **AllowDisallow** from the **Follower** specific value for the Right drop-down list.
15. Select list from the available list. Here you have to select **Navigation List**.
16. Click **Save**. The right has been created.
17. Click **Repository Settings (top-right corner)** > **Advanced Settings** > **Manage Security Group**.
18. You have to customize administrator security group because only administrator user has rights to update the security rights. Select Administrator security group and then click **Edit** on the ribbon bar or else right-click and then click **Edit** from the context menu. The **Modify a Security Group** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

19. Select **Can View Products** from the **Follower** specific value for the **Rights** column and then click **Save**.
20. Click **Cancel** from the **Synchronization** window.
21. Click **OK**.
22. Click **Repository Settings (top-right corner)** > **Advanced Settings** > **Manage Navigation List**.
23. Right-click any navigation list and check the newly created menu is displayed.
24. Click **View Products**, this will redirect you to the **Product Details** page.
25. Right-click and then select **Back**.
26. For some of the list menus, the newly created context menu is not displayed. For example, right-click the **Inbox** menu. Here, the newly created menu item is not displayed. This is because security settings have been defined at this list item level.
27. Right-click the **Inbox** menu list and click **Security Settings** to enable the created custom operation.
28. Click **Advanced Settings** > **Enable List specific Security**.
29. Click **Customize Security Group** for this List Item. The **Security Group** screen is displayed.
30. You have to customize the administrator security group because only the administrator user has rights to update the security rights. Select the **Administrator** security group and then click **Edit** on the ribbon bar or else right-click and then click **Edit** from the context menu. The **Modify a Security Group** page appears.
31. Select **Can View Products** for both **Value** for the **Rights** and the Follower specific value for the Rights column and click Save.
32. Click **Cancel** from the **Synchronization** window.
33. Click **OK**.
34. Click **Repository Settings (top-right corner)** > **Advanced Settings** > **Manage Navigation List**. The **Navigation Details** page appears.
35. Now you are able to see the **View Products** context menu for the Inbox menu list.

Enable and Disable Security at List/List item level

A List is a storage place where a user can store groups of information that possess some common attributes. It has the provision to add new information, edit and delete existing information, and publish/unpublish the information.

In a real-time enterprise application, the user stores expense list, travel claims, leave details, employee details, product details, order details in a list structure. In a typical organization, lists store employee details, product details, and expense details. These lists contain some common attributes. For example, the product list stores information including the product name, price, manufacture date, and expiry date. The order list stores information like the order ID, order date, order item pertaining to the order.

ListItem

Each row or record of information inside a specific list can each be treated as a ListItem. These rows or records possess some common attributes. For example, each item inside a product list is called a ListItem of the product list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

List Operations

The list operations are add, edit, view, and delete. For example, add a new product, modify information about an existing product, delete a product, and publish or unpublish information about a product.

List

To provide the operations mentioned previously, AVEVA Work Tasks has the List infrastructure where you can create a list and store list information. You can perform custom operations like add, edit, view and delete for that list.

AVEVA Work Tasks has implemented the list infrastructure to manage and store lists like Process list, Repository list, Event Provider list, Workflows list, Navigation list, User list, SOA Assembly Folder list, Database connection list, File Watcher list, Lookup list and user defined custom list.

Workflow List - Workflows are maintained as a list. Each workflow has a name, description, owner, version information, item status information, workflow status information and last modification information. The user can perform operations like creating a new workflow, editing the workflow using the Process Designer, viewing workflow, deleting workflow, publishing workflow, unpublishing workflow, saving workflow with different versions, disabling or enabling workflow and associating workflow with events like File watcher.

Users List - Each user has information like name, email information, department information, role and owner. The User Management maintains this information as a list and allows you to add a new user, view user information, edit user information and delete a user.

Menu List - The menus are also maintained as a list to store the information in a list infrastructure.

AVEVA Work Tasks has a new feature called **Forms Designer** and **List Designer**. The **List Designer** is used to create enterprise specific list, product list, orders list, customer details list, and so on. Each list can be used to render the forms specific to that list to create items inside the list.

In a enterprise specific list like **Product List**, which has fields such as Product ID, Manufacturing Date, Warranty Period, Price. Users can add product details, delete/update the existing details.

Security

AVEVA Work Tasks also has the Security feature in the list infrastructure. Using the Security feature, the list as well as each item inside the list can be secured to ensure, certain actions like add, edit, delete can be performed only by authorized users (users who have the rights to perform these actions).

Why List Specific Security?

The List specific security is required when you need to restrict the access to the list and the operations that can be performed on that list. The List Level security provides features that allow you to maintain and manage user privileges across the list.

The **List** level security allows the user to secure access to the various lists available in a Repository.

AVEVA Work Tasks defines a set of rights that can be set for users at the List Level. List level rights can be grouped as follows:

- General Rights - These rights are common across lists.
- List Specific Rights - Rights that are specific to lists. [Workflow, User list, Inbox – Work items list]
- List Level - General Rights: A set of rights that are applicable across all lists
- The user can control the following actions across all lists:
- Can add item - Whether a user can create a new item in the list or not

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Can delete item - Whether a user can delete an item from the list or not
- Can edit item - Whether a user can edit an item in the list or not
- Can publish item - Whether a user can publish an item in the list or not
- Can unpublish item - Whether a user can unpublish an item in the list or not
- Can enable/disable - Whether a user can enable/disable an item in the list or not

  Note: This option is available only for the below lists:
  - Workflow
  - Forms
  - SOA Assembly
  - SOA Folder
  - Event Association
  - SSRS
- Package Item - Whether a user can add the artifact in the list to a package template or not
- Can manage security - Whether a user can manage the security of an item in the list or not
- Can see all versions - Whether a user can see all versions of an item in the list or not
- Can view item - Whether a user can view an item in the list or not
- Can view others draft item - Whether a user can see the items that are in draft mode but owned by other users or not
- Can view pending item - Whether a user can see the pending item (for which work is in progress) in the list or not
- Can consume item - Whether a user can consume an item in the list (it may be in lookups or any place where list items gets consumed) or not
- Show all folder - Whether a user can see all folders in the list or not
- Show only owned items - A user can see only the items that they own and cannot see the items owned by others
- Can save as next major version - Whether a user can create a next major version of an item in the list or not
- Can save as next minor version - Whether a user can create a next minor version of an item in the list or not
- Can save as next version - Whether a user can create a next version of an item in the list or not
- Can save as specific version - Whether a user can create a specific version of an item in the list or not

List Specific Rights

Each list is unique and has a different set of actions which can be performed on it. Hence every list can have a specific set of rights appropriate to it.

For example the actions that can be performed on a Workflow list are: Design, Edit, View Workflows etc, whereas the actions which can be performed on a user list are: add , edit, delete users. In the above case, the design operation is specific only to the workflow list. Hence workflow list has a specific right Can design workflow. Similarly, for every list user can define specific rights appropriate to that list.

The following rights are the Workflow List Specific Rights:

- Can design workflow - Whether a user can design the process definition of an item in the workflow list
- Can do tasks - Whether a user can do tasks like design workflow, start workflow wizard, provision now etc on an item in the workflow list

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Can view properties - Whether a user can see the properties of an item in the workflow list
- Can view Design - Whether a user can see the process definition (design) of an item in the workflow list
- Listed below are the Manage Package List Specific Rights
- Run - Whether a user can install the package or not
- Uninstall - Whether a user can uninstall the package or not
- Can Manage Security - Whether a user can manage security at list item or list level of a Manage Package List or not
- Listed below are the Package Template List Specific Rights
- Export- Whether a user can export the package template or not
- Generate - Whether a user can generate the package template or not
- Mark as Shipped - Whether a user can mark the package template as shipped or not
- Can Manage Security - Whether a user can manage security at list item or list level of a Package Template List or not

Process Interface Specific Rights

AVEVA Work Tasks Security also allows process interface specific rights. It offers specific rights for different process interfaces [Process Designer, Web Work items list]. These rights allow or disallow a user to access specific features of the process interfaces.

- Process Designer Specific Rights: For controlling the Process Designer menus
- Can Export - Allows or disallows a user to export a process definition
- Can Import - Allows or disallows a user to import a process definition
- Can Validate Definition - Permits a user to validate a process definition
- Can Set Work Items Fields - Permits a user to define a custom work item field etc

Rights for controlling categories, actions, and properties:

- Can View Categories - Allows or disallows a user to view and use specific categories of Activities in the Process Designer actions toolbox. The user will be able to select the categories that can be viewed or not by a particular user.
- Can View Actions - Allows or disallows user to view and use specific activities in a particular category. The user will be able to select the actions from a particular category that can be viewed and used or not by a particular user.
- Can View Properties - This will allow the user to view, use and configure different properties of an activity. The user will be able to select the properties that can be viewed and used or not by a particular user.

Web Work Items List Control Rights

Rights that are specific to the Work items list control, the rights defined for the Work items list that are applicable at the Repository level:

- Can Set Alternate Actor - Whether a user can set an alternate actor

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Can Set Preferences - Whether a user can set preferences
- Can See Queue - Whether a user can access the queue control
- Can View Resource Calendar - Whether a user can see the resource calendar
- Can View Work Item Control - Whether a user can see the work item control itself
- Calendar Control Rights

This section contains the rights related to the calendar control. The scope of these rights is the Repository level:

- Can Add Calendar - Whether a user can add a new calendar
- Can Add Calendar Workslot - Whether a user can create a new work slot in a calendar
- Can Delete Calendar - Whether a user can delete an existing calendar
- Can Delete Calendar Workslot - Whether a user can delete a work slot of a calendar
- Can See All Calendars - Whether a user can see all the existing calendars
- Can View Calendar - Whether a user can see the calendar control itself
- Queue Control Rights:
- Specific to the Queue Control, these rights are at the Repository level:
- Can View Queue - Whether a user can see the queue control itself
- Can Add New Queue - Whether a user can add a new queue
- Manage Queue - Whether a user who is not the owner of a queue can modify/delete the queue

BAM Control Security Rights

The BAM Control now comes with enhanced security features that allow business users to restrict permissions for various features in BAM. Some of the BAM rights have Repository level scope and some rights are at the List and List Item level.

- Can View Global Reports - Whether a user can view global reports. Scope – Repository level.
- Can View Others Reports - Allows a user to view other users' reports. Scope – List Specific for List Item level.
- Can View Owned Reports - Whether users can view their owned item reports. Scope – List Specific for List Item level.
- Can View Triggered Reports - Whether users can view reports of their own triggered workflows. Scope – List Specific for List Item level.

Dynamic Form Rendering Based on Security

Based on the security applied, different users can render different forms in the dynamic list to edit and also to view details.

Can Edit Item

This **Can Edit Item** right used to specify the whether logged in user can able to edit an item or not. Earlier this right was checkbox. If this right is checked then user have permission to edit an item otherwise the user doesn't have permission to edit an item. Where this is applicable for all levels. Because of this behavior different users

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

were not able to access different forms at List and List Item level.

To avoid the above behaviour, AVEVA Work Tasks has changed the **Can Edit Item** right values to drop-down list with three options **Allow**, **Disallow** and **OwnedItemsOnly** at List and List Item level.

Note: The **Can Edit Item** right displayed as check box at repository level.

The following are **'Can Edit Item'** right value options:

Allow: On selection of this option, it allows the user to edit an item. At the same time, the admin user can do the following:

1. Specify the form name and version to edit
2. Specify any number of forms with different versions
3. Specify the default form for both owned and non owned forms

   Note: User has a provision to select a form and version to render while editing an item, if admin user specified more than one form with different versions.

Disallow: On selection of this option prevents the user from editing an item.

OwnedItemsOnly: On selection of this option allows the user to edit an item, if the logged in user is the owner of that item. At the same time admin user can do the following:

1. Specify the form name and version to edit
2. Specify any number of forms with different versions
3. Specify the default form for owned items

Note: User has a provision to select a form and version to render while editing an item, if admin specified more than one form with different versions.

The above new right descriptions are summarized in a table format as shown below:

S.No	Right Value	Description
1	Allow	If the user has allow permission, then the user can able to edit all items (both owned and others items).
2	Disallow	If the user has disallow permission, then the user not able to edit any items.
3	OwnedItemsOnly	If the use has OwnedItemsOnly permission, then the user can able to edit their owned items only.

Note: When you select either Allow or OwnedItemsOnly options, small image will be shown on the right side of the control. On clicking this image, the admin user can set the form to render while editing.

The Can Edit Item right with the above options are useful in following scenarios:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Scenario 1

Scenario Description

The Administrator wants a user to:

1. View all the items (owned by the logged in user and as well as owned by others) in the list/grid
2. Allow the user to edit their owned items only.

Scenario Quick Solution

With this new enhancement in Can Edit Item right control, the first scenario can achievable with "OwnedItemsOnly" option. If this option set for a user either at list or list item levels, then the user can able to edit owned items only even though the user able to view other user items or non owned items.

Roles

To describe this scenario effectively, assumed that, there are two users 'Admin' and 'John' are existing in a repository and these users are mapped to different security groups at repository level:

The following table depicts the users and their security groups:

User Name	Security Group
Admin	Administrator
John	HR Security Group (new security group)

Scenario Requirements

The admin user creates a new dynamic list named 'EmployeeDetails' list. Within this list, the admin user creates one item and the user John creates one item.

Now, both 'Admin' and 'John' are able to view the both items in the list or grid because **Show Only Owned Items** for both users set to false and both users can able to edit the items because **Can Edit item** set to **True** at repository level with default form (The form which was used to create the list item).

Scenario Solution

As per the scenario, the user 'admin' wants the user 'John' to edit his owned items only in this list. To do the same, follow the steps given below:

1. Login as the administrator user.
2. Go to **Enterprise Console** > **Lists** > **"Employee Details".**
3. In the Employee Details list, click the Security option in the ribbon bar.
4. Select **Advanced Settings** > **Enable List specific Security** link to enable the security for this List.
5. On click of **Enable List specific Security** link, the Enable Security Wizard window is displayed.
6. Select both **Security Group customization** and **Resource Mapping** and then click **Next**.
7. Select **Security Group** customization and then click Next.
8. In the Security Group list, select the HR Security Group (because John belongs to HRSecurityGroup) or right click and select Edit option from the context menu or click **Edit** in the ribbon bar.
9. The **Modify Existing Security group** screen is displayed. Set **Can Edit Item** as **OwnedItemsOnly** and click **Save**.

   Note: On selecting the **OwnedItemsOnly** option, a small image is shown on the right side of the control. But

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

you are not setting any specific form to render while editing, if not set the default form will be rendering while editing.

10. 'John' will able to edit his owned items only. If this user tries to edit the non-owned items, then system will prompt the user stating that "You can edit owned items only".
11. To test the same, login as 'John'.
12. Go to EmployeeDetails list. Try to edit a non-owned item. Select a non-owned item and click **Edit** option in the ribbon bar. The following message appears:

    You can edit your own items only.
13. Right-click the non-owned item. The **Edit** option will not be displayed in the context menu.
14. Now try to edit the owned item. Select owned item and click **Edit** option in the ribbon bar. You can edit the item.
15. Try to right click on the owned item. Select owned item and right click, in the context menu the **Edit** option is available.

Scenario 2

Roles

To describe this scenario effectively, assumed that there are two users 'Admin' and 'Kevin' in a repository and these users are mapped to different security groups at the repository level.

The following table depicts the users and their security groups:

User Name	Security Group
admin	Administrator
kevin	NetworkSecurityGroup (new security group)

Scenario Requirements

The admin user creates a new dynamic list named employee list with the following fields:

- First Name
- Middle Name
- Last Name
- Address 1
- Address 2
- Phone
- DOB
- DOJ
- System Name
- IP Address
- Email
- Salary
- Salary Advance and TDS

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Also created an item in the list. The Administrator wants a user 'kevin' who belongs to Network team to edit list item (network related details) under this list, and at the same time administrator user do not want to show confidential information like Salary, Salary Advance and TDS in the list item while editing. So administrator user decided and created a new form with network related fields to be rendered for the 'Kevin' user while editing.

Note: admin user created a new form with network related fields with different versions as follows:

Form Name: Network_Employee_Details

Versions: 1, 2, 3

Version 1: available fields

Fields
Title
First Name
Middle Name
Last Name
System Name
IP Address

Version 2: available fields

Fields
Title
First Name
Middle Name
Last Name
System Name
IP Address
Email

Version 3: available fields

Fields
Title
First Name
Middle Name
Last Name

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Fields
Phone
System Name
IP Address
Email

All these versions have been published. Now admin user wants the user 'Kevin' edit all the items (both owned and non-owned) within the employee list. But while editing owned items user can use form with any version of the available versions. While editing non owned items user can use the form with version 1 only. By default, form for owned items is the form with version 3 and default form for non owned items is the form with version 1.

Scenario Solution

To do the same, follow the steps below:

1. Login as the administrator user.
2. Go to **Enterprise Console** > **Lists** > "**Employee Details**".
3. In the Employee Details list, click the **Security** option in the ribbon bar.
4. Select **Advanced Settings** > **Enable List specific Security** link to enable the security for this List.
5. On click of **Enable List specific Security** link, the Enable Security Wizard window is displayed.
6. Select both **Security Group customization** and **Resource Mapping** and then click **Next**.
7. Select **Security Group customization** and then click **Next**.
8. In the **Security Group** list, select the NetworkSecurityGroup (because kevin belong to this security group) either right click and select **Edit** option from the context menu or click **Edit** in the ribbon bar.
9. The **Modify Existing Security group** screen is displayed. Set **Can Edit Item** as Allow and then click **Save**. On selecting the allow option small image will appear on the right side on the control, click on the image.
10. On clicking of this image, the **Form Lookup** screen is displayed.
11. In the above **Form Lookup** page, the first section (Forms) specifies the forms to render both owned and non owned items. Select a form and version and click on add (>) button, the selected form will be added in the selected forms section on the right side. If a form selected with the "Form for Owned Items" option checked then that meant for owned items.

    The second section (Default Form for Owned Items) specifies the default form for owned items. The form which was used to create an item is populated by default.

    The last section (Default Form for Non-Owned Items) specifies the default form for non-owned items. The form which was used to create an item is populated by default.

    Note: In this **Form Lookup**, only published forms only available.
12. In this scenario, admin user wants the user 'Kevin' edit all items (both owned and non-owned) and while editing owned items user can use the form (Network_Employee_Details) with any version of the available 3 versions, and while editing non owned items user can use the form (Network_Employee_Details) with version 1 only. And default form for owned items is the form with version 3 and default form for non owned items is the form with version 1.
13. First, set the form with versions 1, 2 and 3 for owned items.
14. Similarly add the version 2 and 3.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

15. Now for owned items, you have selected 3 versions and for non-owned items. You need to select only the version 1. Select the form name and select version 1 and add the form with out "Form for Owned Items" option checked.
16. Next, set the default form for owned items. In our scenario, default form for owned items is Network_Employee_Details form with version 3. So select the form name and version and click on add (>) in the "Default form for owned items" section.
17. Finally, you have to select the default form for non owned items. In our scenario the default form for non owned items is Network_Employee_Details form with version 1. So select the form name and version and click on add (>) in the "Default form for Non-Owned Items" section.
18. After selecting all required forms click on **Continue** button. This window will be closed and again you will be getting the "Modifying Existing security group" screen click on **Save** and **Synchronize**.
19. Now login with the 'Kevin' user credentials and go to 'EmployeeDetails' list.
20. For our example, in the employee list, you have 2 items. For one item admin is the owner and for another item 'Kevin' is the owner.
21. As per our scenario, 'Kevin' user able to edit both owned and non-owned items. First, try to edit the owned items. Select the owned item and right click on that item.
22. As per our scenario, while editing owned items user can use the form Network_Employee_Details form with version 1, 2 and 3. So when the user right clicks on owned item user will get the edit form name versions.
23. When the user with out selecting the form version clicks on **Edit** option in the context menu then the default form for owned items will be rendered.
24. Similarly, when the user selects an owned item and clicks on the Edit option in the ribbon bar then the default form for owned items will be rendered.
25. Now we will try to edit non-owned items. As per our scenario while editing non-owned items user can use Network_Employee_Details form with version 1 only.
26. Select a non-owned item and right click on that item.
27. As there is only one version, the sub menu for "Form name" will not appear. On clicking the form name, version 1 opens.
28. On clicking **Edit** in the context menu, the default form for non-owned items will be rendered.
29. Similarly, on selecting non-owned item and clicking **Edit** on the ribbon bar, the default form for non-owned items opens.

Enable and Disable Security for a Repository

Introduction

This section explains how to enable or disable the security at repository level. Before you enable/disable security, you must know what is security and how the security will effect the application.

How Does Security Affect the Application?

AVEVA Work Tasks has new enterprise rights and security features. The rights and security can be defined for groups and users can be mapped to groups. The user will be entitled to perform only those tasks for which the right has been given to respective user.

AVEVA Work Tasks has some pre-defined security groups and each security group has default security rights.Pre-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Defined Security Groups:

Security Group Name
Administrator
Contributor
Reader
Limited Access

In AVEVA Work Tasks, while creating a repository: user will be asked to set the administrator at repository level. Once you select a user, then the selected user has been added with administrative privileges. This user can create new users, new roles, and new security groups, mapping users/roles with different security groups at different levels (either repository or list or list item levels) and setting security rights for different security groups at different levels.

How to Check Whether Security is Enabled or Disabled for a Repository?

Whether security is enabled or disabled for a specific repository, depends on the **Properties** Column (properties collection) of **SKRepository** table in the Farm Database.

For example, find below the **Properties** column (properties collection) data for a repository.

```
<properties>
<property key="sourceelements">
<![CDATA[WORKFLOWELEMENTS\Default]]>
</property>
<property key="fromemail"><![CDATA[admin@domain.com]]></property>
<property key="toemail"><![CDATA[admin@domain.com]]></property>
<property key="alertuser"><![CDATA[False]]></property>
<property key="filewatcher"><![CDATA[]]></property>
<property key="enablesecurity"><![CDATA[True]]></property>
<property key="isadminconfigured"><![CDATA[True]]></property>
<property key="theme"><![CDATA[Default]]></property>
</properties>
```

The line highlighted in red in the above code, is used to identify whether security is enabled or disabled for a repository. In this line, you can identify the following:

<property>	**defines the property tag**
key	Attribute of property tag, describes the property name (enablesecurity is name of the property)
<![CDATA[True]]>	It is the value of the property (True- security is enabled)
</property>	Closing tag of property

Enabling security for a repository

In AVEVA Work Tasks, once you create a repository, the security will be enabled by default for that repository.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

If you want to enable security for a repository, then open the SKRepostiory table in the Farm Database and find out the required repository entry (row) and in that specific row in properties column data (properties collection) find out the enablesecurity key and update the <![CDATA[False]]> to <![CDATA[True]]>.

Note: If the enablesecurity key is not found for a repository entry in the properties column then add the following line between <properties> & </properties> tags.

```
<property key="enablesecurity"><![CDATA[True]]></property>
```

Now the security is enabled for that repository. If you want to enable security for another repository do the same as mentioned above.

Disabling security for a repository

In AVEVA Work Tasks, once you create a repository security will be enabled by default for that repository.

If you want to disable security for a repository, for which the security already enabled, then open the SKRepostiory table in the Farm Database and find out the required repository entry (row) and in that specific row in properties column data (properties collection) find out the enablesecurity key and update the <![CDATA[True]]> to <![CDATA[False]]>.

Note: If the enable security key not found for a repository entry in the properties column then add the following line between <properties> & </properties> tags in case if you want to enable security in future.

```
<property key="enablesecurity"><![CDATA[False]]></property>
```

Default Behavior of Repositories Created in AVEVA Work Tasks

In AVEVA Work Tasks, if you create a repository, then security will be enabled by default for that repository. While creating repository user will be asked to set the administrator for that repository, once the user select the user for administrator, the selected user will be added with administrative privileges for that repository.

Once the administrator user is logged in, they can access the security settings (Repository Settings – top right corner). Any user other than the one belonging to the **Administrator** security group cannot access the security settings page

Http Strict Transport Security Enforcement

You can enforce a Http Strict Transport Security to Enterprise Console sites in AVEVA Work Tasks for secured access. When the security is enforced, only the secured sites will be loaded.

To enforce Http Strict Transport Security, add the code in the Enterprise Console **web.config** file available in <Installed path>\Web\BPMUITemplates\Default> as given below:

```
<system.webServer>
    <httpProtocol>
        <customHeaders>
           <add name="Strict-Transport-Security" value="max-age=31536000"/>
        </customHeaders>
    </httpProtocol>
</system.webServer>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Access-Control-Allow-Origin

By default, the Access-Control-Allow-Origin tag value in the Enterprise Console Site of AVEVA Work Tasks is * to allow all Origins.

To restrict, provide the specific Origin values instead of *.

You need to update the Origin values in case AVEVA Work Tasks is connected through Azure Relay.

Security Vulnerabilities

A security vulnerability is a flaw in a product that could allow an attacker to compromise the integrity, availability, or confidentiality of that product. For the software security industry, a vulnerability is a security exposure that results from a product weakness that the product developer did not intend to introduce and should fix once it is discovered.

There are various known security threats such as, XSS - Cross site scripting, SQL injection, and URL Spoofing. AVEVA Work Tasks tries to mitigate most of the security vulnerabilities through SDL approaches.

Verbose HTTP header is another type of security vulnerability. To mitigate this security vulnerability, see Removing Unwanted HTTP Response Headers.

Remove Unwanted HTTP Response Headers

Verbose logging tracks all changes and settings applied using Group Policy and its extension to the local computer and to users who log on to the computer. The log file is located at SystemDrive\Debug. This folder is a hidden folder.

To enhance security measures and remove the unwanted HTTP response headers, you can use the following available provisions:

- Web Configuration Changes
- Native HTTP Module

Web Configuration Changes

```
In the Enterprise Console Web configuration file, add the following code inside the
System.WebServer tag
<system.webServer>
<validation validateIntegratedModeConfiguration="false" />
<modules runAllManagedModulesForAllRequests="true">
<add name="HideServerHeaderModule"
type="Wonderware.Workflow.SecurityConfig.HideServerHeaderModule,
Wonderware.Workflow.SecurityConfig, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=1add7d6ea5a31926" />
</modules>
```

Native HTTP Module

```
After installing AVEVA Work Tasks, ensure to have the following StripHeaders entries in the
application configuration in the following path:
C:\Windows\System32\inetsrv\config\applicationHost
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<location path="WonderwareSkeltaBPMSites/CentralConfig">
    <system.webServer>
        <stripHeaders>
                <header name="Server" />
                <header name="X-Powered-By" />
                <header name="X-Aspnet-Version" />
        </stripHeaders>
        <security>
            <authentication>
                <basicAuthentication enabled="false" />
                <anonymousAuthentication enabled="false" />
                <windowsAuthentication enabled="true" />
            </authentication>
        </security>
        <handlers accessPolicy="Read, Write, Execute, Script" />
    </system.webServer>
</location>
<location path="WonderwareSkeltaBPMSites/EnterpriseConsole">
    <system.webServer>
        <stripHeaders>
          <header name="Server" />
          <header name="X-Powered-By" />
          <header name="X-Aspnet-Version" />
        </stripHeaders>

        <security>
            <authentication>
                <basicAuthentication enabled="false" />
                <anonymousAuthentication enabled="false" />
                <windowsAuthentication enabled="true" />
            </authentication>
        </security>
        <handlers accessPolicy="Read, Write, Execute, Script" />
    </system.webServer>
```

```
StripHeaders is a third party module that you can download from the Internet.
```

Prevent Clickjacking Attack

Clickjacking is a malicious technique of tricking a web user into clicking on something different from what the user perceives they are clicking on. This potentially reveals the users confidential information. It is a browser security issue and is a vulnerability across a variety of browsers and platforms. The **X-Frame-Options** header is known to be a good measurement against the Clickjacking attacks.

AVEVA Work Tasks is secured against **Clickjacking** attacks as **X-Frame-Options** are set for **SAMEORIGIN** value. **SAMEORIGIN** means that the page can only be displayed in a frame on the same origin as the page itself.

For custom pages, we recommend that the page load event must have commands to remove the **X-Frame-Options** header if other application references are in the page. Use the following command to remove the **X-Frame-Options** header:

```
   Page_Load()
   HttpContext.Response.Headers.Remove("X-Frame-Options");
```

Alternatively, add appSettings key in Repository *web.config* file available in <Installed path>\Web\BPMUITemplates\Default as given below:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<add key="X-Frame-Options_SameOrgin" value="OFF"/>
```

SDL Cookie with Secure Flag

AVEVA Work Tasks uses cookies for various purposes. However, these cookies might be utilized by unauthorized parties and might be prone to security vulnerabilities. Therefore, to secure the cookies, it is recommended to use the **httpOnly** attribute in case of Http protocol for most of the cookies.

To ensure that the application is more secure, it is recommended to use the **https** protocol for AVEVA Work Tasks web application.

1. Browse to the location *<Install Path>\Web\EnterpriseConsole*.
2. Edit the **web.config** file.
3. Add the following tag.

```
<system.web>
<httpCookies httpOnlyCookies="true" requireSSL="true" lockItem="true" />
</system.web>
```

4. Save the file
5. Browse to the location *<Installed Path>\Web\CentralConfig*.
6. Repeat steps 2 to 4.

Troubleshoot Security Issues

This section tells you how to troubleshoot security problems.

View Others Items in the Lookup Control

Why user is not able to view other user items in Lookup Control?

Problem: User is unable to view the list items in Lookup Input control.

Resolution: Check the following steps to view the other list items in Lookup Input.

1. Check user security group and rights.
2. Make sure that **Show Only Owned Item** right is set to **False** for that user.
3. Make sure that **Can Consume Item** and **Can View Others Draft Items** right is set to **True** for that user.

Example: This example demonstrates why the user is not able to view items in the Lookup Input control and how to resolve this issue.

Let assume that there is a List called 'Employee List'. Consider that the Employee list consists the Employee name, Employee Id, Department, and Designation Fields. Where Employee name and Employee Id fields are Textfields and Department and Designation Fields are Lookup Input controls. Say Department field fetches the list items from 'Department' list and Designation field fetches the list items from 'Designation' list.

The problem is that user is not able to see all the items in 'Department' and 'Designation' lookup input controls.

The solution is to login as the administrator user and check the group to which the user belongs and what are the security rights defined for the user. Ensure if **Can Consume Item** is selected or not? If not, set this right to

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

True. If the user is still not able to view, then uncheck **Show Only Owned Item** right. If the user is not able to view other's draft items, then select **Can View Others Draft Item** right also. Now the user should be able to view all the Department and Designation items in the lookup input controls.

View Others Items in the Grid Control

Why the user is not able to view other user items in the Grid?

Problem: The user is unable to view the list items in the grid.

Resolution: Follow the steps given below to view the other list items in the grid:

1. Check the user security group and rights.
2. Ensure that the **Show Only Owned Item** right is set to False for the user.
3. Ensure that the **Can Consume Item** and **Can View Others Draft Items** rights are set to **True** for the user.

View Others Items in the List

Why the user is not able to view items of other users in the list?

Problem: The user is unable to view the list items in the list.

Resolution: Follow the steps given below to view the others list items in the list:

1. Check the user security group and rights.
2. Ensure that the **Show Only Owned Item** right is set to **False** for the user.
3. Ensure that the **Can Consume Item** and **Can View Others Draft Items** rights are set to **True** for the user.

View Others Items Irrespective of Status

Why the user is not able to view others items irrespective of status?

Problem: The user is not able to view others items even though the Can Consume Item and Can View Others Item is checked for the user.

Resolution: Follow the steps given below to view others items:

1. Ensure that the **Show Only Owned Item** right is set to **False** for the user.
2. Ensure that the **Can Consume Item** and **Can View Others Draft** Items rights are set to **True** for the user.
3. In this case, the **Show Only Owned Item** right takes precedence over the **Can Consume Item** right. That is why the user is unable to view others items.

Navigation List Menus

1. Why the user who belongs to the Reader Security group is not able to consume menu items in the Enterprise Console pull-down menu?

The user is not able to consume menu items in the Enterprise Console pull-down menu even though the Can Consume Item right is checked.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Resolution

This problem usually happens when the users create their own menu and submenu items and would like to define security for those menus and submenus. In security, by default, the newly created menus and submenus will be considered as draft mode.

To view the others items:

Ensure that the **Can View Others Draft Items** right is set to **True** for the user.

2. Upgrade Manager has detected some duplicate entries in the Navigation list. Re-create the entries with unique titles.

Resolution

While upgrading **AVEVA Work Tasks**, if the **Upgrade Manager** detects some duplicate entries in the **Navigation** list, this message is logged in the **Logger**. You can re-create the existing duplicate link entries with unique titles.

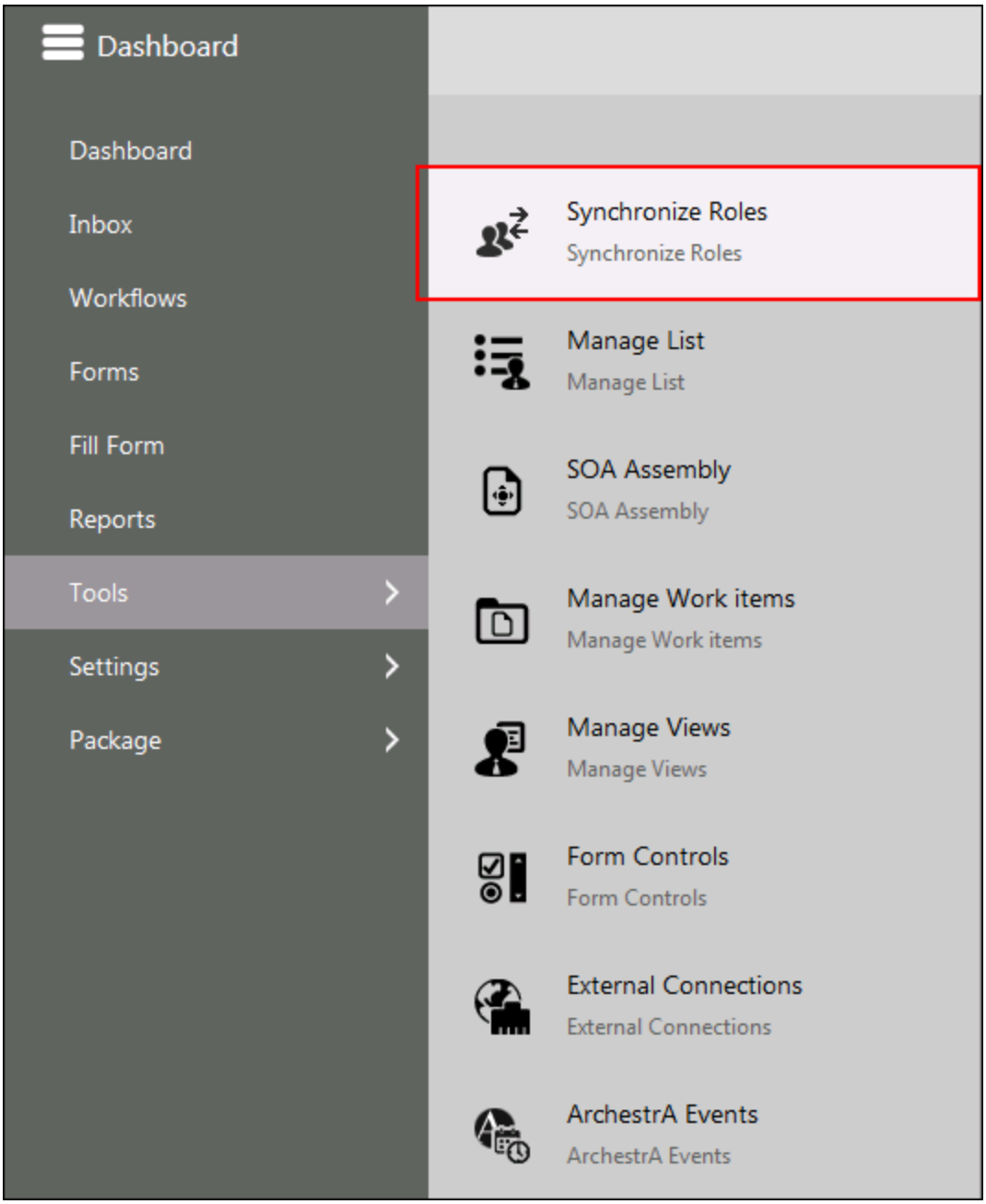
Synchronize Roles

If you want cross domain users and roles mapping across active directory provider, then use the **Synchronize Roles** option to map the roles.

Example: Users 'John' and 'Jim' of domain 'TelematicsUS' and users 'Mike' and 'Bob' from the domain 'TelematicsAU' are added to the role "Team HR" of the domain 'TelematicsMain'.

You can see the Synchronize Roles list item in the Enterprise Console menu that you can use to map the cross domain provider roles. This list item is not available by default in the Enterprise Console menu. While creating a repository, if you choose **Resource Provider** as **Active Directory** and enable the **'Is Cross Domain Role Provider'** option for cross domain users and roles mapping, then the **Synchronize Roles** sub-menu appears in **Enterprise Console** > **Tools**. This menu is available only for users with administrative rights.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



To synchronize roles

1. In the Enterprise console, click the **menu** icon, click **Tools,** and then click **Synchronize Roles**.
   The Synchronize Roles window appears.
2. Enter the role name in the **Search Role** text box.
3. Click **Search**.
   The available roles will be listed in **Choose Roles**.
4. Choose the required role to be synchronized.
   For example, search for 'team' role and it will list the available roles.
5. Click **Synchronize**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

After the synchronization of the selected role(s) is complete, a confirmation message about successful synchronization appears.

Delete Synchronization Cross Domain Tasks

To delete synchronization cross domain tasks

1. Go to Repository Data Source Database.
2. Open **SWTaskList** table.
3. Delete the required synchronization cross domain tasks for repository.

Note: Delete only those records whose Description column is 'Cross domain Role Refresh Task' for that repository.

Web Parts

Web Parts is an integrated set of controls for creating Web sites that enable end users to modify the content, appearance, and behavior of Web pages directly from a browser. The modifications can be applied to all users on the site or to individual users. When users modify pages and controls, the settings can be saved to retain a user's personal preferences across future browser sessions, a feature called personalization. These Web Parts capabilities mean that developers can empower end users to personalize a Web application dynamically, without developer or administrator intervention.

Personalization

Personalization is the core behavior of WebParts. You can modify or personalize the layout, appearance, and behavior of WebParts controls. Personalized settings are persisted for future browser sessions.

Web Parts Manager

Manages all Web Parts on a page, enables or disables personalization, and manages the life cycle of personalization data. One (and only one) WebPartManager control is required for every Web Parts page. Web Part Manager also handles security for the Web Pages including the Adding, Moving, and Deleting the web parts.

Personalization Scope

Personalization scope refers to how personalization data is applicable to different sets of users. Some personalization data is applicable to only a specific user and control on a page. Other personalization data is applicable to all users who view a site.

Web Part Zone

The WebPartZone control serves as the primary control in the Web Parts control set for hosting ASP.NET server side controls on a Web page.

In addition to web parts that inherit from the WebPart class, a WebPartZone control can host controls that do

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

not derive from the WebPart class, by wrapping them with a GenericWebPart control at run time. Consequently, virtually any ASP.NET server control or user control can function as a web part when used inside a WebPartZone control.

The EditorZone control represents the means through which end users may modify and personalize Web pages according to their preferences. Personalizing a Web site includes such things as setting up personal information (such as work item list control, calendar, queue control, etc.). Other kinds of personalization involve setting up appearance and layouts. The EditorZone helps manage this functionality as well as saves and loads those settings so they're available the next time the user logs on.

AppearanceEditorPart

This is used to edit properties that affect the appearance of an associated WebPart or GenericWebPart control.

The properties listed under the Appearance tab are:

Property	Description
Title	This field displays the title of the web part. If the user wants to change the title of the web part, it can be changed in this field.
Chrome Type	This property is used to specify the appearance of the title and border of the web parts. Users can choose to show only the title of the web part or only the border of the web part or none of the two.
Direction	The Direction field is used to set the direction of the web part.
Height	This property is used to set the height of the web part. Users can give the height specifications in percent, pixel, points, picas, inches, millimeters, centimeters, percent, em, ex.
Width	This property is used to set the width of the web part. Users can use this property to change the width specifications in percent, pixel, points, picas, inches, millimeters, centimeters, percent, em, ex.
Hidden	This property can be used when the user wants to hide the view of any web part. The user can check the Hidden property check box and that particular web part will be hidden for the users.

LayoutEditorPart

The LayoutEditorPart is used to edit properties that affect the layout of an associated WebPart or GenericWebPart control.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Chrome State	This property is used to specify the layout of the web part. Users can choose to display the web part either in Normal scope or minimized scope.
Zone	This property can be used to specify the zone for a web part. If the user wants to change the zone of a web part, them you can select a zone from the drop-down list.
Zone Index	This property is to be used when the user has put multiple web parts in a zone. Users can specify the order of appearance of the web part by giving a number for each web part in that zone. By default, the numbering begins from 0. The first web part to appear in the zone should be numbered 0 and the second web part should be numbered 1.

PropertyGridEditorPart

The PropertyGridEditorPart provides a generic user interface (UI) that enables users to edit custom properties on WebPart controls. In contrast, the other EditorPart controls, such as the AppearanceEditorPart and BehaviorEditorPart controls, edit only existing, UI-oriented properties from the WebPart class.

Each web part has custom properties which are listed under the respective web parts.

Use Web Parts Infrastructure

The Enterprise Console provides a rich set of tools and user controls to manage, control workflows, users. All the Controls can be accessed inside the Enterprise console based on the security rights.

Personalization of the controls play an important role in many of the scenarios where in which user can configure their configuration or settings for the control overriding the global or public configuration or view.

And also in many scenarios, there is a requirement of setting the customized home page for the application site where the Controls should be able to place inside a specific layout and configure properties for the same.

Web Parts Framework

The Web Part Framework allows the users of Enterprise Console to be able to customize the content by selecting, removing, and rearranging the contents in the web page(Home Pages).

The Web Parts Framework provides the infrastructure and the building blocks required for creating modular web pages, which can be easily customized by the users. You can use Web Parts to create portal pages that aggregate different types of content, such as Work Item List, BAM, Calendar, Queue, List controls, and the contents that can change at runtime. The users can also change the layout of the Web Parts by dragging and dropping them from one place to another and providing a rich user experience.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

By taking advantage of Web Parts, the EC users can perform the following operations:

- Users can personalize page content.
- Users can personalize the page layout by allowing the users to drag Web Parts from one. zone to another zone, or change its appearance, look and feel, and so on.
- Users can share the page content with others.

Apart from the out-of-the-box Home pages, users can create their own custom web part pages, with different layouts by providing the Web part zones, to meet their business requirement.

Similarly, the user can include their own custom Web Parts as well, to use in the Enterprise console. They can use these custom web parts along with the controls in a single page, without providing a single line of code.

Out-of-the-Box Web Parts

AVEVA Work Tasks provides the following six out-of-the-box web parts:

BAM

The Business Activity Monitoring Dashboard is a key AVEVA Work Tasks Server Control. It enables you to monitor the execution of a workflow and provides an array of real-time reports. This report includes status and performance of workflows.

Property	**Description**
Hide Categories	To specify multiple categories to be hidden. Use semicolon to separate the categories. Example: Dashboard;Analyze
Displays the table border for individual report	To specify the table border for individual reports.
Instances To Be Shown	To specify multiple instances. Use semicolon to separate the instances. Example: 123;124
Show Individual Report	To show individual reports in the BAM.
Report Description	To provide description of a report. This is applicable for individual report mode only.
To show an individual report ['Id' field of 'SKBAMMenu' Table]	To show an individual report. Example: *Id* field of *SKBAMMenu* table
To show an individual report [Report Name with hierarchy.]	To provide a name to a report. Example: Analyze/Executing
Report title	To provide a report title for an individual report. This is applicable for individual report mode only.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
To toggle the visibility of the report border	To change between the visibility of the report border. This is applicable for individual report mode only.
Number of records to export	To set the number of records to export in a grid. This setting is applicable for all grids in the Report page for any instance. If this field is left blank, it takes the default size of 50 records. For the settings to take effect, save the details and then navigate back to the Reports page from the **Enterprise Console** menu. You can also set the *number of records to export* at a grid level using the **Export Settings** icon on the **Reports** page. Setting the number of records to export using the **Export Settings** icon on the **Reports** page takes precedence over the Web part level setting for a particular instance. For example, if you set the number of records to export as 10 using this option, and then for a particular grid, if you set the number of records to export as 20 using the **Export Settings** option, then 20 records will be exported for that particular grid for that particular instance. For more information about **Export Settings**, see Business Activity Monitoring and Reports.
Workflow Name	To specify multiple workflows. Use semicolon to separate multiple workflow names. Example: workflow1;workflow2

Calendar

You can use the Calendar interface to create hierarchical organization and resource calendars. Use Calendars to determine availability of resources, calculate timeouts, and perform other scheduling functions.

Property	Description
Calendar Name	To give a name to the calendar
Can Add calendar	Right to add a new calendar
Can Add calendar workslot	Right to add a calendar workslot
Can Add Resource Calendar	Right to add a resource calendar

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Can Delete calendar workslot	Right to delete a calendar workslot
Can Delete calendar	Right to delete a calendar

Queue Management

You can use the Queue Management to create and manage queues for allocating workflow activities. This is done using the Queue control.

Queue control consists of only **Appearance** and **Layout** properties. It does not comprise of any property grid.

Work Items List

The Work items list is a Web-based interface which lists the various workflow activities in AVEVA Work Tasks. The Work items list provides detailed information of an activity. You can view and respond to work items from an interface similar to Microsoft Office Outlook.

Property	Description
Allow ownership on row click automatically	To take ownership of the work item on clicking the row.
Display the columns with autofit	Display the columns with autofit in the layout.
Enable Multi Select	To select multiple work items.
Display left tree menu without grouping the workflow(s)	To view the Work items list page without grouping the workflow(s).
Enable Move to the next work item after submitting the work item from the detail view	To move to the next work item as soon as you submit a work item from the detailed view. **Note:** When the **Show reading pane in pop-up window** option is checked, and when an Invoke Form work item (configured with **Save as draft** option as yes), is opened and the **Save as draft** button is clicked, then the pop-up window closes. When the **Show reading pane in pop-up window** option is unchecked, and when you click on **Save as draft** button, the form is saved as draft, but the form content is still displayed inline.
Display Classic Paging	To display the Classic Paging properties.
Enable Export	To control the visibility of the **Export** icon in the work

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	item details grid status bar.
Enable Hold	To put work items on hold. The **Hold** button would be visible for a work item if hold is enabled.
Display Next and Previous buttons in the work item detail view	To navigate to the next or previous work item from the work item detailed view page.
Display multi-select option, by default, for queue items	To display queue items with the multi select option enabled by default.
Enter the average column name	To calculate the average of columns when the grouping operation is performed on columns. You can specify multiple column names with semicolon. Example: Expense Amount,true;Leave Days,false
Enter the group count column name	To calculate the count of the columns when the grouping operation is performed on columns. You can specify multiple column names with semicolon. Example: Leave Days;Expense Type
Enter the group total column name	To calculate the total of the column names when the grouping operation is performed on columns. You can specify multiple column names with semicolon. Example :Expense Amount,true;Leave Days,false
Show inbox details without left grid view	To display only the **Inbox** view without the folder list to the left tree menu.
Layout View Name	To load a specific layout. Multiple workflow names can be specified with semicolon. Example:Leave Application;HRProcess.
Show Manager View	To display only the **Manager View** for the list.
Show reading pane in pop-up window	To display the work item details in a separate window and not in the **Reading Pane** after the **Inbox**.
Record Size	To display the number of messages in the work items based on this property. For example, assume that there are 35 messages. If this property is set as 20, then 20 messages out of 35 are displayed in the work items.
Retain the sort column	To retain the selection of recently sorted column after submitting the work item and also while expanding the left tree menu view items. If this is set to true,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
	then selection of the recently sorted column is retained. If set to false, then the selection of the recently sorted column is not retained.
Display the count on the left side of the grid view	To display the work items count in the left-hand-side grid view.
Show a custom link on the left side of the grid menu	To display a custom link in the left pane of Work items Web Part.
Show detail view on row double-click	To open the work item details page (Custom ASPX page) on double-click.
Show detail view as modal window	To choose the detail view to appear as a modal window or a modeless window when you click a work item to edit.
Show detail view with multi-select option enabled	To display the detail view with the multi-select option enabled by default.
Show filter bar always	To display or hide the filter bar.
Show multi select window in the center	To display a multi-select window in the center of the page.
Show queues	To display a folder for queues that the user is a participant in.
Show queue detail view without picking	To act on a work item without picking it.
Queue Name	To display the queue name in the left tree menu of the **Work items Web Par**t. If the queue name is not specified, all available queues appears in the left tree menu. Multiple queue names can be specified with semicolon. Example: Manager;Support
Show queues on the left-side of grid view menu only if the queues have work items	To show queues in the left-hand-side grid view menu only if they have work items.
Show standard columns	To displays the work item fields columns based on the **Workflow Name** property of work item control.
Workflow Name	To display the workflow name in the left tree menu of the Work items Web Part. If the workflow name is not specified, all available workflows appears in the left tree menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Show Work Item Fields	To display work item fields.
Display pop-up messages while acting on a Queue item in Inbox	To enable or disable pop-up messages in the inbox while performing actions on queue work items in inbox. For example, if you do not enable this option, and when you pick and recall an allocated item, the pop-up window with information message does not appear. By default, this field is enabled.

Forms

A Form is used as an interface to enter data in the Lists. The Forms Designer is an intermediate layer between a user and the Forms Engine. The Forms Designer is an interface that you can use to design, customize, and maintain form templates, and define the business logic specific to a form. Forms are associated with workflows. These workflows are responsible for driving end-to-end business applications. A form triggers a business process.

You can use the Forms web part to fill a published form by using the Fill Form in the Enterprise Console menu. To access the Forms Web Part page, you must have **Can Fill Form** permissions in AVEVA Work Tasks.

The following table explains the properties of Form Web Part:

Property	Description
Form Name	To specify a name for the form.
Form Display Mode	To select the display mode from the drop-down list. By default, Form Display Mode property is set as *Tree* mode.
Show Fill form as Pop-up	To view the Fill Form as a pop-up on the screen. If this option is not checked, the form is loaded inline to the tree view.
Inline Form Height	To specify the height of the inline form.
Tree View Height	To specify the height of the tree view of the web part.

List

The List is a collection of items such as Task lists or Item lists captured using Forms. It acts as a logical table (similar to a database table) but involves a more complicated structure to represent an entire entity with links to child tables. The data entered in the List is stored in the Repository. It stores application-level data thus facilitating generation of business reports. AVEVA Work Tasks lists group related information.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Property	Description
Menu Control Path	To specify the path for the Menu Control. Example: /Repository/ListControl/WorkflowRibbonBar.ascx
List Name	To provide a name to the list.
Enter List ViewName to the view	To provide a name to the view.
Show header	To display the header.
Show ListViews	To display the list views.
Show ribbon bar	To display the ribbon bar for the lists.

Other Web Part Properties

While creating a web part, the user can access the Editor zone and specify the web part properties. To view the editor zone, click the drop-down arrow in the right-end corner of the home page.

The Appearance and Layout properties are common for all the web parts. The properties and their descriptions are given below:

Appearance

- Title: The **Title** field displays the title of the web part. If the user wants to change the title of the web part, it can be changed in this field.
- Chrome Type: This property is used to specify the appearance of the title and border of the web parts. Users can choose to show only the title of the web part or only the border of the web part or none of the two.
- Direction: The Direction field is used to set the direction of the web part.
- Height: This property is used to set the height of the web part. Users can give the height specifications in percent, pixel, points, picas, inches, millimeters, centimeters, percent, em, ex.
- Width: This property is used to set the width of the web part. Users can use this property to change the width specifications in percent, pixel, points, picas, inches, millimeters, centimeters, percent, em, ex.
- Hidden: This property can be used when the user wants to hide the view of any web part. The user can check the Hidden property check box and that particular web part will be hidden for the users.

Layout

- Chrome State: This property is used to specify the layout of the web part. Users can choose to display the web part either in Normal scope or minimized scope.
- Zone: This property can be used to specify the zone for a web part. If the user wants to change the zone of a web part, them you can select a zone from the drop-down list.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Zone Index: This property is to be used when the user has put multiple web parts in a zone. Users can specify the order of appearance of the web part by giving a number for each web part in that zone. By default, the numbering begins from 0. The first web part to appear in the zone should be numbered 0 and the second web part should be numbered 1.

Authentication and Personalization

For the users to be able to personalize the web parts, they have to ensure that the Anonymous Authentication is disabled and the Windows Authentication is enabled. If the Anonymous Authentication is enabled, then the user will be able to view the web parts only in browse mode and will not be able to add, edit, or delete a web part.

To be able to Add, Edit, or Delete the Web Parts

1. Click Start.
2. Point to **Administrative Tools** > **Internet Information Services (IIS) Manager**. The **Internet Information Services (IIS) Manager** screen appears.
3. Expand **Sites** and select **Enterprise Config** site. Then select **Authentication** from the right pane.
4. Ensure that the authentication settings are as follows:
   - Anonymous Authentication: Disabled
   - ASP.NET Impersonation: Disabled
   - Basic Authentication: Disabled
   - Forms Authentication: Disabled
   - Windows Authentication: Enabled

If the Anonymous Authentication has been changed to 'Enabled' instead of the default 'Disabled' state as shown in the image given below, then the user has to disable Windows Authentication and enable Forms Authentication.

Follow the steps given below to change the Windows Authentication:

1. Right-click the **Windows Authentication** Mode and select **Disable**.
2. Right-click the **Forms Authentication** Mode and select **Enable**.

Note: This always works in the above mentioned combination. If the Forms Authentication is enabled then the Windows Authentication must be disabled and the Anonymous Authentication must be enabled in the IIS Manager. If in the above scenario, the Anonymous Authentication is not enabled, then the Auto-redirection for Active Directory will skip the login page and load the web parts only in Browse mode. In order to be able to add, edit, or delete the web parts, the user has to enable the Anonymous Authentication.

If the user changes the authentication mode to Forms, then the code given below should be added to the web.config file:

```
<authentication mode="Forms">
<forms timeout="500000" loginUrl="~/BPMUITemplates/Default/Repository/Site/Login.aspx">
</forms>
</authentication>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create Custom Web Parts

AVEVA Work Tasks provides six out of the box web parts. However, you can also create your own custom web parts to be used in the Enterprise Console.

For example, consider a scenario where you would like to see time zone information on the dashboard. The time zone information can be displayed on the dashboard by creating a custom web part and then adding it to the dashboard web part settings. You can select it from the dashboard web part settings and add it to the specified web part zone.

Note: You can customize the following sample code to create your own custom web parts.

To create custom web parts

Code to create a class library:

Note: In the class libarary, ensure that you add a reference to Skelta.WebParts and also inherit BaseWebPart as Skelta.WebParts.BaseWebPart.

```
using System.Data;
using Workflow.NET.Interfaces;
using Workflow.NET.Storage;
namespace CustomWebPart
{
public class TimeZone : Skelta.WebParts.BaseWebPart
{
protected override void CreateChildControls()
{
base.CreateChildControls();
System.Web.UI.WebControls.GridView gridView = new System.Web.UI.WebControls.GridView();
gridView.DataSource = this.GetData();
gridView.DataBind();
this.Controls.Add(gridView);
}
private DataTable GetData()
{
DataTable table = new DataTable();
IDataHandler dbHandler = DataHandlerFactory.GetDataHandler(new Workflow.NET.Config(Applicat
ionName));
using (dbHandler)
{
string query = "Select RegionalSetting, UTCDifference from SKTimeZones";
table = dbHandler.ReadData(query, null);
}
return table;
}
}
}
```

Place the dll in *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Bin* and AVEVA Work Tasks*Web**EnterpriseConsole**Bin* folders.

Note: If the assembly is strong named, place it in GAC and also add an entry in the web.config file.

Add a row in the SKAddinProvider table using the script given below: INSERT INTO [dbo].[SKAddInProviders] (Type,[Name],ClassName,Assembly) VALUES ('WebPart','<WebPartName>','<ClassName>','Bin\ProjectName.dll')

Navigate to **Enterprise Console** > **Settings** > **Web Part Pages**. Select any AVEVA Work Tasks home page.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Click the expand (+) icon of the Webparts Settings. The Webparts Settings pane expands.

In the **Select Web Part** list, select **TimeZone**.

In the **Select Zone** list, select **Web Part Zone Left**. Click **Add Web Part**. The TimeZone web part appears on the left zone of AVEVA Work Tasks home page.

Web Part Display Modes

AVEVA Work Tasks provides three display modes:

Browse: The **Browse** mode is a view-only mode.

Design: The **Design** mode allows the user to even delete the web part along with the **Minimize** and **Restore** options.

Edit: The **Edit** mode allows the user to edit the properties of web part.

Browse Mode

AVEVA Work Tasks provides three display modes - Browse, Design, and Edit in their home pages.

The Browse mode allows the user to only view the web page. The user can either minimize or restore the web page view. The user will not be able to delete or edit the web part in this mode.

To view the Browse mode

1. Go to **Enterprise Console** > **Settings** > **Web Part Settings** > **Page 1**.
2. Expand the **Webparts Settings** bar.
3. Click the **Browse** mode from the Web Part Display Mode drop-down list.
   By default, the home page is in **Browse** mode. You can see that the browse mode allows the user only to minimize the web part view. The user can neither edit nor delete the web part.

Design Mode

The Design mode allows the user to even delete the web part along with the Minimize and Restore options. The user can also move custom web parts into any of the web part zones.

For more information on adding a web part to the zones, refer to Adding Web Part to the Web Part Zone.

To view the Design Mode

1. Go to **Enterprise Console** > **Settings** > **Web Part Settings** > **Page 1**.
2. Expand the **Web Parts Settings** bar.
3. In the **Web Part Display Mode** drop-down list, select the Design mode.
   The Design mode is displayed on the screen.

You can see that in the design mode the Web part Zones are displayed as Web Part Zone Header, Web Part Zone Left, and Web Part Zone Right. The user can also use the minimize zone option and also delete the zone option. On using the minimize option, the user can click the Restore option in the same drop-down list to restore the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

web part zone.

Edit Mode

In addition to the Minimize/Restore and Delete options, the Edit mode allows the user to edit the properties of web part. In this mode the user can open the Editor Zone pane and select the required properties and customize the web parts according to their preferences. Even in this mode, the user can move the web parts across different zones and even add new web parts to any of the zones.

To view the Edit Mode

1. Go to **Enterprise Console** > **Settings** > **Web Part Settings** > **Page 1**.
2. Expand the Webparts Settings bar.
3. In the Web Part Display Mode drop-down list, select **Edit** mode.

   Now you can even see the Edit Web Part drop-down field with all the options – Minimize, Delete, and Edit.
4. Click the **Edit** option in the drop-down list, the Editor zone pane opens. Here, the user can select any of the properties to change the nature of the web part.

Web Part Enabled Home Pages

In the Enterprise Console, users can see four web parts - Inbox, BAM, Calendar, and Queue. In addition to these, AVEVA Work Tasks also provides three default home pages to work with.

Home page is like any other web page, a web part enabled page, to which AVEVA Work Tasks has provided multiple webpartzones, where the user can place web parts. A home page can also be used as the landing page based on the security group rights.

By default, AVEVA Work Tasks provides three home page templates. Based on the security group, the home page will be set as the landing page for the users who belong to that security group. Also, the users can change the personalization settings of the home pages based on security rights.

See Web Part Rights for more information.

Let us understand the Home Pages using a scenario. In a typical IT organization, we can find three levels of employees - Senior Developers, Division Head, and CEO. AVEVA Work Tasks home pages are assigned to each of these levels, based on the four default security groups. The Senior Developer is assigned to the Reader security group; Division Head is assigned to the Contributor group; and CEO is assigned to the Administrator group. However, the fourth security group, Limited Access, is not attached to any home page. Users assigned to this group can only view the Inbox.

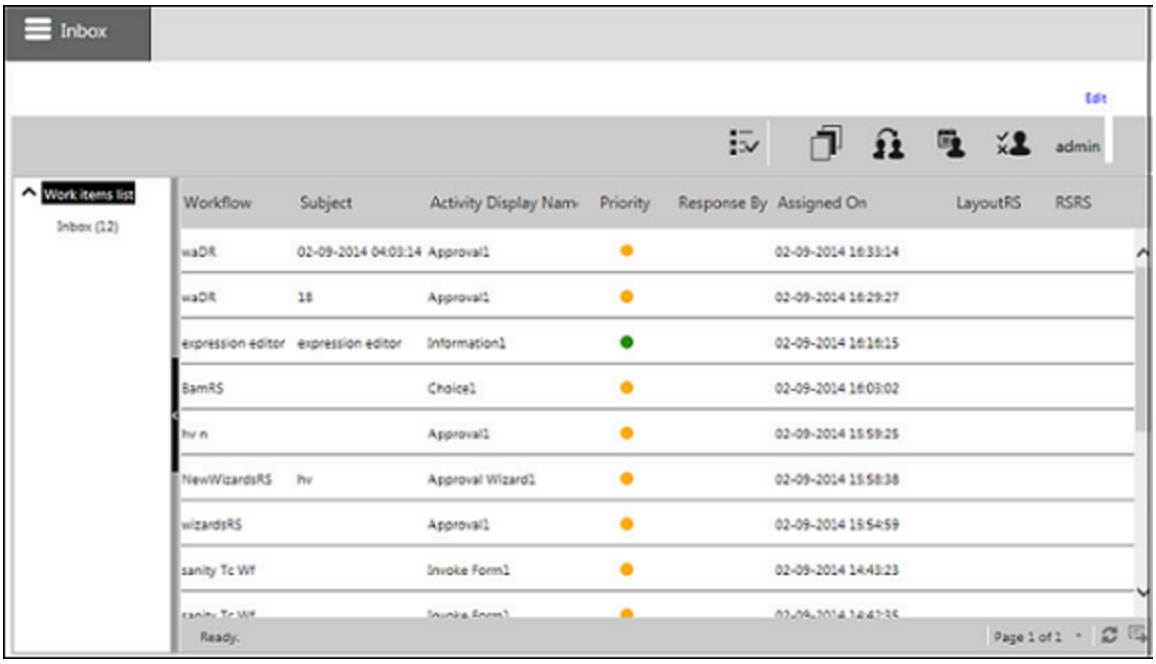
Even the default web parts are placed in these home pages according to the security group the user belongs to. However, the Inbox or work item web part is common for all security groups and will be a user-specific Inbox.

Inbox

Inbox

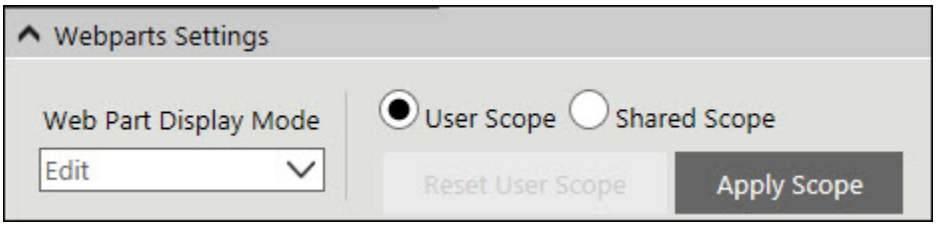
The Inbox web part page provides enhanced usability features. Workflow actors can view and respond to work items assigned in a user-interface that is similar to Microsoft Office Outlook. Work items are delivered to the actor's Inbox with a detailed view available on selection of the item. The name of the logged in user is displayed at the top right corner of the window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

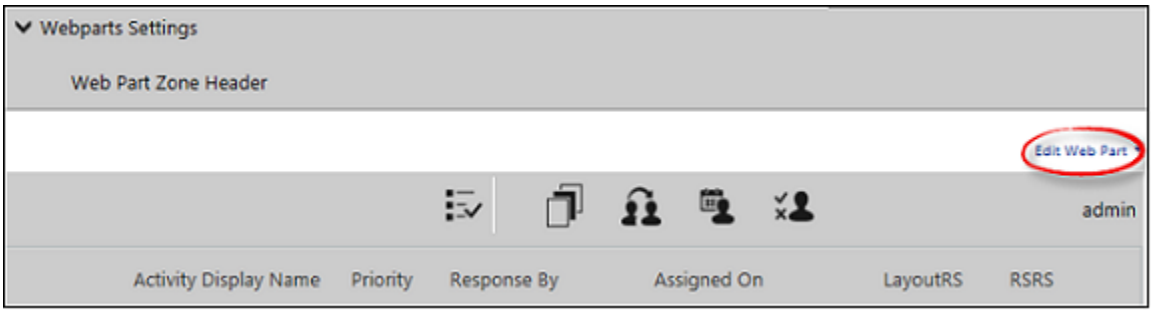


Based on the security provisions given, you can change the personalization settings.

To change the settings, expand **Webparts Settings** tab. Set the User Scope or Shared Scope in this tab. You can also reset the user scope.

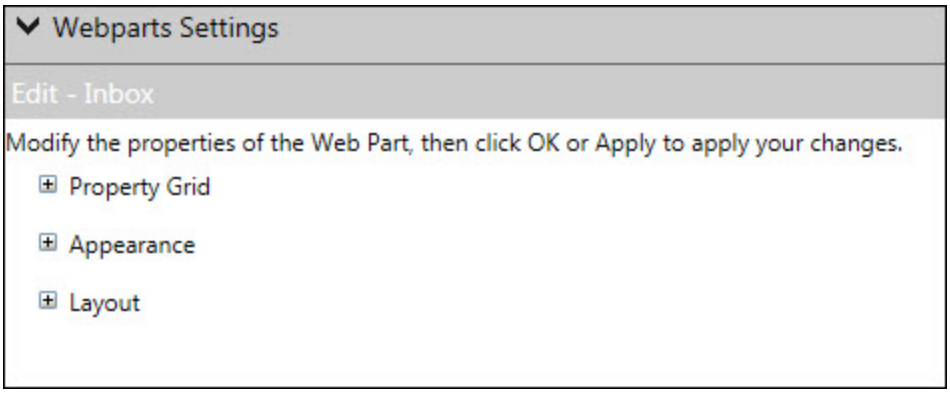


You can edit the web part from this page. Click **Edit Web Part**, and then select **Edit** to edit the web part



The editor zone appears. Select any of the properties under the Property Grid, Appearance, or Layout properties to change the web parts.

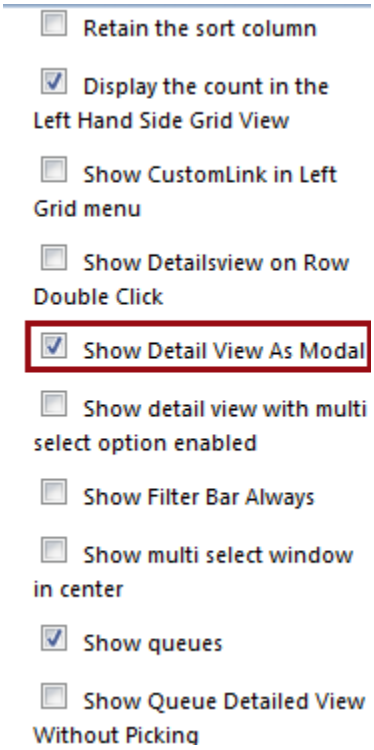
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



For more information, see Out-of-the-Box Web Parts page.

Modal or Modeless Window Property

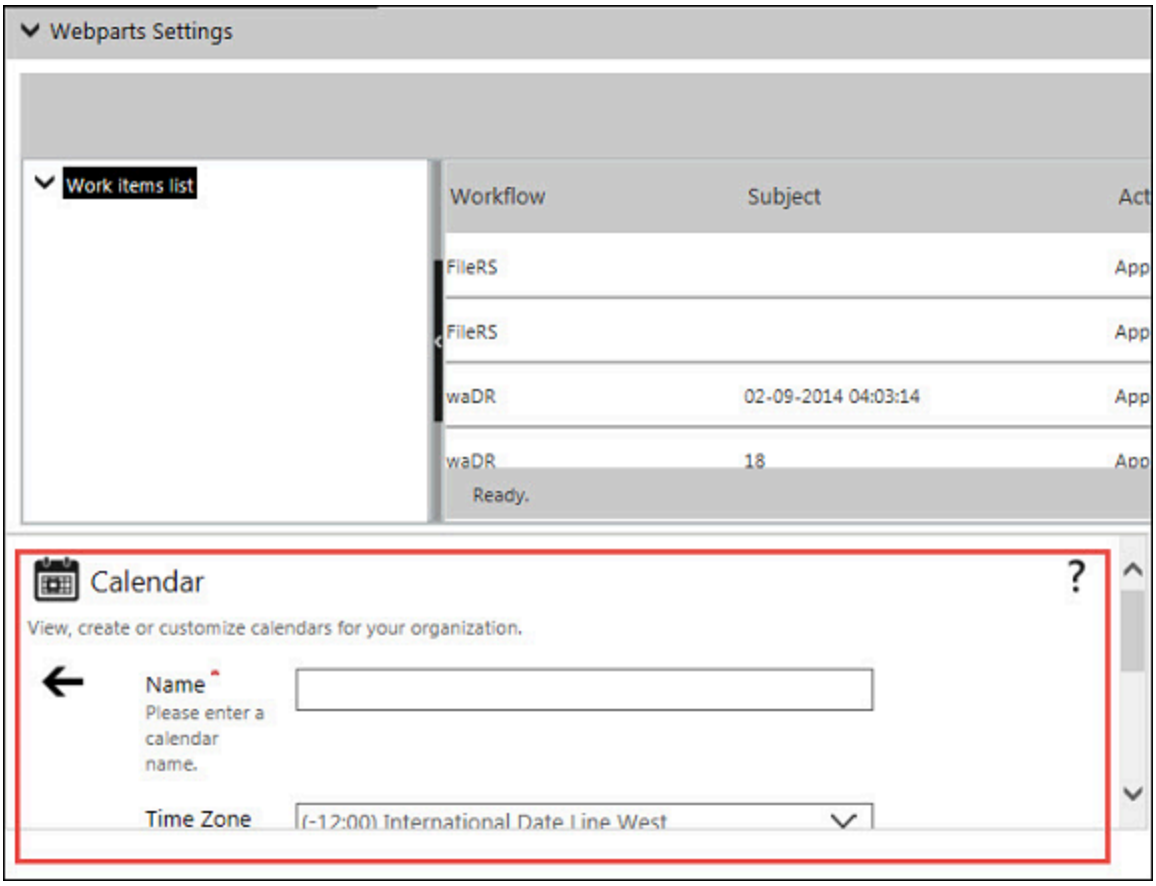
When you click a work item to edit, you can choose the editor to appear as a modal window or a modeless window. Select the Show Detail View As Modal option in the editor as shown in the following image:



Calendar

The AVEVA Work Tasks Calendar interface enables the creation of hierarchical organization and resource calendars. The Calendars are used to determine availability of resources, calculate time outs and perform other scheduling functions.

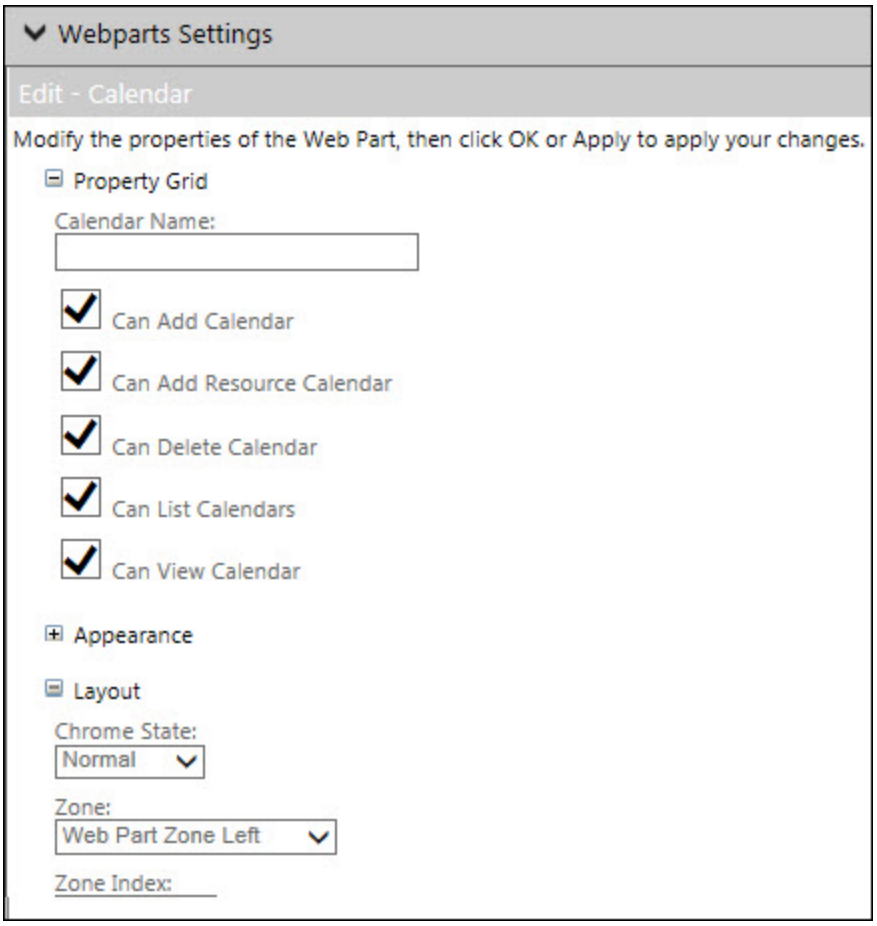
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The user can add a new calendar or set the resource calendar.

The personalization settings for the Calendar are the same as that of the Inbox we part page, but the properties in the editor zone differs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



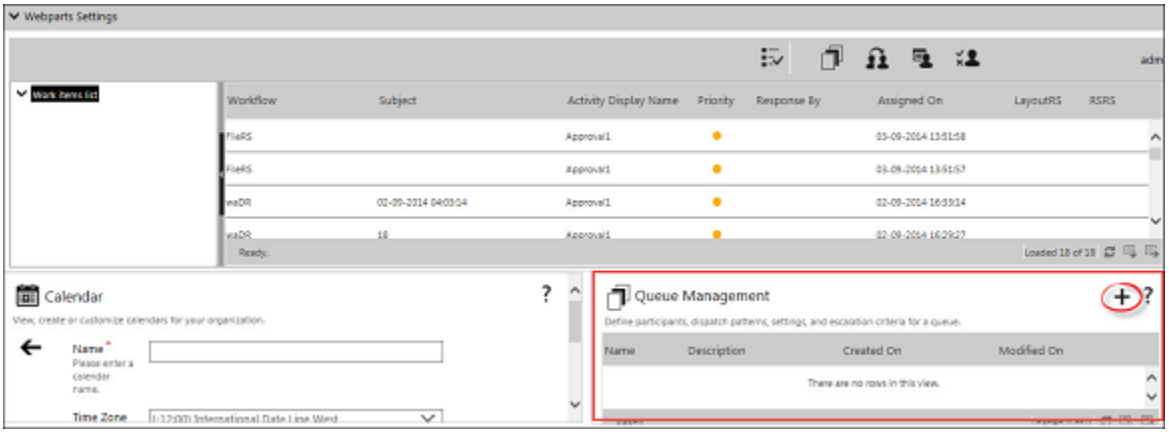
The user can edit any of the properties given here and change the web part.

For more information, see Out-of-the-Box Web Parts page.

Queue Management

The Queue Management enables you to create and manage queues for allocating workflow activities. This is done using the Queue control.

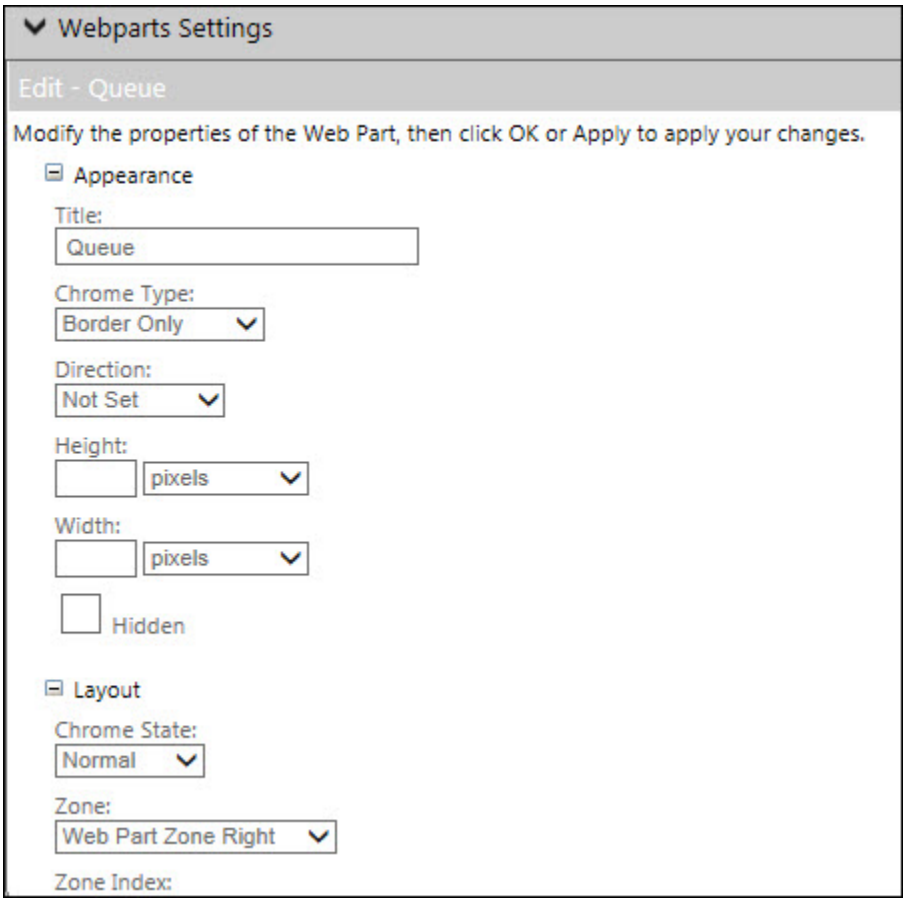
Queue control consists of only Appearance and Layout properties. It does not comprise of any property grid.



The users can add a new queue by clicking the **New Queue** button.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The personalization scope for **Queue Management** remains the same as the Inbox and Calendar Web Part pages but the editing properties differ.

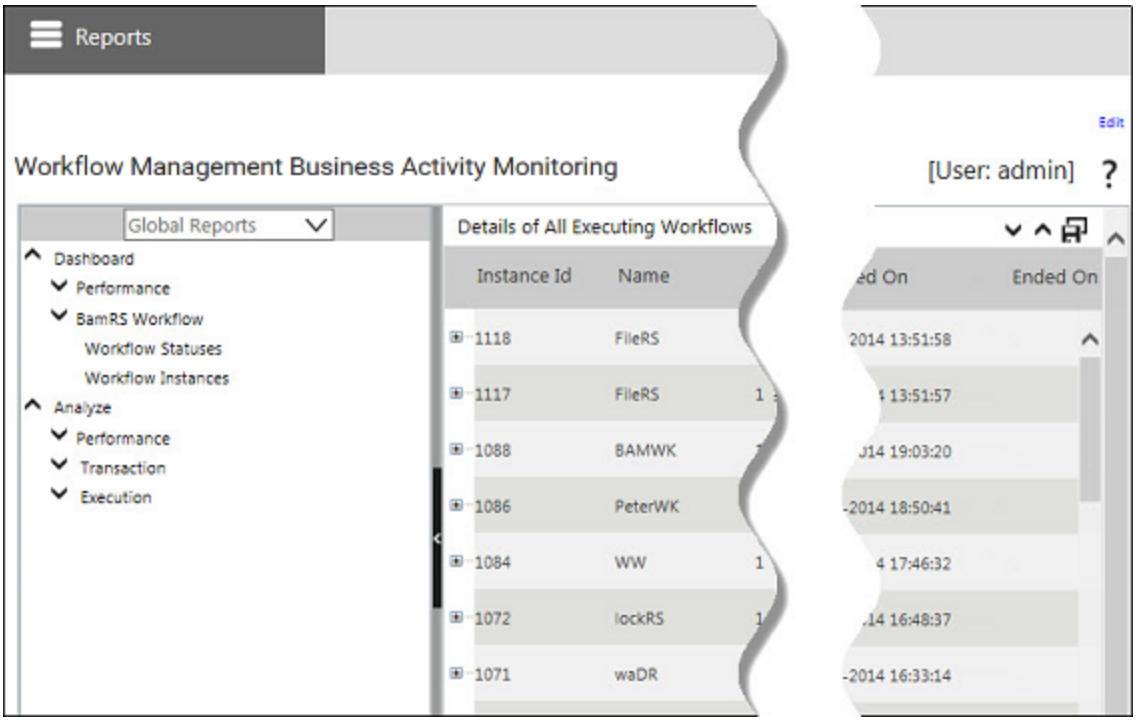


For more information, see Out-of-the-Box Web Parts page.

Business Activity Monitoring

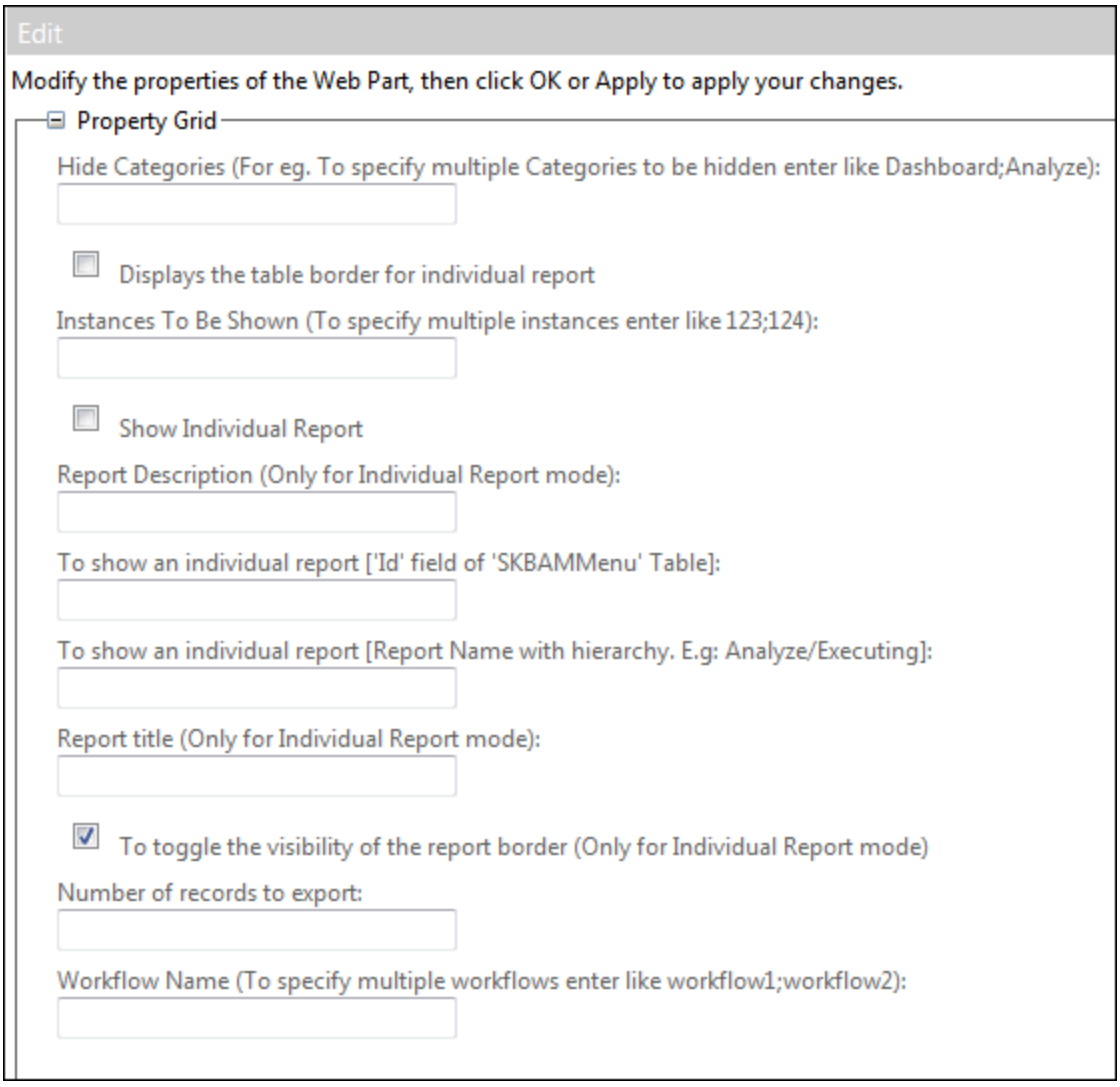
The **Business Activity Monitoring** Dashboard is a key AVEVA Work Tasks Server Control. You can use the dashboard to monitor the execution of a workflow and provides an array of real-time reports. This report includes status and performance of workflows.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



You can edit the way the reports are displayed in the BAM by using the editor zone.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



For more information on editing the BAM properties, see Out-of-the-Box Web Parts page.

Page 1

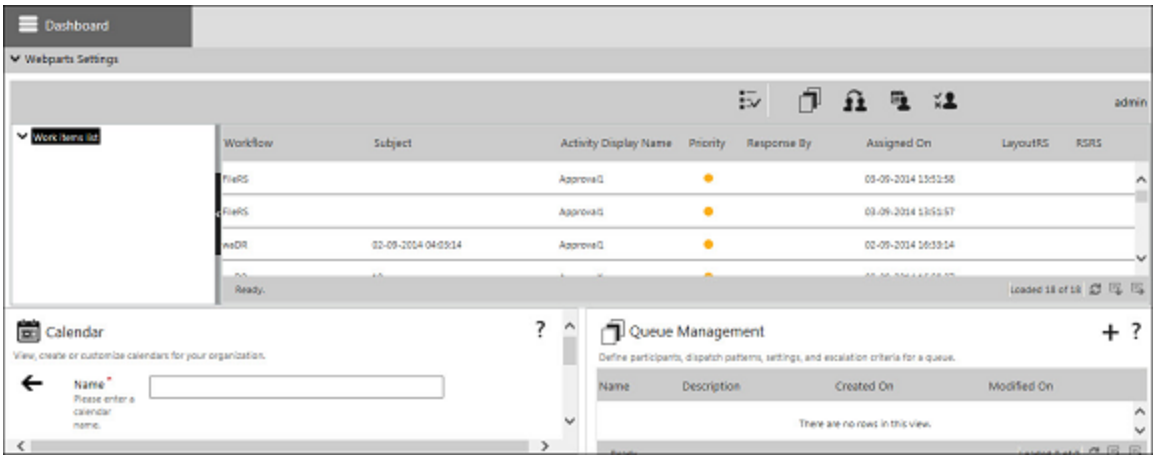
This Home Page is basically designed for the users who for example play the role of a Senior Developer. Home Page 1 has three web part zones.

AVEVA Work Tasks provides default web parts located in the above web part zones for Home Page 1 as shown below:

- Web Part Zone Header
  In this zone, the Work Item List web part is added.
- Web Part Zone Left
  In this zone, the Calendar web part is added.
- Web Part Zone Right

In this zone, the **Queue** web part is added.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

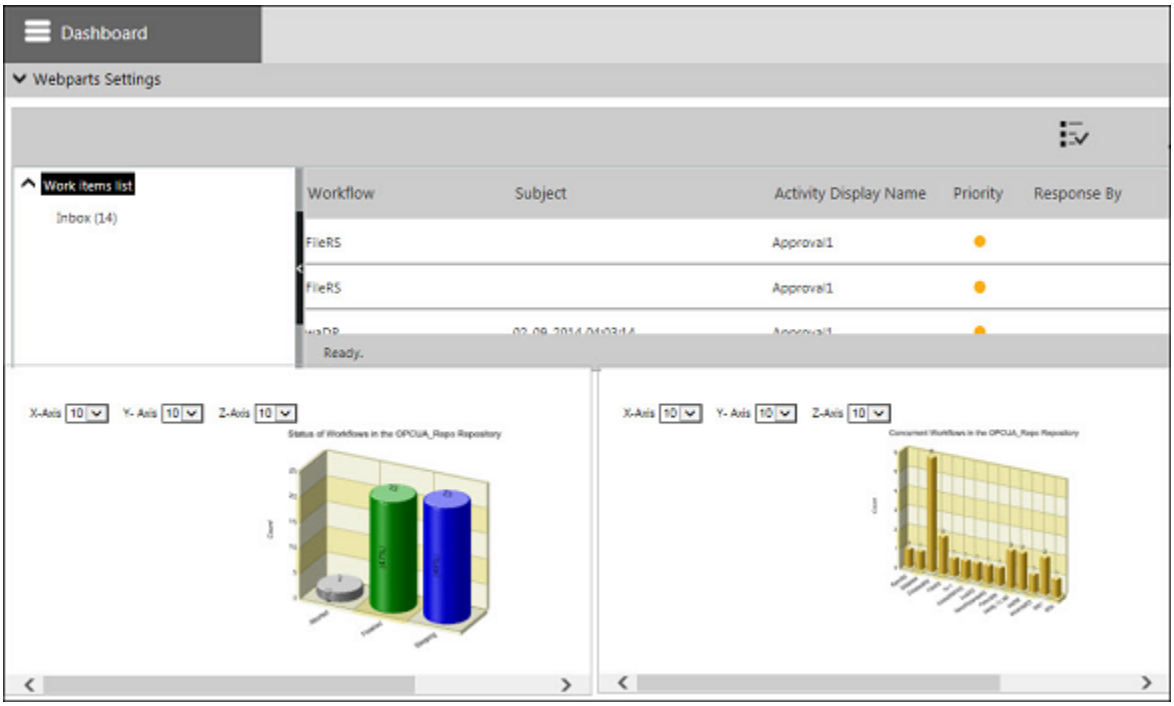


Users can add additional controls to the same zone, provided they have the required security rights.

Home Page 1 is the landing page for the users who belong to the Reader security group. For more information on Security rights, refer to Web Part Rights.

Page 2

This Home Page is basically designed for users who for example play the role of a Division head. Hence, this home page is provided with three default web parts - Inbox and Business Activity Monitoring reports. Home Page 2 has three web part zones.



AVEVA Work Tasks provides default web parts located in the above web part zones for Home Page 2 as show below:

- Web Part Zone Header - In this zone, the Work Item List web part is added.
- Web Part Zone Left - In this zone, the BAM Workflow Statuses web part is added.
- Web Part Zone Right - In this zone, the BAM Workflow Instances web part is added.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Users can add additional controls to the same zone, provided they have the required security rights.

Home Page 2 is the landing page for the users who belong to the Contributor security group. For more information on Security rights, refer to Web Part Rights.

Page 3

When you click **Web Part Pages > Page 3,** you will always be redirected to **Page 2**.

Create Custom Page

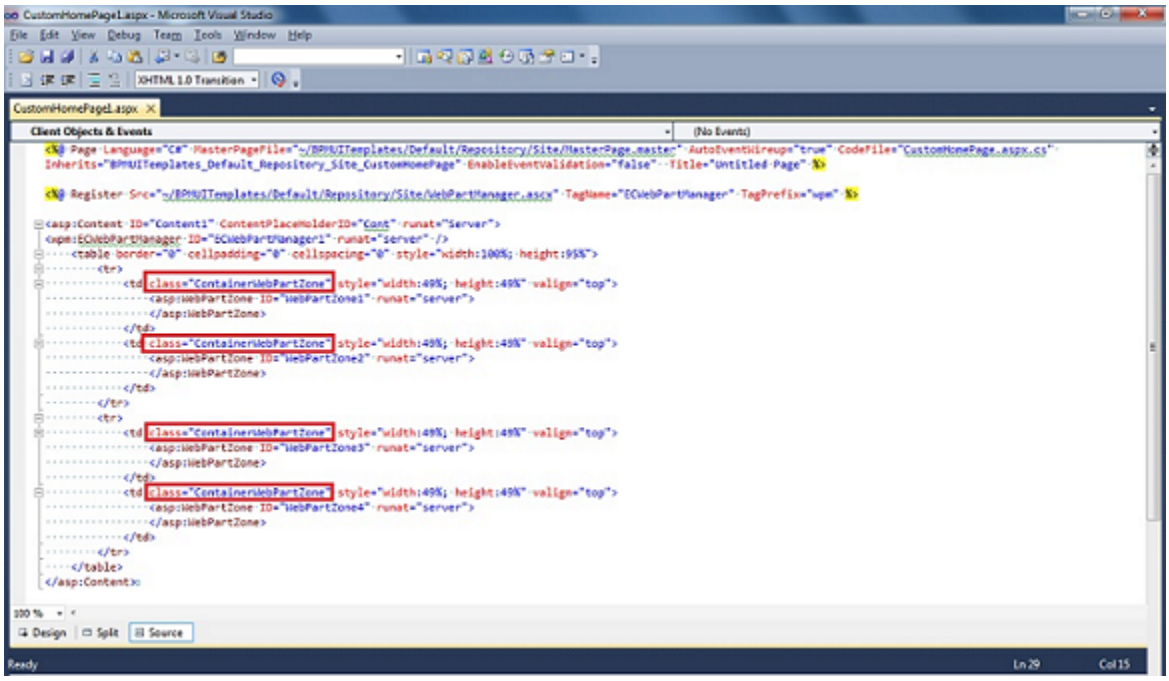
Custom home pages can be created according to the organizational requirements and needs. The section below gives details on how to create Custom Home Page.

To create a new custom home page

1. Right click on the folder in solution explorer, where you want to create a new custom home page, or click on **File** > **New** > **File** in Visual Studio.
2. Specify the page name, for example CustomHomePage.aspx, and click **Add**.
3. Include the following Register directive for the WebPartManager.

   <%@ Register Src="~/BPMUITemplates/Default/Repository/Site/WebPartManager.ascx" TagName="ECWebPartManager" TagPrefix="wpm" %>
4. Provide the html for the page layout and place the web part zones in an appropriate order.
5. Apply the styles and specify names and labels for them.

   Create a table to define the web part zones in an appropriate order. Add the class "ContainerWebPartZone" to the container of the web part zone, refer image below. This class will ensure that the height and width of the web part is adjusted to the height and width of its web zone .



In the above image <td> is the container of the web part zone and the class within the container is highlighted.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. Specify the below property to the Page directive:

   EnableEventValidation="false"

7. The ECWebPartManager user control should be the first control in the page:

   <wpm:ECWebPartManager ID="ECWebPartManager1" runat="server" />

8. Create a navigation menu for the new home page
   a. Navigate to **Repository Settings** > **Advanced Settings** > **Manage Navigation List**.
   b. Click on **Navigation** and specify the details.
9. Log out and Login, as the Enterprise Console menu is cached.
10. Drill down the menu and click on it.
11. A blank page with only the Webparts Settings option will be displayed.

    The page is blank, as only the web part zone(place holders) have been added and not the web parts. Web Parts have to be added through the Webparts Settings option, the following steps will give you details on how to add web parts:
12. Expand Webparts Settings at the top left corner, to view the toolbar.
13. Select the Web Part, which you want to add to the page from the Select Web Part drop down.
14. Select the Web Part Zone, to which you want to add the selected web part from the Select Zone drop down. Click **Add Web Part** button.

The web part will be added to the selected web part zone.

Similarly you can add other web parts to the other web part zones.

Setting up the custom home page as the landing page (Dashboard) for the logged-in user:

We need to make an entry into the SKEWebPartTemplates and SKEWebPartLayouts tables to reflect the custom home page in 'Default Home Page' right.

```
insert into SKEWebPartTemplates values(NEWID(), '<ApplicationName>', '<HomePage Name>',
'<HomePage Path>', 'User')
insert into SKEWebPartLayouts values(NEWID(), '<TemplateId>', '<HomePage Name>', null,
 '<ApplicationName>', GETUTCDATE())
```

Example:

```
insert into SKEWebPartTemplates values(NEWID(), 'Bata7Repo', 'Custom home page',
'~/BPMUITemplates/Default/Repository/Site/CustomHomePage.aspx', 'User')
insert into SKEWebPartLayouts values(NEWID(), '980D6D0E-137C-46AF-8417-81BD7F1D52A7',
'Custom home page', null,  'Bata7Repo', GETUTCDATE())
```

After that we can edit the security group and set the custom home page as the Default Home page. Then when the user logs-in Enterprise Console, they will be landed to the Custom home page.

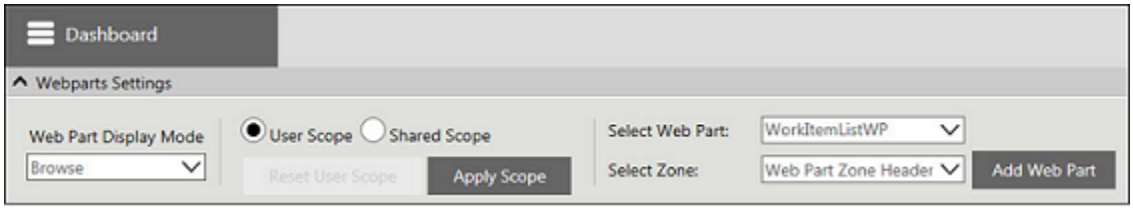
Note: To use the **Fill Form** web part in the custom ASPX page include the following script in that page.

```
<script>
function open_win(path)
{
OpenWindow(path);
}
</script>
```

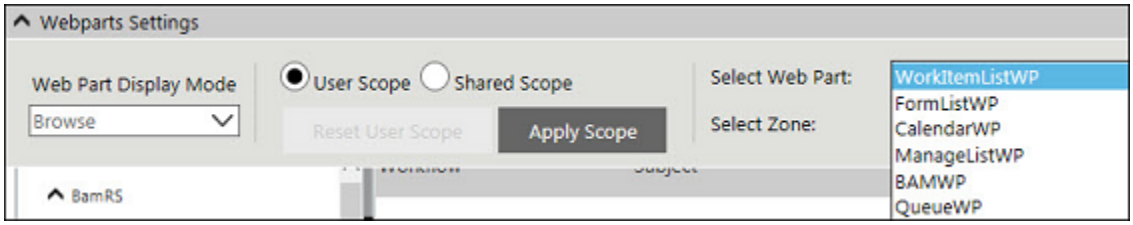
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Add a Web Part to the Web Part Zone

1. Go to **Enterprise Console** > **Settings** > **Web Part Settings** > **Page 1**.
2. Expand Webparts Settings.

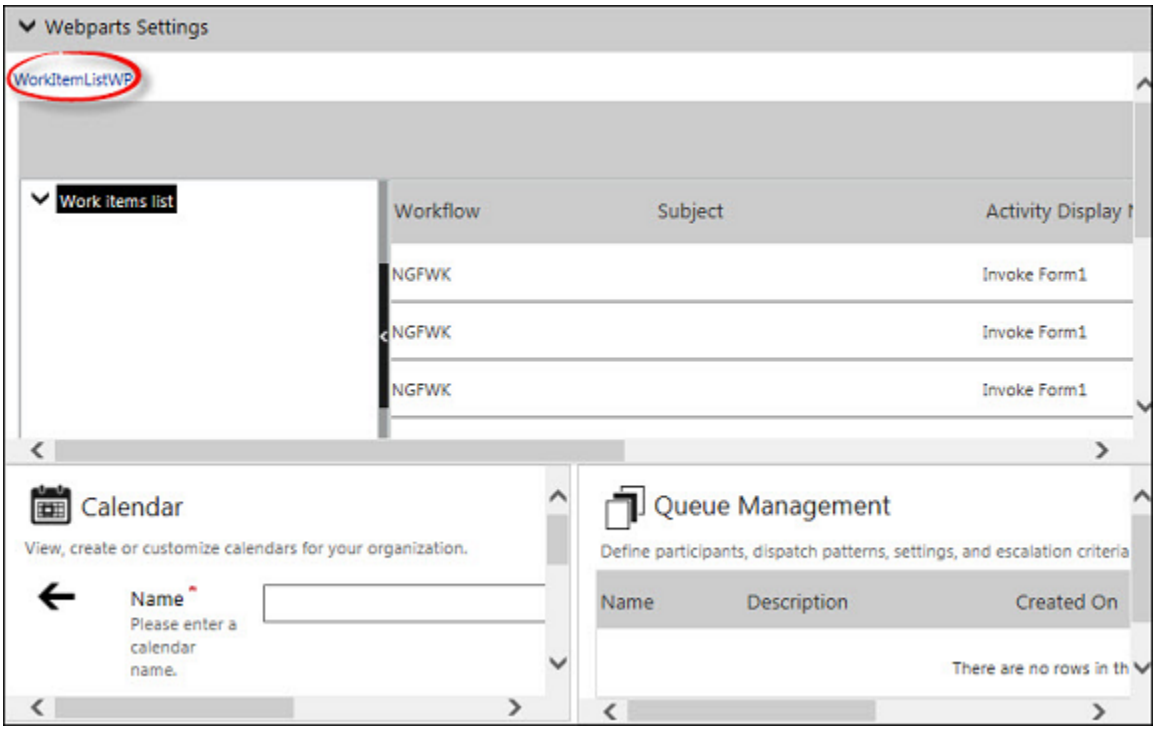


3. Select **WorkItemListWP** from the **Select Web Part** drop-down list.



4. You can see that all the default web parts and the custom web parts are listed in this drop-down list.
5. Select the **Web Part Zone Header** for the web part from the **Select Zone** drop-down list. Since you have selected Home Page 1, you can add the web part in any one of the three default zones - Header, Left, and Right.
6. Click **Add Web Part**.

After adding the web part to the selected zone, the home page will look like the image displayed below.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Delete Web Part

Users can delete a web part by going to the design mode or edit mode in the Web Part Settings ribbon bar.

To delete a web part

1. Open a home page from which you want to delete a web part..
2. Go to the **Design** or **Edit** mode in the **Web Part Settings** ribbon bar.
3. Select the **Delete** option in the Edit Web Part drop-down list of the web part you want to delete.
4. A message from the web page appears asking you if you want to delete this web part permanently.
   You can see that the **My Inbox Web Part** is deleted from the header zone and only the Work Item List web part is retained.

Edit a Web Part

Users can edit a web part by accessing the editor zone pane. To enable editing, you have to make sure that the web part is in the Edit mode.

To edit a web part

1. Go to the Edit mode in the Web part Settings ribbon bar.
2. Select the **Edit** option in the drop-down list of the web part you want to edit. The editor zone pane opens.
3. In the **Appearance** section of editor zone pane, select the **Title Only** property in the **Chrome Type** drop-down list of the Appearance properties. By selecting this property, only the title of the web part will be displayed and the border will be hidden.
4. Click **Apply** and **OK**.
   You can see that the borders of the Calendar web part are hidden and only the title is displayed.

Minimize and Restore Web Parts

Users can minimize or restore the web part. On using the minimize option, the user can click the Restore option in the same drop-down list to restore the web part zone.

To minimize the web part

1. Open a home page in any of the modes.
2. Click the **Minimize** option given in a drop-down list at the right - end of the web part.

To restore the minimized web parts

- Click the **Restore** option given in a drop-down list at the right - end of the web part.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Web Part Rights

AVEVA Work Tasks provides a set of web part rights to enable you to use the web parts more effectively.

Security Rights for Web Parts

Name of Web Part Right	Description
Web Parts Rights	It is the parent right for grouping the below rights.
Add Web Parts	Authorizes the user to add the web parts to the home page.
Delete Web Parts	Authorizes the user to delete the web parts properties of the home page.
Edit BAM Personalization Scope	Authorizes the user to change the personalization scope for the BAM control page.
Edit Calendar Personalization Scope	Authorizes the user to change the personalization scope for the Calendar control page.
Edit Home Page Personalization Scope	Authorizes the user to change the personalization scope for the logged-in user.
Edit Queue Personalization Scope	Authorizes the user to change the personalization scope for the Queue control page.
Edit Web Parts	Authorizes the user to edit the web parts properties of the home page.
Edit Work Item Personalization Scope	Authorizes the user to change the personalization scope for the Work items list control page.
Default Home Page	This right gives the provision to specify the default home page for the security group.
Personalize BAM Control	Authorizes the user to personalize the BAM control.
Personalize Calendar Control	Authorizes the user to personalize the Calendar control.
Personalize Queue Control	Authorizes the user to personalize the Queue control
Personalize Work Item Control	Authorizes the user to personalize the Work items

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Name of Web Part Right	Description
	list control.
Reset User Personalization Scope	Authorizes the user to reset the personalization data for “user scope”.
Show Home Page Web Part Settings	This right authorizes the user to see the home page web parts setting, so the user can change the settings based on the individual web parts setting rights.
Show Ribbon Bar	Authorizes the user to make visible/invisible the ribbon bar of the list control web part.

This table shows the different web part rights given to the security groups:

Right Code	Administrator	Contributor	Reader	Limited Access
WebPartsRights	Yes	Yes	Yes	No
HomePage	Yes	Yes	No	No
ShowHomePageWebPartSettings	Yes	Yes	No	No
EditHomePageScope	Yes	Yes	No	No
ResetUserScope	Yes	No	No	No
AddWebParts	Yes	No	No	No
EditWebParts	Yes	Yes	No	No
DeleteWebParts	Yes	No	No	No
PersonalizeWorkitemControl	Yes	No	No	No
EditWorkitemScope	Yes	Yes	No	No
PersonalizeCalendarControl	Yes	No	No	No
EditCalendarScope	Yes	Yes	No	No
PersonalizeQueueControl	Yes	No	No	No
EditQueueScope	Yes	Yes	No	No

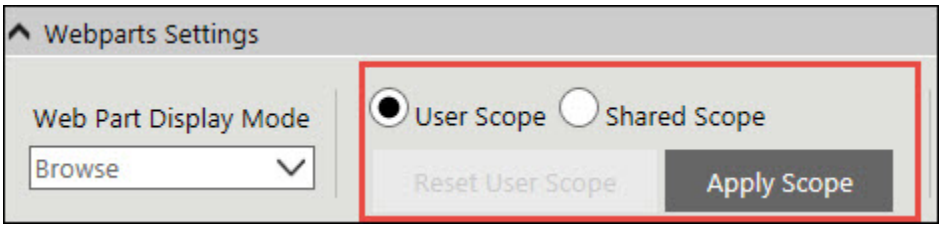
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Right Code	Administrator	Contributor	Reader	Limited Access
PersonalizeBAMControl	Yes	No	No	No
EditBAMScope	Yes	Yes	No	No
ShowRibbonBar	Yes	Yes	No	No

Note: If security flag is disabled, then the above given rights for the web parts will not be effective.

Personalize Scope in Web Parts

In the AVEVA Work Tasks Web Parts, users can use two scopes – User scope and Shared scope.



By default, all the home pages are in the User scope. When the user selects the Shared scope, by default the home page will be empty. On clicking the Reset User Scope option, the existing controls will be deleted and the default controls will be added.

- **User:** Individual users can not personalize properties scoped as Shared. They can personalize User-scoped properties, and their values for these properties take precedence over the property values assigned when the page was in Share scope. If the user-specific personalization data on a control is lost or reset, User-scoped properties revert to the values they had when the page was in Shared scope.
  Individual users can close a shared control for themselves, which adds it to the Page Catalog, but they cannot permanently delete it.
  Individual users can personalize both Shared and User scoped personalizable properties on the control for themselves, because the control instance is completely private.
  Individual users can also permanently delete the control.
- **Shared:** WebPart controls are shared by default. An authorized user can personalize both Shared and User scoped properties on the control for all users.
  In the case of a dynamic control (a control that is added to the page programmatically, or from a catalog of controls), an authorized user can permanently delete it for all users.
  In the case of a static control (a control that is declared in the markup of an .aspx page), it cannot be deleted, although an authorized user can close the control for all users.
- **Per-user control:** The control cannot be personalized with the page in Shared scope, because the control does not even appear on the page. The control only appears when the page is in User scope.

Recommendations in Web Parts

- Security Group Mapping in List Control Web Part should be used at List level and ListItem level.
- Show Content from Multiple Lists
  Mandatory configurations for Adding Multiple List Control Web Part in Dashboard:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Disable the properties Show Header and Show Ribbon Bar
- Use unique list view names across all the List Control Web Part
- Create list views with All items enabled
- Let us assume a scenario where the user creates similar list in multiple list control web parts placed in one home page. In such a scenario, the user may not be able to view the list views properly

- Users are recommended to refresh the grid manually after creating items or folders in a grid.
- Users can specify height and width in percentage or in pixels for all the web parts except the Queue Control web part. However, for the FormsList web part the users can specify the height and width in pixels alone.
- In Firefox Dashboard, the users can move the web parts across the zones by using the Web Editor.
- Users are recommended to use the below mentioned Web Parts only once in a home page:
  - Calendar controls
  - Queue

  Note: When multiple lists are present in the Web Parts, changes done to the lists like adding new list item or modifying/deleting an existing list item may not be reflected in the lists present in the web parts. User needs to manually refresh the grid.

Report Service

AVEVA Work Tasks Reports are hosted using the SQL Server Reporting Services. These reports are generated using the templates deployed in the reporting server. The reporting server contains several report templates from which the user selects an appropriate template and generates the required reports. The **Report Generator** activity provides an execution facility for those reports and enables the user to send the report as an Email attachment, a link, or as an embedded report using the viewer control in **AVEVA Work Tasks Form**.

The AVEVA Work Tasks, by default, does not have ArchestrA reporting server information. Hence, AVEVA Work Tasks workflow requires the ArchestrA reporting service information, to execute the ArchestrA report from the workflow.

To send the reports through AVEVA Work Tasks, use the Report Generator activity and the following AVEVA Work Tasks lists:

- **Report Generator Activity:** Use this activity to configure a Report Template and also to save the generated report in the Reports list. For more information, see *Report Generator Activity* in the *Developer Guide*.
- **Connections:** This list contains the report server information, report execution credentials, and the default path. Use this information to connect to the SSRS web service to fetch the report template information in order to execute the report. For more information, see Configuring Report Connection.
- **Templates:** This list contains report templates that runs at run time using the report server information mentioned above. For more information, see Configuring Report Templates.
- **Reports:** This list contains reports which are generated using the Report Generator activity. You can manually import the previously created reports. For more information, see Configuring Reports.

  Note: An error message *ASP.NET session has expired or could not be found* is displayed by SQL Server Reporting services when a Form has Report Viewer control and is open for a long time.
  For Report Server, the default session timeout is 600 seconds (10 minutes). You can increase this timeout to any value and based on these settings, session timeout will occur.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Sample Business Case Study

Using AVEVA Work Tasks, we can configure a production report on completion of different stages of production and also send the report as an email attachment based on which the 'officer in charge' approves the finalization of the product. This enhances the discrete manufacturing line of AVEVA MES.

Perrot's manufacturing unit is involved in production of Perrot airplanes (Boeing 787 Dream Liner). AVEVA MES is configured to represent a discrete manufacturing line, where completing every stage is monitored and recorded. Due approval is required for the finalization of airplanes and hence, AVEVA Work Tasks can be used for managing this whole process. AVEVA Work Tasks is capable of the following functions:

- Generate a production report which resides on the AVEVA Information Server as an AVEVA Workflow MES report.
- The report uses AVEVA's MES Database and AVEVA Alarm Database to connect production time, production line with the Aircraft ID and other context parameters which can go upto 256.
- The generated report is sent as an email attachment to different departments, including Quality Management, Aircraft control, production planning, shipment and customer management.
- Copies of the same email can be sent to the CEO and CFO of Perrot as the completion of an aircraft is an event to be celebrated.
- Officer-in-charge of each department reviews the report received and approves the finalization.
- On obtaining the approvals the Customer Management arranges for FAT (Acceptance Test).
- On obtaining FAT, the procurement departments issues the invoice

Once the invoice is issued, AVEVA Work Tasks scenario ends. AVEVA Work Tasks facilitates the generation of a detailed production report and intimate the approver, officer's in charge or heads of different departments and the company, obtain required approval and manage the process till the invoice is issues by the procurement department.

Connections

This section provides information on how to create, configure, modify, or delete a Report Connection.

Configure Report Connection

To configure a Report Connection, do the following steps:

1. Log on to the **Enterprise Console**.
2. In the Enterprise Console, click menu icon, click **Settings**, click **Reporting Service**, and click **Report Connections**.

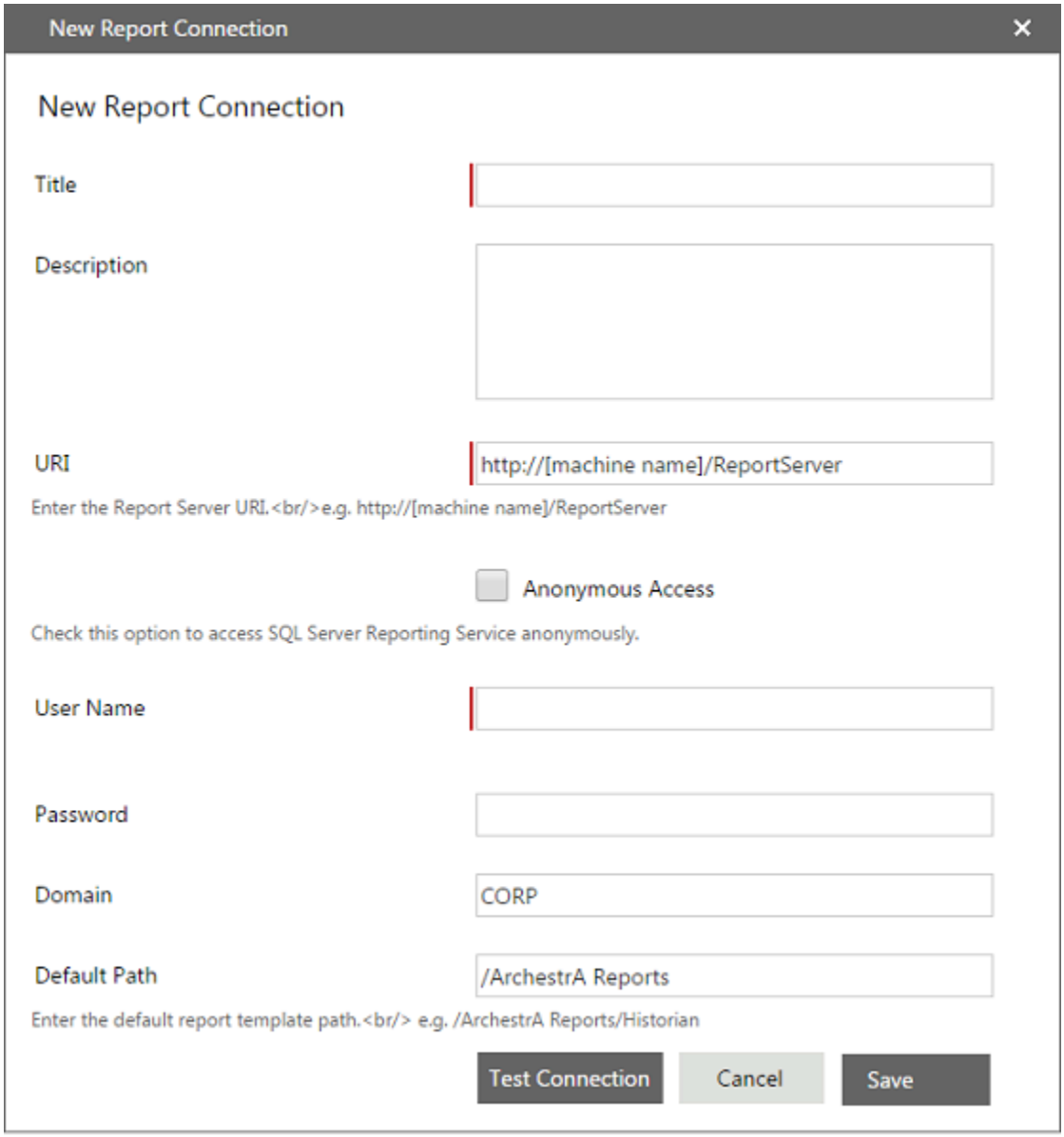
   The **Report Connections** page appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Create a New Report Connection Item

To create a new report connection item

1. On the action bar, click **New**. The **New Report Connection** wizard appears.



New Report Connections - Properties:

- **Title:** Title for a new Report Connection. The text entered in this text box cannot be edited as this is a unique text box.

  Note: Title cannot contain special characters such as # ~ & + \ " ' < > | @ ! $ % * ( ) = / { } ?

- **Description**: Description of the new Report Connection.

  Note: Title cannot contain special characters such as # ~ & + \ " ' < > | @ ! $ % * ( ) = / { } ?

- **URI**: Report Server URI. For example, http://

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Unique Resource Identifier (URI) is the URL of the web service to which the user needs to connect to execute a report.

- **Anonymous Access**: Check this option if the URI specified above is configured to be accessed anonymously. The test connection succeeds only if the URI is accessible anonymously. The user need not specify the user credentials if this option is checked.

  Note: The **Anonymous Access** option would not work for SSRS 2008, as it is not supported out of the box.

- **User Name**: User name to access the above URI. If the Anonymous Access option is not checked, then enter the user name in this text box to access the above mentioned URI.
- **Password**: Password for User Name.
- **Domain**: Name of the domain in which the above mentioned user exists.
- **Default Path**: Enter the default report template path. This information is required to locate a specific group of report templates that have been deployed in the reporting server. For example, the templates are stored in this location - /ArchestrA Reports/Historian
- **Save**: Save option is for saving the entries made.
- **Test Connection**: To test whether the report connection is valid or not.

2. Click **Test Connection**. If you have not provided valid URI and connections, then the test connection fails. If the connection is successful, then test connection successful message appears.
3. Click **Save** to store the details. By clicking the **Save** option, the connection will be tested with all the given information and then will save it. The new connection appears in the **Report Connections** list window.

Modify Report Connection Item

The **Edit** option allows you to modify an existing report connection list item.

To edit an existing report connection item

1. Select the report connection item to be modified.
2. Select **Edit** from the right-click context menu or click **Edit** from the ribbon bar.

   The **Edit Report Connection** window appears.
3. Edit the information given in all the text boxes except the Title text box. This text box is unique and non-editable.
4. Click **Test Connection**.
5. Click **Save** to store the details.

Delete Report Connection Item

The **Delete** option enables you to delete an existing report connection item.

To delete a report connection item

1. Select the report connection item to be deleted.
2. Select **Delete** from right-click context menu or click **Delete** from ribbon bar.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

3. In the pop-up window, click **Yes** to delete the selected report connection item.

   If the connection is being used by any of the **Report Template** list items, then the connection cannot be deleted.

   If the report connection is deleted successfully, then the successful deletion message appears.

Report Templates

This section provides information on how to create, configure, modify, or delete a Report Template.

Configure Report Templates

To configure Report Templates

1. Log on to the **Enterprise Console**.
2. In the Enterprise Console, click menu icon, click **Settings**, click **Reporting Service**, and then click **Report Templates**.

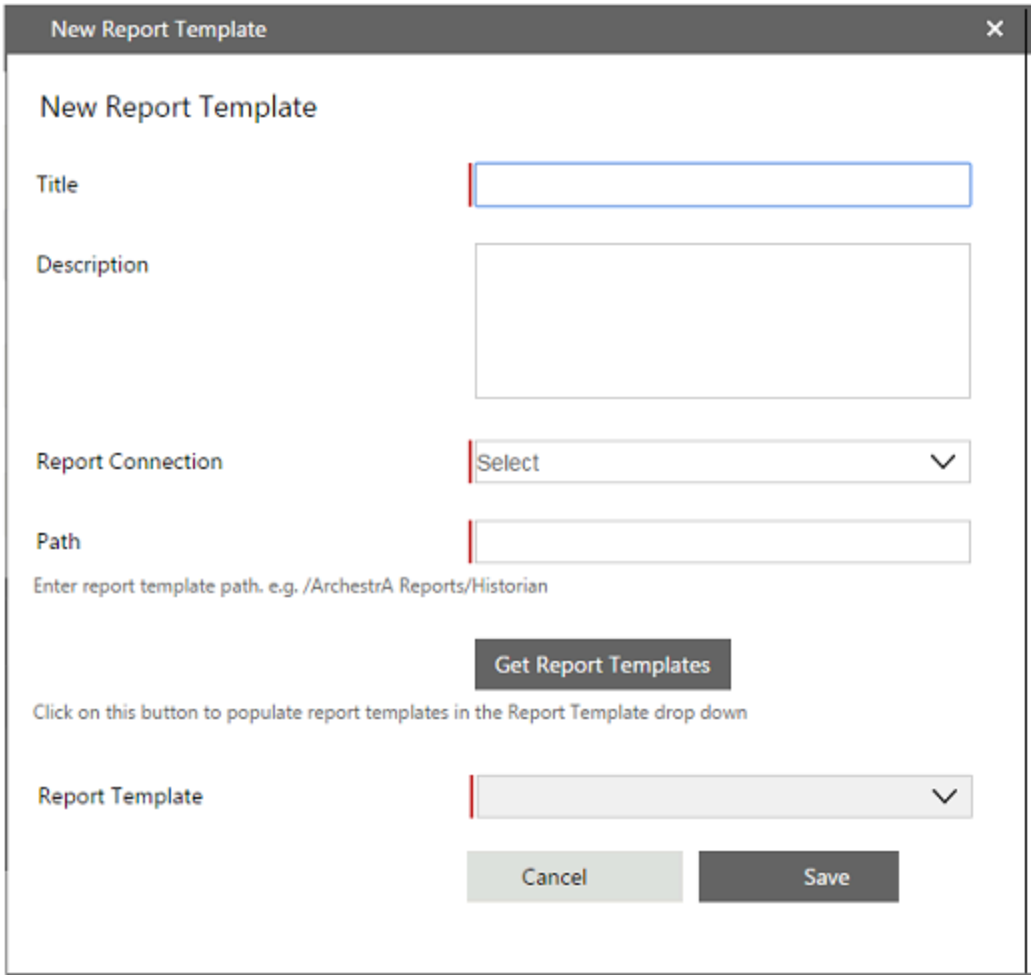
   The **Report Templates** page appears.

Create a New Report Template Item

To create a new report template item

1. On the action bar, click **New**. The **New Report Template** wizard appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



New Report Templates - Properties

- **Title**: Title for a new Report Template. The text entered in this text box cannot be edited as this is a unique text box.

  Note: Title cannot contain special characters such as # ~ & + \ " ' < > | @ ! $ % * ( ) = / { } ?

- **Description**: The description for the Report Template.

  Note: Description cannot contain special characters such as # ~ & + \ " ' < > | @ ! $ % * ( ) = / { } ?

- **Connection**: Select the required Report Connection from the drop-down list of available connections created in the Report Connection list.
- **Path**: Enter the path where all the report templates are stored. If the path has been mentioned while creating a Report Connection, then the path is populated in this text box upon selecting that Report Connection. The user can change this path and give a different path.
- **Get Report Template**: By clicking on the Get Report Template button, the available report templates will be populated in the Report Template text box.
- **Report Template**: Select the required report template.
- **Save**: This property is to save the entries made.

2. Click **Save** to store the details. If the report connection is successful, then the report template will be saved successfully. The new report template appears in the Report Template list window.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Modify a Report Template Item

The **Edit** option allows you to modify an existing report template item.

To edit an existing report template item

1. Select the report template item to be modified.
2. Select **Edit** from the right-click context menu or click **Edit** from the ribbon bar.
3. The **Edit Report Template** window appears.
4. Edit the information in all the text boxes except the **Title** text box. This text box is unique and non-editable.
5. Click **Save** to store the details.

Delete a Report Template Item

The **Delete** option enables you to delete an existing report template item.

To delete a report template item

1. Select the report template item to be deleted.
2. Select **Delete** from right-click context menu or click **Delete** from ribbon bar.
3. In the pop-up window, click **Yes** to delete the selected report template item.

Reports

This section provides information on how to create, configure, modify, organize, or delete a Report Item.

Configure Reports

To configure a Reports Connection

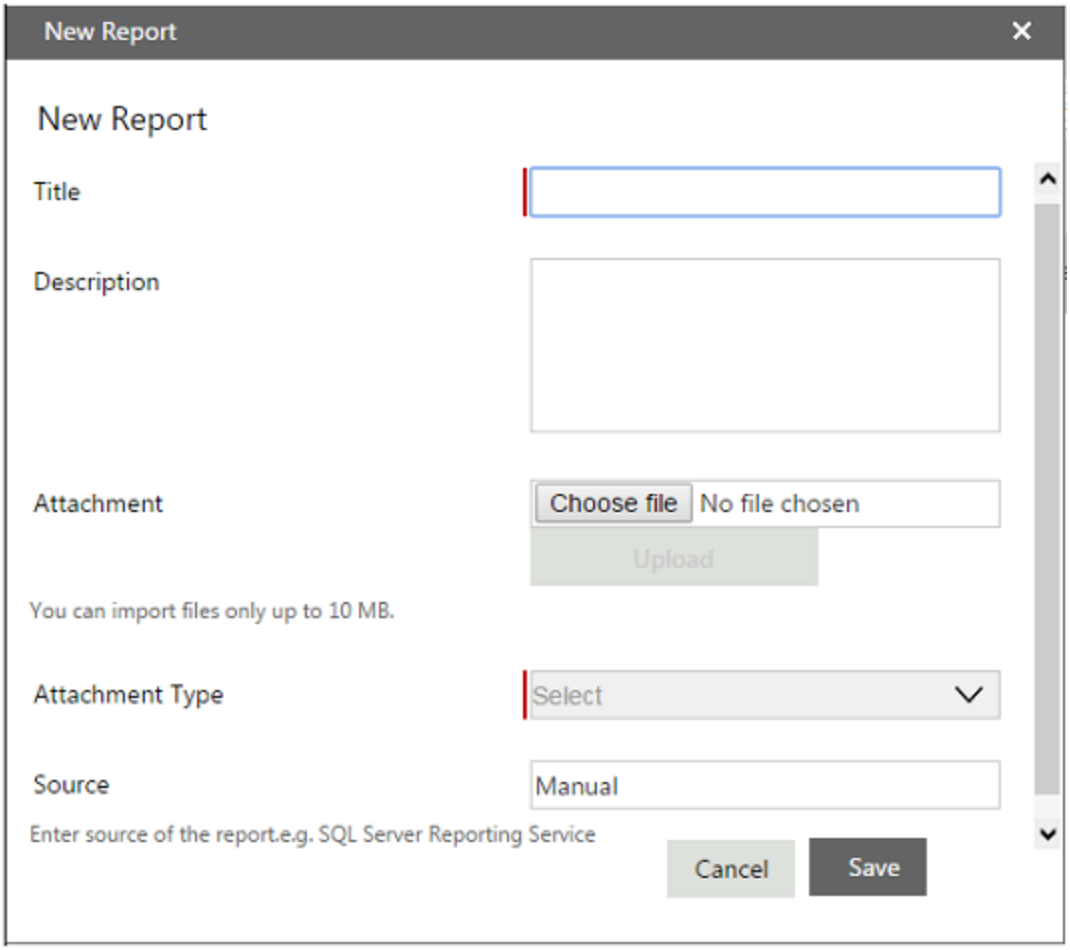
1. Log on to the **Enterprise Console**.
2. In the Enterprise Console, click menu icon, click **Settings**, click **Reporting Service**, and then click **Reports**. The **Reports** page appears.
3. To create a new report item, see Create a New Report.

Create a New Report Item

To create a new report item

1. On the Action bar, click **New**, and then click **Report**. The **New Report** wizard appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



New Report - Properties

- **Title**: Title for a new Report. The text entered in this text box cannot be edited as this is a unique text box.

  Note: Title cannot contain special characters such as # ~ & + \ " ' < > | @ ! $ % * ( ) = / { } ?

- **Description**: Description of the new Report.

  Note: Title cannot contain special characters such as # ~ & + \ " ' < > | @ ! $ % * ( ) = / { } ?

- **Attach Report**: This property allows the user to select the required report to be attached. Upon clicking the Click here to Attach link, the required report can be browsed and attached. The user can even detach the report and attach a different report in this property.
- **Attachment Type**: This property allows the user to select the type of attachment for the report to be attached. The attachment formats supported are PDF, Excel, Image, and others.
- **Source**: This property allows the user to give the source of the report to be attached. The default value in this text box is Manual. If a report is stored in the List using Report Generator activity, then the source would be "Report - Workflow Name".
- **Save**: This property is to save the entries made.

2. Click **Save**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Modify a Report Item

The **Edit** option allows you to modify an existing report item.

To edit an existing report Item

1. Select the report item to be modified.
2. Select **Edit** from the right-click context menu or click **Edit** from the ribbon bar.
3. The **Edit Report** screen appears.
4. Edit the information in all the text boxes except the **Title** text box. This text box is unique and non-editable.
5. Click **Save** to store the details.

Delete a Report Item

The **Delete** option enables you to delete an existing report item.

To delete a report item

1. Select the report item to be deleted.
2. Select **Delete** from right-click context menu or click **Delete** from ribbon bar.
3. In the pop-up window, click **Yes** to delete the selected report item.

Organize Reports

Organize a report by moving the report to a folder or by making the report a child item of another report.

To organize a report by moving the report to a folder

1. In the **Enterprise Console**, click menu icon, click **Settings**, click **Reporting Service**, and then click **Reports**. The **Reports** page appears.
2. On the action bar, click **Organize**. The **Organize Reports** dialog box appears.
3. Drag a report to a folder. The report is moved to the folder.
4. Click **Close**. The **Organize Reports** dialog box is closed.

To organize a report by making the report a child item of another report

1. In the **Enterprise Console**, click menu icon, click **Settings**, click **Reporting Service**, and then click **Reports**. The **Reports** page appears.
2. On the action bar, click **Organize**. The **Organize Reports** dialog box appears.
3. Drag a report to another report. The report is now a child item of another report.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click **Close**. The **Organize Reports** dialog box is closed.

Configure Reports for Reporting Service

Given below are the steps to configure AVEVA Work Tasks Reports for Reporting Service. This is a five-step process which comprises of:

Prerequisites

1. AVEVA Information Server should be configured.
2. The Historian and Alarm applications should also be configured before configuring Reports.

To configure Reports

1. After configuring the AVEVA Information Server, open the **Configurator** from the **Start** > **All Programs** > **AVEVA** > **Information Server**
2. Select **Archestra Reporting** in the Configurator window.
3. Create a Data Source for Historian Sample Reports and Alarm Sample Reports.
4. Click the **Configure** button to configure Reports.

To verify Reports Configuration

1. After configuring the Reporting, open browser and enter the following URLs:
   a. http://[Machine Name]/ReportServer (You can use the IP address instead of the [Machine Name])
   b. http://[Machine Name]/Reports (You can use the IP address instead of [Machine Name])

   If the Reporting is configured properly, then the following paths will be available in the SQL Server Reporting Service (SSRS): /AVEVA Workflow Report/Historian
   a. By browsing the **Home** > **AVEVA Workflow Reports** > **Historian** folder, you can see the reports deployed in this folder. /AVEVA Workflow Report/Alarm
   b. By browsing the **Home** > **AVEVA Workflow Reports** > **Alarms** folder, you can see the reports deployed in this folder. (Here the report template is named as Alarm History Report.)

To test Reporting Connectivity from the Server

Perform the same steps as given in the previous section from the server. If the host name is not accessible from this server, then you can use the Internet Protocol (IP) address instead of the machine name to connect.

To create a Connection in the Reporting Service

1. Login to Enterprise Console.
2. From the Enterprise Console pull-down menu, select **Settings** > **Reporting Services** > **Connections**.
3. Click **New** to create a new connection.
4. Enter all the mandatory information.

   Note: The format of the given path must be followed as given in this document. It should start with “/” and there should be no “/” at the end of the path.
5. Click the **Test Connection**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

6. When the connection succeeds, click **Save** to save the connection.

To create a Template

1. Login to Enterprise Console.
2. From the Enterprise Console pull-down menu, select **Settings** > **Reporting Services** > **Templates**.
3. Click **New** to create a new template.
4. Click **Save** to store the details.

User Tools

In AVEVA Work Tasks, the **User Tools** feature helps you identify the artifacts where the user needs to be replaced, when a particular user is no longer associated with the organization or the process. It also helps you identify the occurrences of the workflow where the user's direct reference is used and view and unlock list items locked by a particular user.

Importance of Replace User

To ensure the process is not affected and all tasks are completed on time, it is important to replace an user who is no longer associated with the organization or process with an alternate user. Listed below are some of the scenarios where user needs to be replaced.

- List item owned by the User to be Replaced.
- User to be Replaced is a participant of the queue and also has some work items assigned.
- User to be Replaced is as an alternate actor for any user.
- User to be Replaced is a repository administrator.
- User to be Replaced is mapped to a security group at repository, list or list item level.
- User to the Replaced is configured in the exception mail settings of the repository.

For more details, see Replace User.

Importance of View Workflow User Reference

This option of User Tools helps you search for an Actor's direct references in the published versions of the workflow.

For more details, see View User Reference.

Importance of View and Unlock Locked Items:

This option of User Tools helps you view the list items locked by the selected user and unlock it if required.

For more details, see Unlock Locked Items.

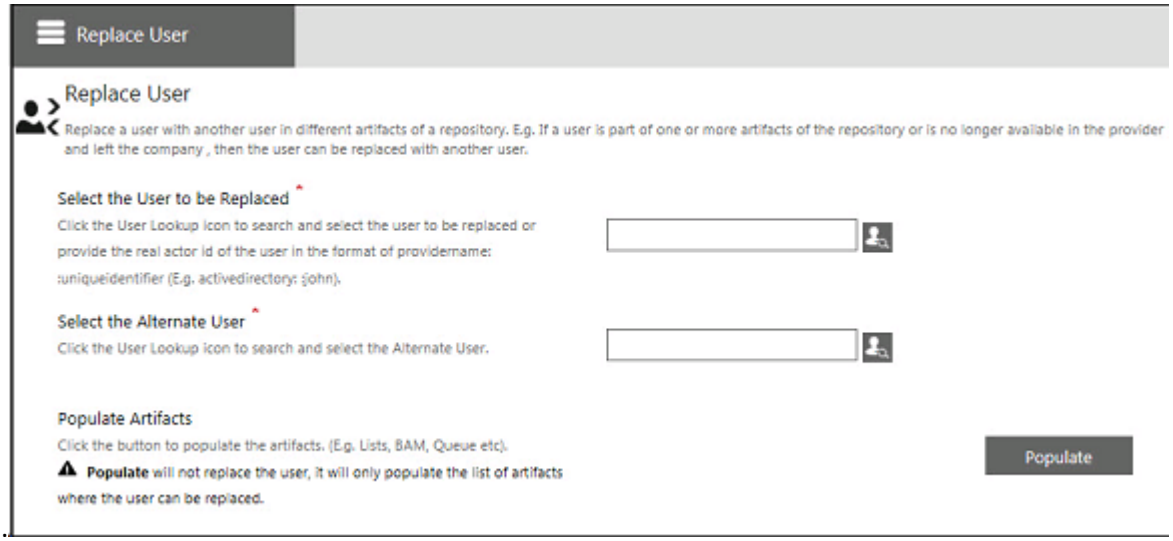
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Replace User

Replace User accessed from **Enterprise Console > Tools > User Tools** menu, enables you to replace an user (who is no longer associated with an organization or process) with an alternate user in different artifacts (Lists, Queue) of a repository.

To replace user

1. Log on to **Enterprise Console** and go to **Tools > User Tools > Replace User**. The **Replace User** page appears. You can select *an User to be Replaced and Alternate User* for different artifacts of a repository.



2. In the **Select the User to be Replaced** field, click the **User Lookup** icon. The **User Lookup** dialog box appears.
3. Search and select the user to be replaced, and then click **Update**.

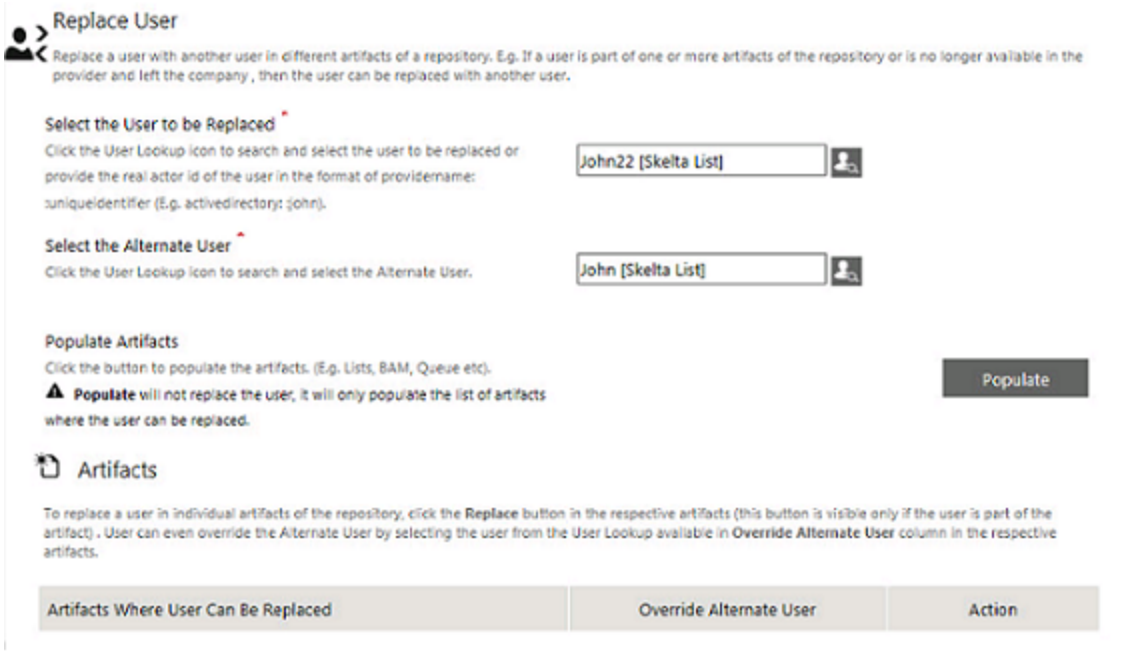
   Or

   Enter the **Real Actor Id** of the user as specified in the examples below:

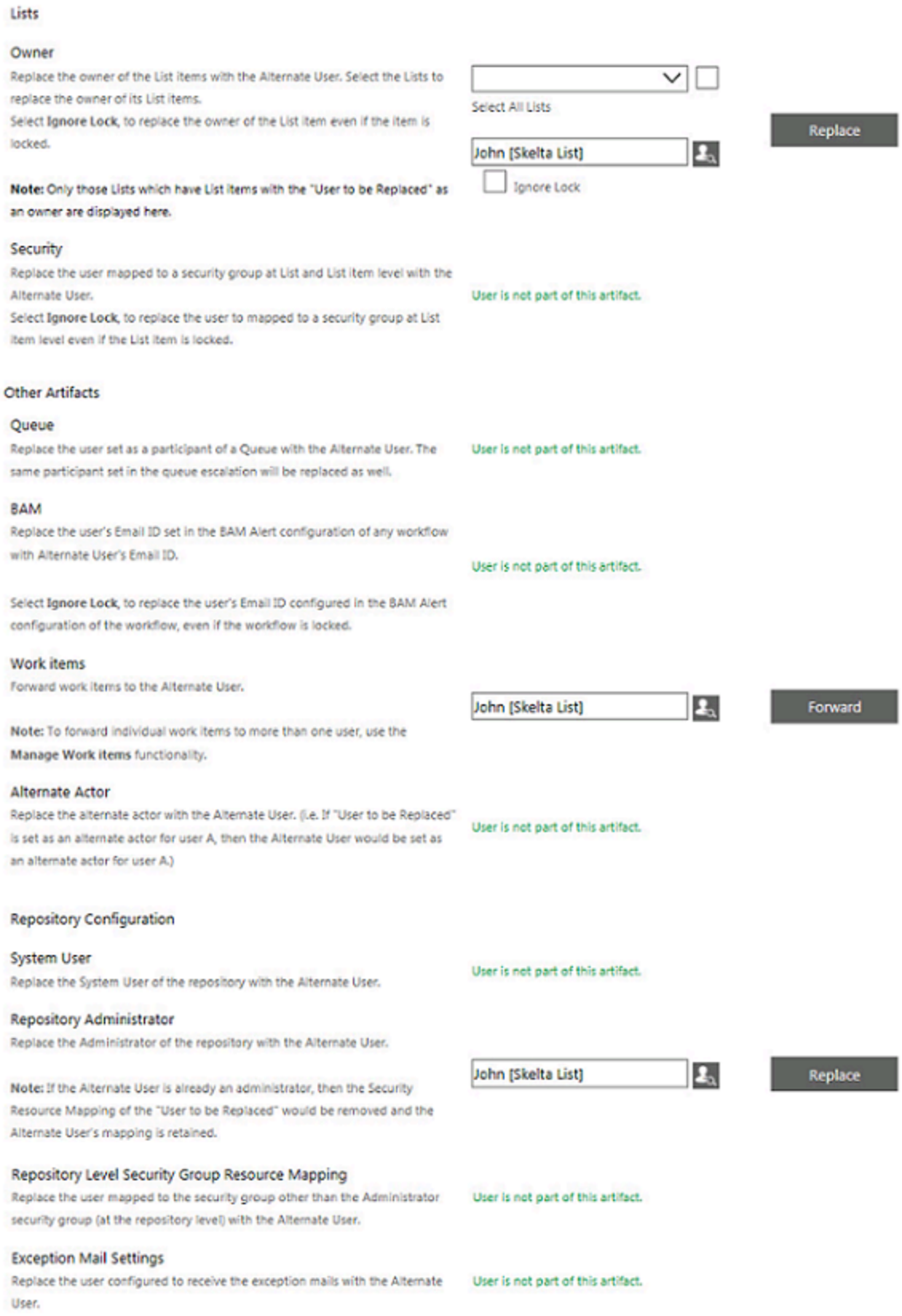
   - For Active Directory: **activedirectory::john22**
   - For the List Provider: **skeltalist:: C99B4110-C8CC-410B-BF70-93F3A7E34EDF**
   - For SQL Provider: **sqlprovider::6fb03011-68b7-437e-972e-21c67e4e2fe7**
   - For ASP.NET membership: **membershipprovider::a9bfc969-cc82-4c4b-99b9-50d9e1a91c01**
4. In **Select the Alternate User field**, click **User Lookup**. The **User Lookup** dialog box appears.
5. Search and select for an user who will be used as an alternate user, and then click **Update**. The selected user will be updated in the **Select the Alternate User** field. The user John22 will be updated as an Alternate User for John in all the artifacts of the repository.

   Note: In the Select the Alternate User field, you cannot enter the Real Actor Id, the Alternate User has to be selected through the User lookup.
6. Click **Populate**. The Replace User dialog will be refreshed and the section **Artifacts** will be displayed. By default the John will be populated as an Alternate User for each of the artifacts.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Note: You can override the Alternate User which is populated for each of the artifacts.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Replace User in Artifacts of a Repository

You can replace a user in the artifacts of a repository in AVEVA Work Tasks.

Lists

This section provides information on how to replace list item owners or users in security resource mapping.

Replace List Item Owners

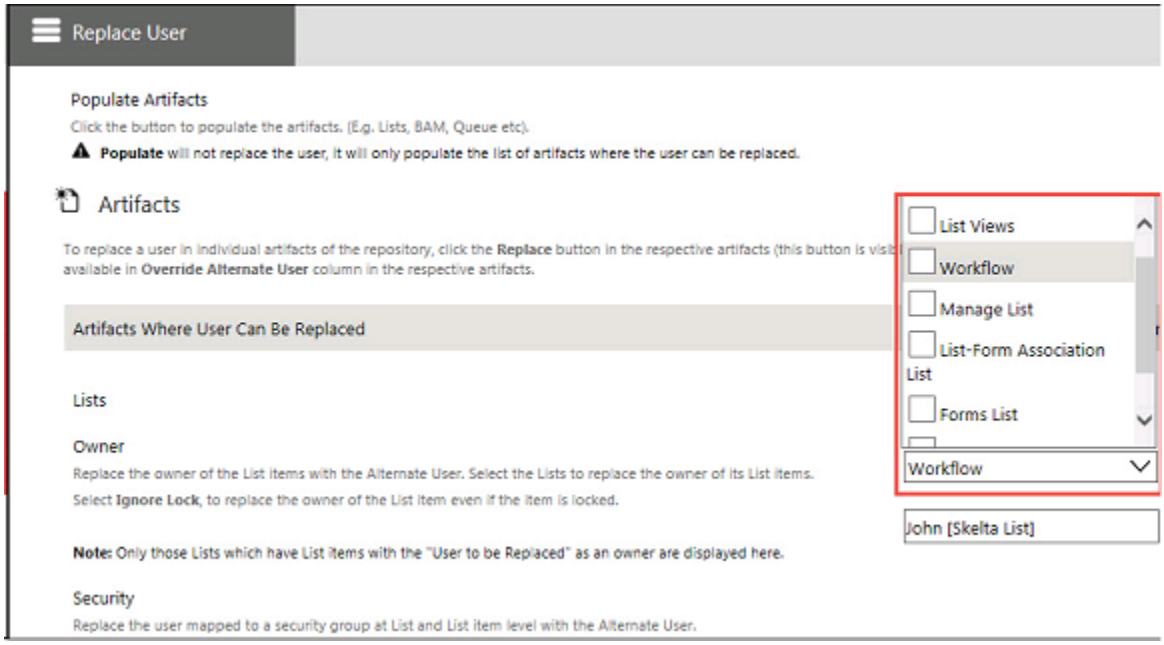
Replace List Item Owners

The **Owner** option in the Lists Artifacts helps you replace an user who owns list items. By default the user selected in the Alternate User field will be populated for this field. However, you can override this and select another user as an Alternate User.

To Replace Owner of List items

1. In the **List** drop down, select one or more lists.

   This drop down will list only those lists which have list items wherein the User to be Replaced (Rick) is a owner.



2. To select all the lists which are displayed in the drop down check the option **Select All Lists** .
3. The Alternate User for Owner Artifacts is populated by default (for more information on how this information is populated click here), if you want to override it click on the **User Lookup** icon to search and select the required user.
4. Select the option **Ignore Lock**, if you want to change the owner for the list item which is locked. Here the lock

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

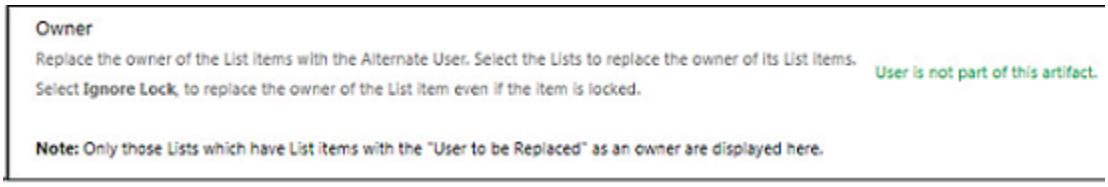
applied on the list item will not be considered and the owner information will be updated even though the list item is locked.

If the option **Ignore Lock** was not selected, then on replacing the owner a message error occurred while replacing user, check logger for more details will be displayed and the Owner for the locked list items will not be changed.

5. Click **Replace**, a message 'Are you sure you want to Replace is displayed', click **OK** to replace the User to be Replaced (Rick) with Kevin.

   This will replace the owner of the list items where *Rick* is the owner, with the **Alternate User** Kevin for all the selected lists.

   Note: If the User to be Replaced is not an owner for any of the list items, then the Owner Artifact will be updated as *User is not part of this Artifact*



Replace User in Security Resource Mapping

Replace User in Security Resource Mapping

The **Replacing User in Security Resource Mapping** option in the Lists Artifacts helps you replace the user mapped to a security group at List and List item level with an Alternate User.

- Viewing the report and replacing the security group resource mapping of Lists where User to be Replaced Rick is mapped and alternate user John is not mapped. From this option you can access the **Security Group - Resource Mapping Report** dialog, wherein you can select a list and view the security group mapping of the User to be Replaced for the selected list and/or list item level.
  - Select the check box beside each of items, click **Replace**. The selected Alternate User will be mapped to the **User to be replaced Security Group** for the selected list or list item level.
  - If the check box at the header level is selected, all the items will be selected.
- Viewing the report and replacing the security group resource mapping of Lists where both User to be Replaced Rick and alternate user John are mapped. From this option you can access the **Security Group - Resource Mapping Report** dialog, wherein you can select a list and view the security group mapping where both the User to be Replaced and Alternate User are mapped to the same list or list item level.
  - Select the check box beside each of items, click **Replace**. The selected Alternate User will be mapped to the **User to be Replaced Security Group** for the selected list or list item level.

Scenario 1

If both the users are mapped to the same security group

Consider Rick (User to be Repalced) and John (Alternate User) are mapped to the security group Contributor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The security group mapping of Rick (User to be replaced) done for the selected list/list item level will be removed.
- The security group mapping of John (Alternate User) will be retained for the selected list or list item level.

Scenario 2

If the User to be Replaced and Alternate User are mapped to different security groups

Consider Rick (User to be Replaced) is mapped to security group Administrator and John (Alternate User) is mapped to the security group Contributor.

- The security group mapping of Rick (User to be replaced) done for the selected list/list item level will be removed.
- The security group mapping of John (Alternate User) will be changed to Administrator for the selected list or list item level.

Consider Rick (User to be Replaced) is mapped to security group Reader and John (Alternate User) is mapped to the security group Administrator.

- The security group mapping of Rick (User to be replaced) done for the selected list/list item level will be removed.
- The security group mapping of John (Alternate User) will be changed to Reader (security group of User to be Replaced) for the selected list or list item level.

Note: This dialog is displayed to ensure the user can view that the User to be Replaced and Alternate User belong to different security groups. As we have no defined hierarchy among security groups the administrator needs to take the required action to ensure the Alternate User gets all the required privileges.

Other Artifacts

This section shows the different ways to replace a user:

- Queue: This option in **Replace User** allows you to replace the participants of a queue with an alternate actor.
- BAM: This option allows you to replace the email address of the user who is configured to receive BAM alerts of a workflow with an alternate user's email address.
- Work Items: The **Work Items** option allows you to replace an actor to whom the work item is assigned with an alternate actor.
- Alternate Actors: The **Alternate Actor** option allows you to replace an actor who is set as an alternate actor with an alternate user.

Queue

Queue

This option in Replace User allows you to replace the participants of a queue with an alternate actor.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Functionality

- If the *User to be Replaced* is not a participant of any Queue, then the message *User is not a part of this artifact* will be displayed.
- If the *User to be Replaced* has picked up any work items, then it has to be manually forwarded to another user. In case the work items picked up by the User to be Replaced are not forwarded before replacing the user, a message indicating the same will be displayed and an entry will be logged in the logger.
- When both the *User to be Replaced and Alternate User* are the participants of the Queue, then the entry for the *User to be Replaced* will be removed and the entry for Alternate User will be retained.
- When only the User to be Replaced is a participant of the queue and the Alternate User is not a participant of the same queue, the Alternate User will be added as a participant of the queue and the entry for the User to be *Replaced* will be removed.
- When replacing the queue participant, the rights of the *Alternate User* will be updated to the rights of the *User to be Replaced*.

Example

1. Let the User to be *Replaced be Lee and the Alternate User* be Royan.
2. Lee has higher rights compared to Royan.
3. On replacing the *User to be Replaced (Lee)* with Alternate User (Royan), rights of Royan will be updated and will be the same as Lee. This action will also remove Lee from the Queue. Therefore, Royan will now have higher rights. In the above case, if Royan had higher rights when compared to Lee, Royan's rights will be updated and would be updated to what Lee had. Therefore, now Lee will have lesser rights.

BAM

BAM

The **BAM** option, allows you to replace the email address of the user who is configured to receive BAM alerts of a workflow, with an alternate user's email address.

Functionality

- If the email address of the *User to be Replaced* is not configured to receive any BAM Alerts, then the message *User is not a part of this artifact* is displayed.
- If both the *User to be Replaced and the Alternate User* are configured to receive BAM Alerts, then the entry for the *User to be Replaced* will be removed and the entry for *Alternate User* will be retained.
- If the *Alternate User* has no email address defined, then a message *Error occurred while replacing the user configured in the BAM Alert configuration of workflow(s).Check logger for more details is displayed. In the logger you will find a message, that the email address is not configured. To proceed you have to configure the email address of the Alternate User*.
- If the option **Ignore Lock** is selected, then the email address of the *User to be Replaced* **will be replaced** with

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

the email of the Alternate User even for locked workflows having BAM Alerts configured.

- When the option **Ignore Lock** is not selected, then the email address of the *User to be Replaced* **will not be replaced** with the email of the *Alternate User* for locked workflows having BAM Alerts configured.
- When the option **Ignore Lock** is not selected, then the email address of the *User to be Replaced* **will not be replaced** with the email of the Alternate User for locked workflows having BAM Alerts configured, even if the logged in user has locked the workflows.

Work Items

Work Items

The option Work Items, allows you to replace an actor to whom the work item is assigned with an alternate actor.

Functionality

- If no work items are assigned to the User to be Replaced either through Web Channel or Mail Channel, then a message User *is not a part of this artifact* is displayed.
- To re-assign the work items of the User to be Replaced to different users, you have to user the Manage Work Items feature available through **Tools > Manage Work items**.
- To re-assign the work items of the User to be Replaced to the same user, you can use this **Replace User** feature.

Alternate Actor

Alternate Actor

The option Alternate Actor, allows you to replace an actor who is set as an alternate actor with an alternate user.

Functionality

- If the *User to be Replaced* is not set as an alternate actor, then a message User not part of this *artifact* is displayed.
- If both the *User to be Replaced and the Alternate User* are set as an alternate actor, then the entry for the *User to be Replaced* will be removed and the entry for the *Alternate User* will be retained.
- If the *Alternate User* is set as the alternate actor for the *User to be Replaced*, then the rule of alternate actor will be removed for the User to be Replaced. For more details read the scenario below:
- **Scenario:**
  - *User to be Replaced is Lee*
  - *Alternate actor set for Lee is Kevin*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- *Alternate User is Kevin*

- In this scenario since the Kevin who is the Alternate User is already set as the alternate actor for Lee, the rule of Alternate Actor for Lee will be removed and Lee will not have any alternate actor set.

Repository Configuration

You can replace a user at different levels of repository configuration:

- System User: The **System User** option allows you to replace an actor who is the system user with an alternate user.
- Repository Administrator: The **Repository Administrator** option allows you to replace the repository administrator with an alternate user.
- Repository Level Security Group Resource Mapping:The **Repository Level Security Group Resource Mapping** option allows you to replace the user who is mapped to a security group other than the administrator with an alternate actor.
- Exception Mail Settings: The **Exception Mail Settings** option allows you to replace an user who is configured to receive exception mails with the Alternate User.

System User

System User

The option **System User**, allows you to replace an actor who is the system user with an alternate user.

Functionality

If the *User to be Replaced* is not a system user, than the message *User is not a part of this artifact* is displayed.

Repository Administrator

Repository Administrator

Repository Administrator option allows you to replace the repository administrator with an alternate user.

Functionality

- If the *User to be Replaced* is mapped to a security group other than the administrator (for example contributor or reader), then a message *User is not part of this artifact* is displayed.
- If the *Alternate User* is not an administrator at repository level, than the security rights of the *Alternate*

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

User will be updated and the security rights of the *User to be Replaced* will be applicable.

- If the *Alternate User* is an administrator at repository level, than the entry of the *User to be Replaced* will be removed and the *Alternate Actor* will be replaced.

Repository Level Security Group Resource Mapping

Repository Level Security Group Resource Mapping

Repository Level Security Group Resource Mapping allows you to replace the user who is mapped to a security group other than the administrator with an alternate actor.

Functionality

- If the *User to be Replaced* is mapped to a security group other than the administrator (for example: mapped to security group contributor or reader), a message *User is not part of this artifact* is displayed.
- If the *User to be Replaced* has higher security rights than the Alternate User, then the security rights of *Alternate User* will be updated and the security rights of the *User to be Replaced* will be applicable to the *Alternate User*.
- If the *Alternate User* has higher security rights than the *User to be Replaced* , then the security rights of *Alternate User* will be updated and the security rights of the *User to be Replaced* will be applicable to the Alternate User. In this scenario the *Alternate User's* security rights will now be lower. For example: If the *Alternate User* (Royan) is an administrator and the *User to be Replaced* (Lee) is a contributor, then the *Alternate Users's* (Royan) security right will be updated as contributor.

Note: A report to view the security group resource mapping done for the User to Replaced is not available for this artifact like Replacing User in Security Resource Mapping.

Exception Mail Settings

Exception Mail Settings

The **Exception Mail Settings** option allows you to replace an user who is configured to receive exception mails with the Alternate User.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Functionality

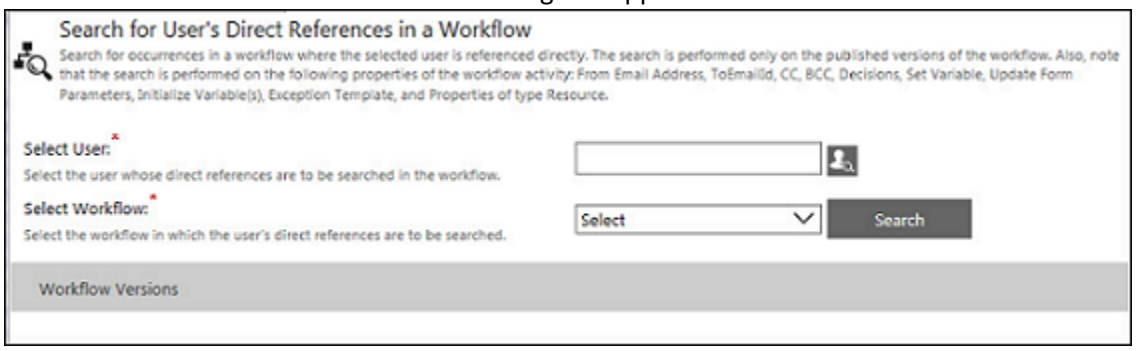
- If the *User to be Replaced* is not configured for any exception mails, then a message *User is not a part of this artifact* is displayed.
- If the *User to be Replaced* is the only user configured for exception mails, then the entry for the *User to be Replaced* will be removed and an entry for the Alternate User will be added.
- If both the *User to be Replaced* and the *Alternate User* are configured for exception mails, then the entry for the *User to be Replaced* will be removed and the entry for the *Alternate User* will be retained.

View User Reference

This option of **User Tools** helps you search for an Actor's direct references in the published versions of the workflow.

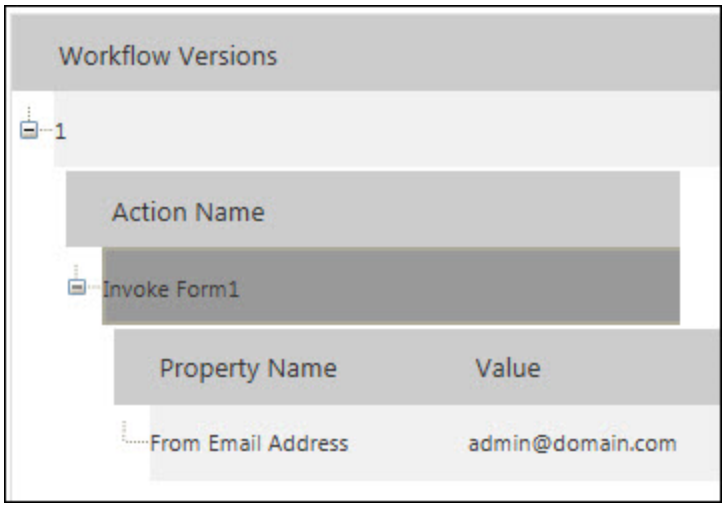
To Search for an actor's direct references in a Workflow

1. Log in to **Enterprise Console** and navigate to **Tools** > **User Tools** > **View User Reference**. The **Search for User's Direct References in a Workflow** dialog box appears.



2. Click on the **User Lookup** icon beside the **Select Actor** field. The **User Lookup** dialog is displayed.
3. Search and select the User.
4. Click **Update**. The selected actor will be updated in the **Select Actor** field.
5. Click on the drop down in the **Select Workflow** field and select a workflow to check if the actor *Rick* is directly referenced in it.
6. Click **Search**. The **View Workflow User Reference** dialog will be refreshed and all published versions of the selected workflow where the selected actors direct reference is used will be displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Published versions of the workflow will be displayed here when the actors direct reference is present in any one of the following activity properties in the workflow:

- Property of type resource is used like:
  - Assign Actor in Human Activities.
  - Communication Activities:
    - To in Email Activity
    - CCResource in Email Activity
    - BCCResource in Email Activity
    - RecipientResource in Send SMS Activity
  - Engine Activity
    - Actors in Resource Variable Activity
- From Email Address, ToEmailId, CC, BCC properties across all activities
- Decision property of Decision Activity
- Set Variable property of Update Variable Activity
- Set Form Parameters of Invoke Form Activity
- Initialize Variable(s)and Exception Template in Start Activity

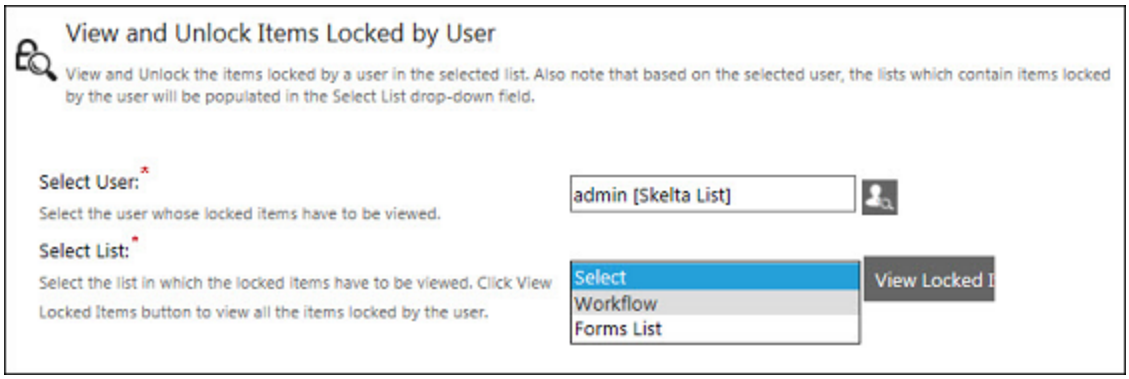
Unlock Locked Items

The **Unlock Locked Items** option in **User Tools** allows you to view the list items locked by the selected user and also unlock it if required.

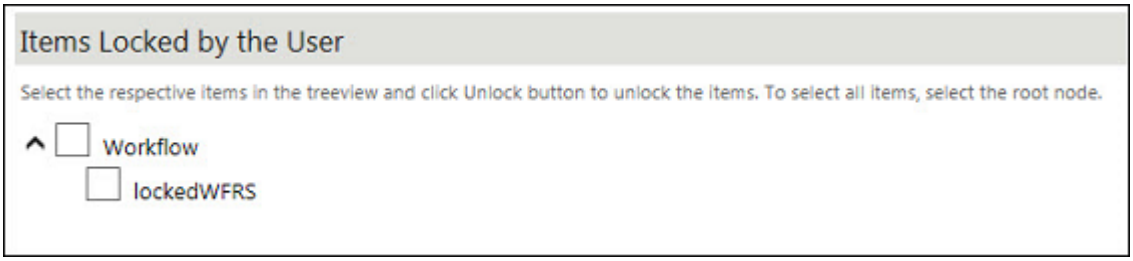
To View and Unlock Locked Items

1. Log in to **Enterprise Console** and navigate to **Tools** > **User Tools** > **Unlock Locked Items**. The **View and Unlock Items Locked by User** window is displayed.
2. To select an user to view the items locked click on the **User Lookup** icon in the **Select User** field. The **User Lookup dialog** is displayed, select the user, for example Kirsten.
3. In the **Select List** drop down, select the List to view the locked items.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



4. Click **View Locked Items**, the **View and Unlock Locked Items** window will be refreshed and the items locked by the selected user for the selected list will be displayed in the **Items Locked By User** section.



   In the image above the selected user is admin and the selected list is Workflow. Hence the **Items locked by User** section displays the workflows locked by *admin*.

5. Select the items to be unlocked and click **Unlock**.
   - To unlock all the items of the List select the parent node, in our example selecting workflow will ensure all the locked items displayed in this window are selected.
   - To unlock individual items select the check box beside the appropriate one.
   - If the parent workflow is not locked and only the child workflow is locked, the Items locked by User section will display only the locked child workflows.

Deleted List Items

The **Deleted List Items** feature of AVEVA Work Tasks contains a record of all the deleted list items from lists such as the Workflow List, Forms List, Navigation List and also other dynamic lists.

It records details such as the Date Time, List Type, List Owner, and other relevant details. There is a provision to export the Definition Xml for the lists where it is applicable.

Introduction

Most of the process centric organizations are required to protect and maintain the information of deleted or discontinued records.

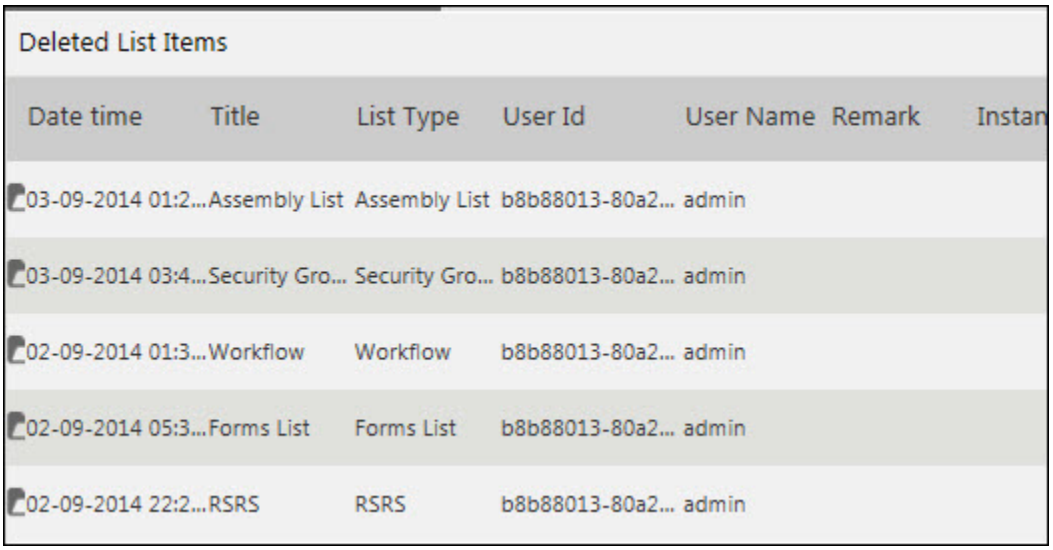
Say, for example in scenarios where a process has to be discontinued due to a business requirement, AVEVA Work Tasks allows you to delete the workflow which controls that process. Similarly, list items can be deleted from other lists such as Forms List, Navigation List and other dynamic lists. However, a record of the deleted information can now be maintained in a separate list that is, the **Deleted List Items**.

The **Deleted List Items** feature of AVEVA Work Tasks contains a record of all the deleted list items from lists such as the Workflow List, Forms List, Navigation List and also other dynamic lists. It records details such as the Date

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Time, List Type, List Owner, and other relevant details. There is a provision to export the Definition Xml for the lists where it is applicable.

The deleted list items information is stored under folders of the respective list type. Say, if a workflow is deleted, a list item is added to the **Deleted List Items** list under the Workflow folder. If a Form is deleted, a list item is added to Deleted List Items list under the Forms folder.



Click here for a scenario to understand the use of **Deleted List Items** in a business environment.

Use Deleted List Items

Consider a scenario from a pharmaceutical company, say Telematics Pharma which has a set of critical business processes being controlled through the use of workflows.

Workflows are designed to automate the business processes. So, whenever any workflow is removed or deleted from the system, the company wishes to maintain a record of that. The record maintained must have details such as the Workflow Owner, Modified Date Time etc. Also there should be a provision to retrieve the deleted workflows in the form of Definition XML whenever required.

The **Deleted List Items** feature helps the company to achieve the above scenario.

Steps to achieve the scenario

1. Telematics Pharma has a workflow named WF_Check_Ingredients, which has to be discontinued due to some business requirements. However, a deletion record has to be maintained for the deleted workflow. Right-click and select **Delete** to delete the workflow.
2. Open **Deleted List Items** from the Enterprise Console menu.
3. A folder **Workflow** gets created in the **Deleted List Items** list.
4. Double-click the Workflow folder to see the record of the deleted workflow.

   Find below a brief description of some of the important fields being displayed for the deleted list item:

Column Name	Description
Title	This field contains the title of the list item which has

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Column Name	Description
	been deleted.
List Type	This field contains the type of the list item which has been deleted. For example, Workflow, Form.
User Id	The user id field is populated based on the provider configured for the repository: • **Active Directory provider: Name** is populated as the User Id. • **Forms Provider: GUID** is populated as the User Id. • **ASP.NET Membership: Name/GUID/ Integer** (depending on the provider configuration)
Instance Xml	This field contains the form instance XML of the deleted list item. **Note:** Each list has been internally mapped to a form and each list item is mapped to a run-time instance of the form. The Form Instance XML contains the run-time information of the form associated with the list.
Item Details	This field contains information about the list item like Title, Owner, Version History, Created Date Time, and Last Modified Date Time.
Audit Details	This field contains the audit history of the deleted list item.
Definition Xml	This field is applicable only for list types such as workflows, forms, and lists. The exported definition XML of the deleted list item can be used to restore the list item in any repository.

You can export the definition XML of the workflow using the option available in the context menu. The exported XML can then be used to import the workflow definition in any repository, if required.

Enable or Disable Deleted List Items feature for a Repository

This section explains how to enable or disable the Deleted List Items at the repository level.

Default Behavior of Repositories Created in AVEVA Work Tasks

In AVEVA Work Tasks, Deleted List Items will be enabled by default for a repository that is created. Whenever any list item of any list type is deleted, a new list item is added to the Deleted List Items under the respective folder.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: For a fresh installation of AVEVA Work Tasks, the Deleted List Items is enabled by default. However, after upgrading an existing installation of AVEVA Work Tasks, the Deleted List Items feature is disabled by default.

How to Check Whether Deleted List Items is Enabled or Disabled for a Repository?

To check whether **Deleted List Items** is Enabled or Disabled for a Repository verify the Properties Column (properties collection) of SKRepository table in the Farm Database. When the value of the property key **"isDeletedListItemsTraceEnabled"** is **True**, **Deleted List Items** is enabled for the repository. If the value of the key is **False**, **Deleted List Items** is disabled for the repository.

For example, find below the Properties column (properties collection) data for a repository.

```
<properties>
<property key="sourceelements">
<![CDATA[WORKFLOWELEMENTS\Default]]>
</property>
<property key="fromemail"><![CDATA[admin@domain.com]]></property>
<property key="toemail"><![CDATA[admin@domain.com]]></property>
<property key="alertuser"><![CDATA[False]]></property>
<property key="filewatcher"><![CDATA[]]></property>
<property key="enablelock"><![CDATA[True]]></property>
<property key="isadminconfigured"><![CDATA[True]]></property>
<property key="isDeletedListItemsTraceEnabled"><![CDATA[True]]></property>
<property key="theme"><![CDATA[Default]]></property>
</properties>
```

The line highlighted in red in the above code, is used to identify whether **Deleted List Items** is enabled or disabled for a repository. In this line, you can identify the following:

<property>	**Defines the property tag**
key	Attribute of property tag, describes the property name (isDeletedListItemsTraceEnabled is name of the property)
<![CDATA[True]]>	It is the value of the property (True- Deleted List Items is enabled)
</property>	Closing tag of property

Enabling Deleted List Items for a repository

In AVEVA Work Tasks, once you create a repository, **Deleted List Items** will be enabled by default for that repository.

If you want to enable **Deleted List Items** for a repository, then open the SKRepository table in the Farm Database and find out the required repository entry (row). In that specific row in properties column data (properties collection) find out the **isDeletedListItemsTraceEnabled** key and update the `<![CDATA[False]]>` to `<![CDATA[True]]>`.

If the i**sDeletedListItemsTraceEnabled** key is not found for a repository entry in the properties column then add the following line between <properties> & </properties> tags.

```
<property key="isDeletedListItemsTraceEnabled"><![CDATA[True]]>
</property>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Now **Deleted List Items** will be enabled for that repository. If you want to enable Deleted List Items for another repository perform the steps mentioned above.

Disabling Deleted List Items for a repository

In AVEVA Work Tasks, once you create a repository **Deleted List Items** will be enabled by default for that repository.

If you want to disable **Deleted List Items** for a repository, for which the **Deleted List Items** is already enabled, then open the SKRepository table in the Farm Database and find out the required repository entry (row). In that specific row, in properties column data (properties collection) find out the **isDeletedListItemsTraceEnabled** key and update the `<![CDATA[True]]>` to `<![CDATA[False]]>`.

If the **isDeletedListItemsTraceEnabled** key is not found for a repository entry in the properties column, then add the following line between <properties> & </properties> tags in case if you want to enable Deleted List Items in future.

```
<property key="isDeletedListItemsTraceEnabled"><![CDATA[False]]>
</property>
```

Package

Using the **Package** feature of AVEVA Work Tasks, you can deploy one or more artifacts from one environment or repository to another in a easy and simple manner.

The artifacts can be deployed along with its associated and child artifacts, configuration state and containment state from one environment or repository to another.

Prerequisites

- Ensure that the **AVEVA Work Tasks-Client Service** is running while installing or uninstalling the package.
- In an environment that consists of multiple servers and clients, ensure that the **AVEVA Work Tasks-Client Service** is running on the machine which has the Enterprise Console site configured.

Note: If you have created and exported a package in the previous version of AVEVA Work Tasks, then ensure that you use the previous version to import the package. Otherwise, you must upgrade to the latest version of AVEVA Work Tasks, create new package template, add the artifacts again to the package template, and then export the package.

For more information about package, see *Packages* section in the *Developer Guide*.

Advantages of Package

Currently, packaging and deploying one or many artifacts from one environment or repository to another is done either by exporting and importing the artifacts. You can use the recovery task of packaging to deploy one or many artifacts from one environment or repository to another.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Package Versus Export Import

- Export and import feature is helpful when you want to deploy a single artifact in a different environment.
- You can export and import only one artifacts at a time.
- An imported artifact is not always valid on the target environment, if the associated artifacts are not present.

Example:

Create a '*LeaveWorkflow*' add an Information activity and assign it to a queue say 'UsersQueue'.

Export the '*LeaveWorkflow*' from the source environment and import it to the target environment. The target environment has no queues created.

Since '*UsersQueue*' does not exist in the target environment, the '*LeaveWorkflow*' is invalid.

You have to manually create the 'UsersQueue' in the target environment.

Package Versus Recovery

The recovery feature allows you to replicate the entire repository with all the artifacts in the target environment. A choice to add only few artifacts to the target environment is not available. This option is beneficial when the entire repository needs to be deployed in another environment.

About Package

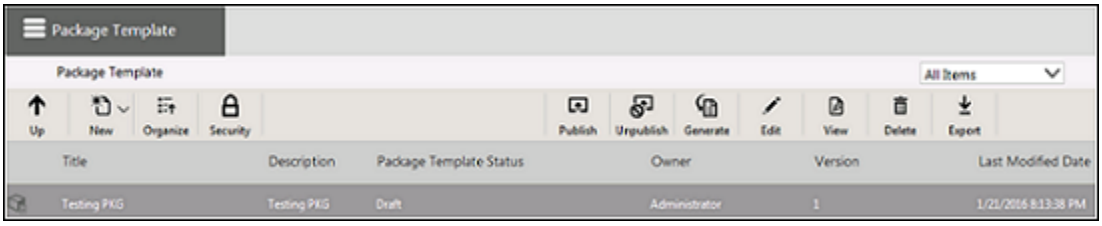
A proper understanding of the **Package** interface allows you to use the various features effectively. The Package interface consists of the following components:

- Package Template: Use to design a Package, define artifacts to the template, and complete the entire process of designing and building a Package to exporting the package file.
- Manage Package: Use to import packages that are exported from the Package Template of source environment, and install it in the target environment.

Note: Access to the above mentioned functionalities is based on the rights given to the users. The functionality will either be disabled or it will not be visible if the user does not have rights to execute them.

Package Template

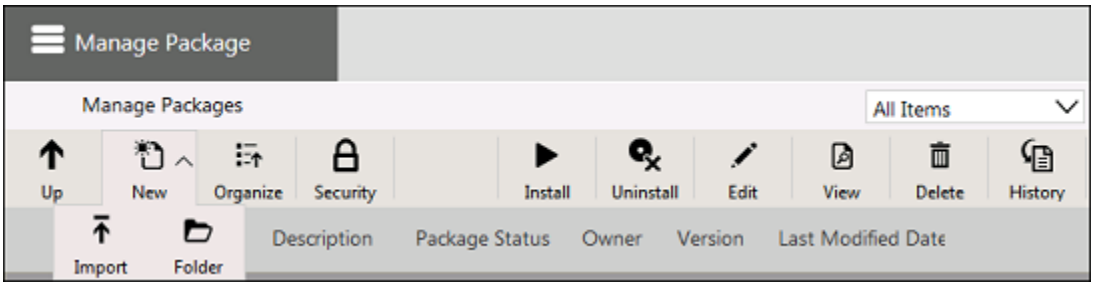
You can use the **Package Template** list in the source environment to create and design a package template. You can add artifacts and update the configuration settings to the created package template. You can complete the entire process of designing, publishing, generating, and exporting the package file. The behavior of the package at the time of installation is defined in a **Package Template**.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Manage Package

You can use the **Manage Package** list in the target environment to import packages that are exported from the Package Template of source environment, and install, update, or uninstall the package in the target environment. You can also change the pre-defined configuration that affects the behavior of package at the time of installation, updates, or uninstallation.



ArchestrA Events

You can create ArchestrA applications that generate alarms and events to provide the status of a running application. ArchestrA Events is such an application of AVEVA Work Tasks which you can use to consume events from the event providers of the ArchestrA Event Services platform. The ArchestrA Event Services platform uses the ArchestrA Service Bus (ASB) for sending and receiving data. When an event occurs, a notification is reported to the event distributor. Then based on the filtering conditions, the required the workflow is triggered.

Alarms warn about process conditions that can potentially cause problems. Typically, you set up an alarm to become active when a process value exceeds a defined limit. For example, you can set an alarm for a pump that warns when no fluid pressure is detected.

Events represent normal system, application, or user occurrences that produce status messages. A typical event occurs when an operator logs on to an application at the beginning of a work shift. Application Server can detect events, store them as historical data, and report them to client applications.

ArchestrA objects include built-in event and alarming reporting capabilities. You must configure alarms for each object in the IDE to use the event and alarm functions.

Application Server creates several types of run-time event messages, which are saved as historical data.

- System events
- Security-audit (operator change) events
- Application (or process) related events

For more information about these ArchestrA events, see *AVEVA Application Server User Guide*.

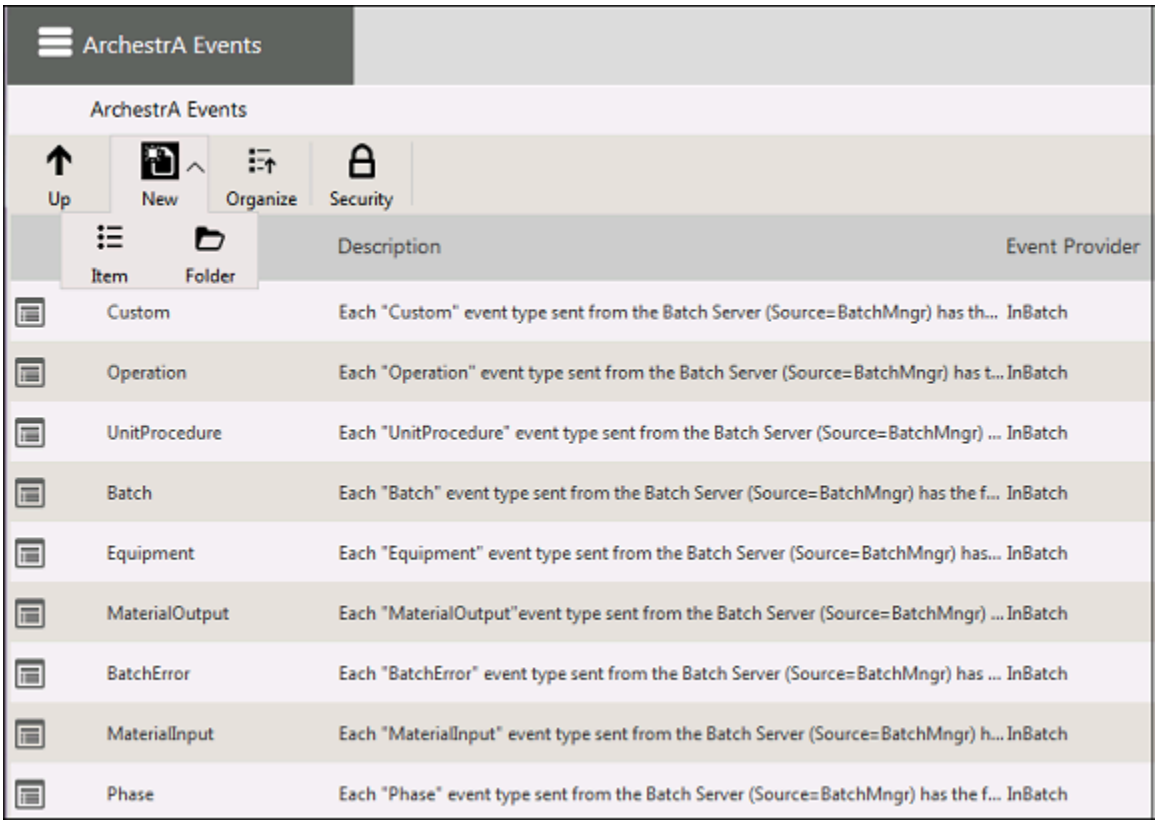
ArchestrA Events and AVEVA Work Tasks

You can use the ArchestrA Events window to create an *ArchestrA Events List Item* in the Enterprise Console to store ArchestrA Event Type and the relevant context data properties associated with the event type. When the set filter conditions are met, based on the tasks scheduled, the workflow is triggered.

The following image shows the main list of ArchestrA Events available by default for InBatch. This main list contains the predefined event types of InBatch event provider. You can use this main list while associating a

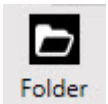
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

workflow to an ArchestrA Event. For more information about workflow and ArchestrA Events association, see Associate Workflow with ArchestrA Events.

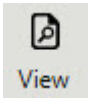
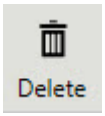


Actions Executable in the Ribbon Menu

The following table lists the actions that you can perform from the ribbon of the ArchestrA Events list:

Icon	Description
	Navigate icon to navigate from the current folder to the parent folder. This feature is helpful when you are accessing a folder within the ArchestrA Events and you want to navigate back to the parent folder of the ArchestrA Events List.
	New ArchestrA Events List Item icon to create an ArchestrA Events List Item.
	**Folder** icon to group and categorize the ArchestrA Events List Item.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Icon	Description
	Edit icon to modify the ArchestrA Events List Items.
	**View** icon to view the ArchestrA Events List Items in read-only mode.
	**Delete** icon to delete the selected ArchestrA Events List Items.
	Organize icon to organize the ArchestrA Events List Items and group it in specific folders.
	Security Settings icon to define security at the List Item level.

Actions Executable in the Shortcut Menu

The following table lists the shortcut menu items of all the actions that you can perform from the shortcut menu of the ArchestrA Events window:

Menu Item	Description
Edit	**Edit** option to modify the ArchestrA Events List Items.
Delete	**Delete** option to delete the selected ArchestrA Events List Items. .
Security Settings	**Security Settings** option to define security at the List Item level.

Associate Workflow with ArchestrA Events

Use the **Associate** option to associate a workflow with ArchestrA Events.

The ArchestrA Event association window appears.

To associate a workflow with an ArchestrA Event for a specific event

1. Select the workflow, and then click **Associate** from the ribbon.
   Alternatively, select and right-click the workflow, and then select **Create Associations**.
2. Click **ArchestrA Event** to associate the workflow with the ArchestrA Event list for a specific event.
3. Enter the following details:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Event Provider**: Select the name of the event provider from which you want to poll for new events.

  The Event Provider field is mandatory, and the maximum length of the event provider name is 250 characters.

  Based on the data available in the ArchestrA Event main list, the Event Provider list is populated. You can add other event providers to this main list, if required.

  Example of Event Providers are InBatch Server or Avantis.
- **Event Type**: Select the Event Type of the Event Provider.

  It is the type of event which defines the properties the event will have.

  For InBatch Server, Batch is an event, and possible values for Batch events are such as, Batch, Equipment, UnitProcedure, Operation, Phase, MaterialInput, MaterialOutput, Custom, and BatchError.

  Based on the data available in the ArchestrA Event main list, the Event Type list is populated for the selected Event Provider. You can add other event types for the event provider, if required.
- **Source**: Type the Source property name of the event. It is the name of the event provider sending the event. For example, for a *Batch* event, *BatchMngr* is the source.

  Select the appropriate operator from the operator drop-down list.
- **Area**: Type the Area of the event. Areas provide a key organizational role in grouping alarm information and assigning it to users who use alarm and event clients to monitor their Areas of responsibility.

  The Area object plays a key role in alarm and event distribution. All objects belong to an Area. Areas can contain sub-Areas. Alarm and event clients are configured to subscribe to a set of Areas.
- **Priority**: Type the priority for each designated event.

  Priority, which is an integer value from 1 to 999 indicating the severity of the alarm which raises an event. An alarm priority of 1 is most urgent and 999 least urgent.
- **Workflow Version**: Select the version of the workflow to be associated with the ArchestrA Event. The workflow will be initiated only if the published version at the time of the occurrence of the event is the same as the version selected here. To trigger the currently published version of the workflow whenever the event occurs, select *Latest*. When a version is specified, the workflow alerts the latest published version of that version series. For example, if the specified version is 1.1 and the latest published minor version for 1.1 is 1.1.1.3, then 1.1.1.3 is alerted.
- **Properties**: Based on the **Event Type** that you enter, select values in the Properties section. For example, for a Batch event type, BatchServer is a property of the data type String. The Properties field is optional.

  Based on the data available in the ArchestrA Event main list, the properties of the Event Type list is populated. You can add other properties of the event type, if required.

4. Click **Attach Event** to associate the workflow with ArchestrA Event for that specific event.

   Click **Cancel** to abort.

   The confirmation message, *Successfully associated the workflow* appears.

Limitations

- The *Between* operator is not supported for **Priority** and **Properties**.
- The data type of the operators are not validated. Ensure to provide valid values to the property data type.
- Only one workflow with one event provider is supported. That is, different event providers for a workflow with different association is not supported.
- While creating a filter condition, duplicate properties are not supported.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- The *OR* condition is not supported while creating a filter condition.

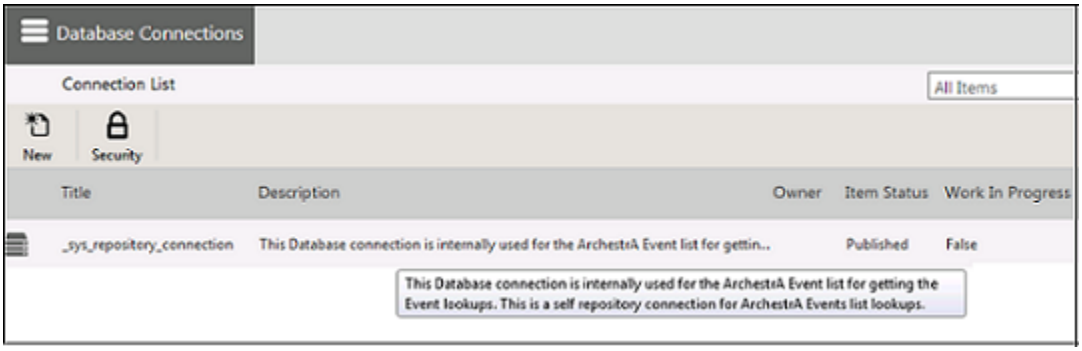
ArchestrA Events Lookup List

When you create a repository, the following system Lookup List items are created:

_sys_repository_connection:

This database connection is used internally by the ArchestrA Event List for getting the Event lookups. It is a system generated lookup, and we recommend to not delete it. This lookup does not have any owner as it a system generated connection.

The following image shows the **-sys_repository_connection** lookup list item:



Database Lookup Lists

A **System** folder with the following Lookup Lists item is created. This **System** folder is internally used by the ArchestrA Event list for getting the Event details. This is a system generated folder for ArchestrA Events list lookups, and we recommend to not delete the folder and its contents.

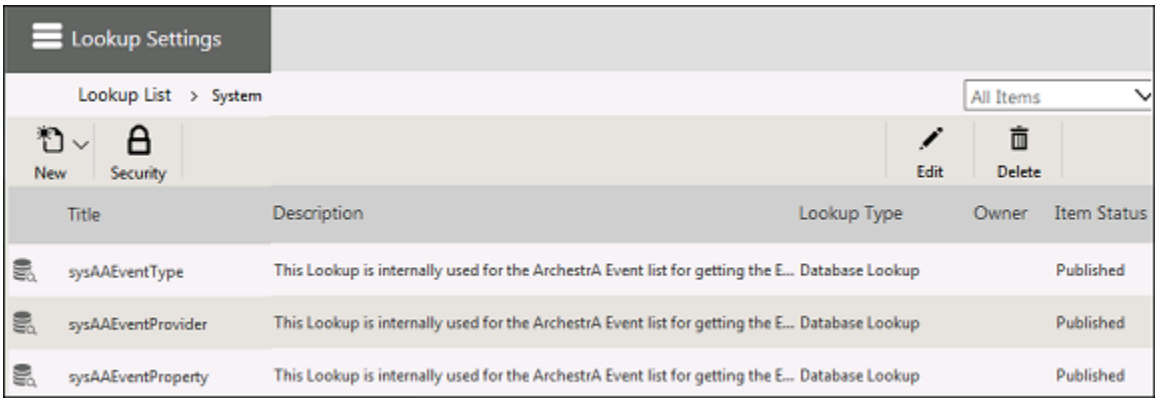
The **System** folder contains the following Lookup List items:

- **sysAAEventProvider**: Lookup item internally used by the ArchestrA Event list for getting the Event Providers.
  .
- **sysAAEventProperty**: Lookup item internally used by the ArchestrA Event list for getting the Event Properties.
- **sysAAEventType:** Lookup item internally used for the ArchestrA Event list for getting the Event Types.

The following image shows the **System** folder:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The following image shows the system generated Lookup items in the **System** folder:



Create ArchestrA Events List Item

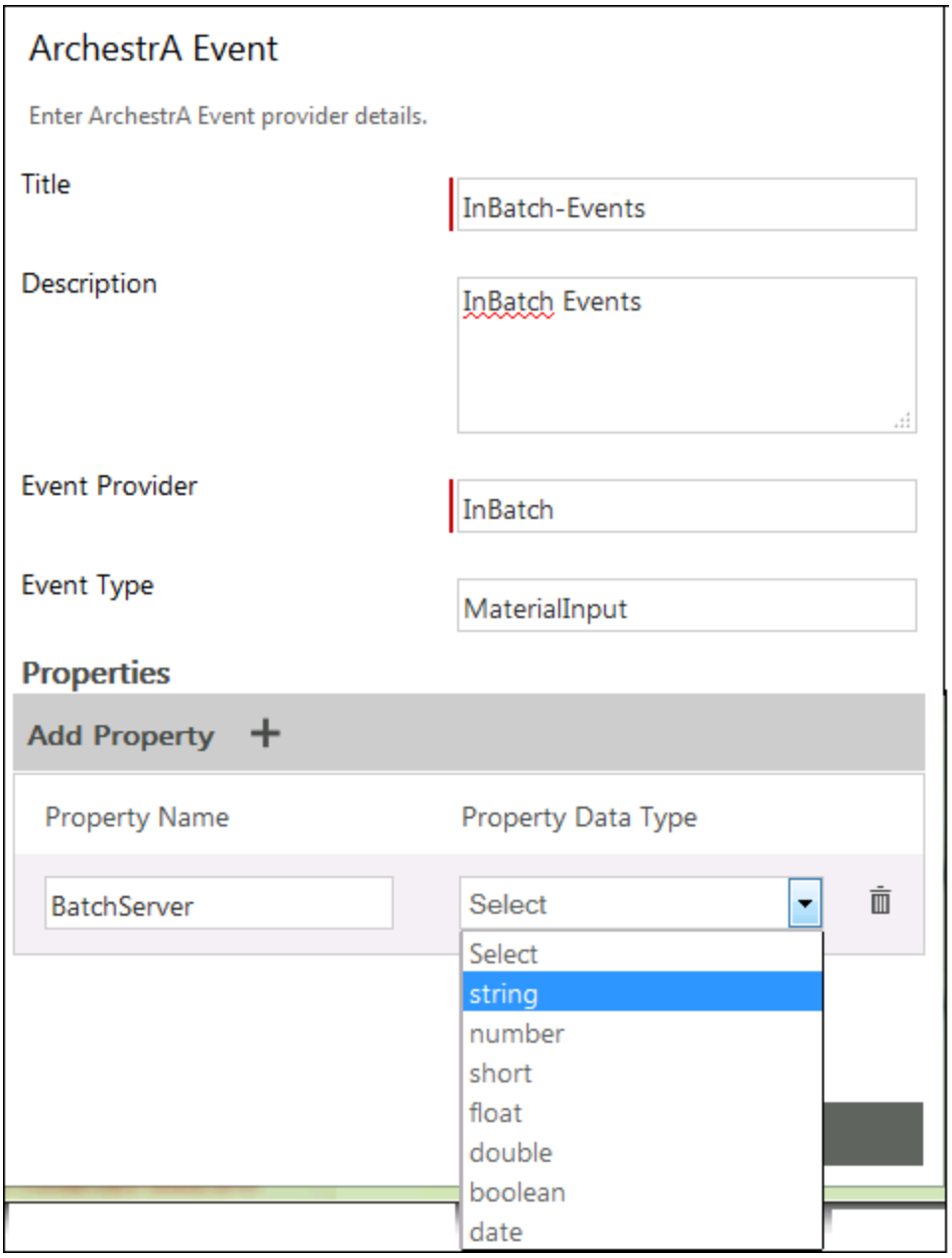
You can create an **ArchestrA Events List Item** in the **Enterprise Console** to store ArchestrA Event Type and the relevant context data properties associated with the event type. You can then use this list to populate the ArchestrA Event details received through the ArchestrA Service Bus (ASB) when an event occurs. When an ArchestrA Event occurs, and if the event conditions meet the filter conditions set by AVEVA Work Tasks, then a workflow triggers.

By default, the ArchestrA Event main list lists the event types and related properties of InBatch server. You can add more event types and properties to the existing list, edit the event types and properties, or even delete unwanted event types and properties.

You can also add other Event Providers, if required.

The following image shows the window to create an **ArchestrA Events List Item:**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



To create ArchestrA Events List Item

1. In the **Enterprise Console**, select menu icon, select **Tools**, and then select **ArchestrA Events**.
   The ArchestrA Events page appears.
2. On the action bar, click **New**, and then click **ArchestrA Events**.
   The New ArchestrA Events window appears.
3. In the **Title** box, type a name for the ArchestrA Events List Item, and in the **Description** box, type a description for the ArchestrA Events List Item.
4. In the **Event Provider** box, type the name of the event provider from which you want to poll for new events.
   Example of Event Providers are InBatch Server or Avantis.
   The **Event Provider** field is mandatory, and the maximum length of the event provider name is 250

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

characters.

5. In the **Event Type** box, type the Event type of the Event Provider.

   It is the type of event which defines the properties the event will have. For InBatch Server, *Batch* is an event, and possible values for *Batch e*vents are such as, *Batch, Equipment, UnitProcedure, Operation, Phase, MaterialInput, MaterialOutput, Custom*, and *BatchError*.

   The maximum length of the event type is 250 characters.

6. Based on the **Event Type** that you enter, type values in the **Properties** section. For example, for a *Batch* event type, *BatchServer* is a property name of the data type *String*.

   The **Property Name** field is optional. However, if you enter a property name, then the data type of the property is mandatory.

   The maximum length of the **Property Name** is 250 characters.

7. Click **Submit**.

   The ArchestrA Events List Item is saved successfully.

Edit ArchestrA Events List Item

Edit an ArchestrA Events list item to change description and Event Provider related information.

To edit an ArchestrA Events list item

1. In the **Enterprise Console**, click menu icon, click **Tools**, and then click **ArchestrA Events**.

   The ArchestrA Events page appears.

2. Right-click on an ArchestrA Event, and then click **Edit** (or click on an ArchestrA Event, and then on the action bar, click **Edit**).

   The Edit ArchestrA Event dialog box appears.

3. Change the ArchestrA Event details as desired.
4. Click **Save**.

   The ArchestrA Event details are saved.

Delete ArchestrA Events List item

Delete an ArchestrA Event List item when it is no longer required.

To delete an ArchestrA Event List item

1. In the **Enterprise Console**, select menu icon, select **Tools**, and then select **ArchestrA Events**.

   The ArchestrA Events page appears.

2. Right-click an **ArchestrA Event**, and then click **Delete** (or click on an ArchestrA Event, and then on the action bar, click **Delete**).

   A confirmation message appears.

3. Click **OK**.

   The confirmation message is closed, the ArchestrA Event is deleted, and an information message appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

4. Click **Close** (or click the Close button).

   The information message is closed.

Process Designer

About the Process Designer

The **Process Designer** provides the interface to model workflow processes. Workflows can be modeled, designed, and deployed from this window. It opens when the Process Designer control is executed from a web form in the web application.

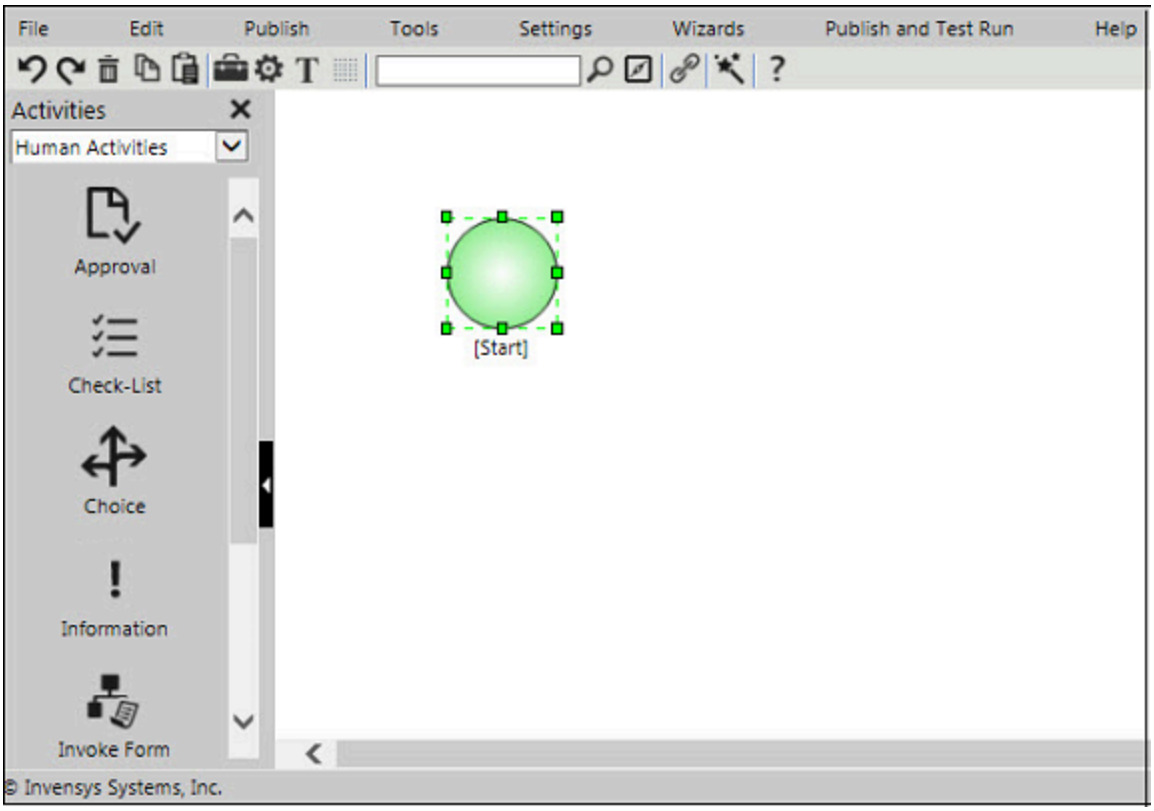
The **Process Designer** is an AJAX based ASP.NET server control that can be embedded in any web based product, ASP Engine, or web page. It is a browser based application that requires no plug-in or IDE. It provides a comprehensive set of standard activities, which can be easily dragged and dropped on a design mosaic and linked to design processes.

The **Process Designer** user interface has the following parts:

- Menu bar
- Tool bar
- Design Mosaic
- Activities
- Activity Properties

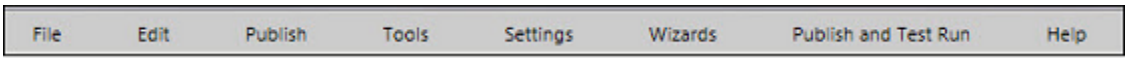
The following image shows the Process Designer window:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Menu bar

The following image shows the menu bar of the Process Designer window:

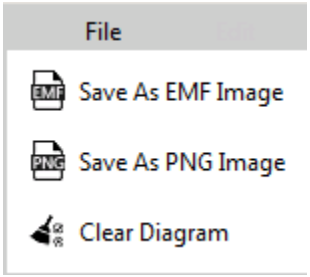


Note: To change the Menu in the Process Designer, edit the **pdmenu.txt** file under the **Locales** folder.

The menu bar has the following menus.

File

The **File** menu has the following options to perform various activities on the files in which workflow process designs are saved:



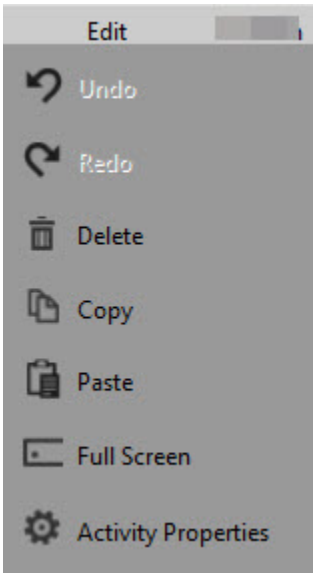
- Save As EMF Image: Use this option to save the current process design as an image file in the Enhanced Windows Metafile (EMF) format. This is the ideal format for printing the process or to embed it in MS Office files.
- Save As PNG Image: Use this option to save the current process design to a PNG image file.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Clear Diagram: Use this option to clear the current process design from the design area.

Edit

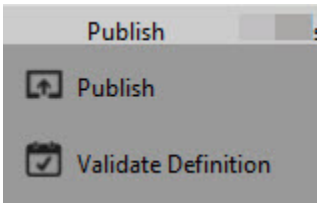
The **Edit** menu has the following options to perform various activities on the Workflow process design:



- **Undo:** Use this option to undo the previous actions.
- **Redo:** Use this option to redo the previous actions.
- **Delete:** Use this option to delete the selected activity.
- **Copy:** Use this option to copy the selected activity.
- **Paste:** Use this option to paste the activity copied using the **Copy** command.
- **Full Screen:** Use this option to switch between the full screen and normal display modes.
- **Activity Properties:** Use this option to open the **Activity Properties** window to set the properties for the selected activity.

Publish

A workflow is deployed and activated only if it is published.

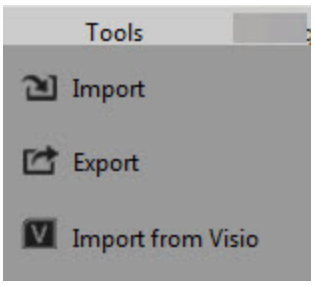


- **Publish**: Use this option to publish the workflow design.
- **Validate Definition**: Use this option to validate the workflow definition.

Tools

The **Tools** menu has the following options to export and import workflow definitions:

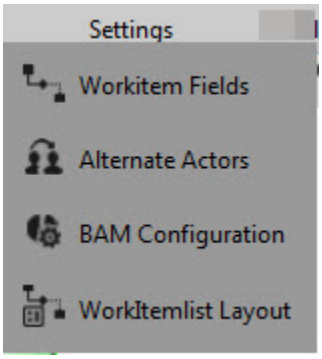
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



- Import: Use this option to import process definition XML files, which were exported earlier, into the Process Designer.
- Export: Use this option to export a workflow definition to an XML file. The exported XML definition can be manually edited or moved to another database.
- Import from Visio: Use this option to import a workflow process model XML file created using IT Pearls BPMN Stencil for Visio.

Settings

The **Settings** menu has the following options to configure different process level settings for other AVEVA Work Tasks Controls:



- Work Item Fields: Use this option to add and configure work item fields.
- Alternate Actors: Use this option to specify alternate actors to the actual assigned actors.
- BAM Configuration: Use this option to setup critical performance measurement criteria for a workflow.
- Work Item List Layout: Use this option to display all the work item fields that have been added.

Tool bar

The tool bar is located just below the menu bar at the top of the **Process Designer** window. It has the following tools:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Undo

The **Undo** tool can be used to undo the previous actions that were performed in the Process Designer.

Redo

The **Redo** tool can be used to repeat the previous actions that were performed in the Process Designer.

Delete

The **Delete** tool can be used to delete an activity from the workflow.

Copy

The **Copy** tool can be used to copy an activity in a workflow to paste it elsewhere.

Paste

The **Paste** tool can be used to paste a cut or copied activity in the workflow.

Activities Toolbox

The **Activities Toolbox** tool can be used to show or hide the Activities Toolbox (AVEVA Work Tasks activities), which has the list of activities.

Activity Properties

The **Activity Properties** tool can be used to show or hide the properties area for the selected activity.

Edit Text

The **Edit Text** tool can be used to edit the text corresponding to the selected element on the design mosaic. It could be an activity or a link.

Show Grid

The **Show Grid** tool can be used to show or hide the grid pattern on the design mosaic.

Search

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Search** tool can be used to search for a specific value in the workflow. You can enter the value to be searched in the field and click the magnifying glass icon next to it to start the search.

Zoom

The **Zoom** tool can be used to zoom in or out on the workflow design in the design mosaic. You can do this by specifying the percentage of zoom using the Zoom Level drop-down in the window that opens. You can also use this image to navigate through the process diagram.

See Also: Using the Process Map Image for Navigation

This feature is browser dependent. FireFox is not supported.

Link

The Link tool can be used to create a new link between two activities. You can do this by specifying the two activities to be linked in the window that is displayed.

Wizards

The Wizards tool wizards allow you to create and define workflows in an easy way.

Help

The **Help** tool can be used to access help content for the Process Designer and its elements.

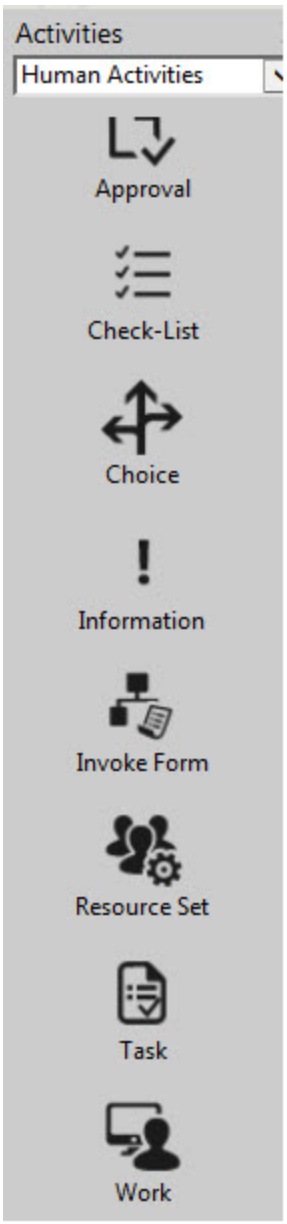
Activities

You can use the AVEVA Work Tasks activities area to access the different activities that can be used to design the workflow process in the Process Designer. There are several activities organized under different groups, which you can access from the drop-down list. These correspond to the standard workflow activities available with AVEVA Work Tasks.

Human Activities

When this option is selected from the drop-down list, it displays all the activities in this group. They can be used to enable human-workflow interaction. These activities include Approval, Check-List, Choice, Invoke Form, Resource Set, Information, Resource Set, Task, and Work.

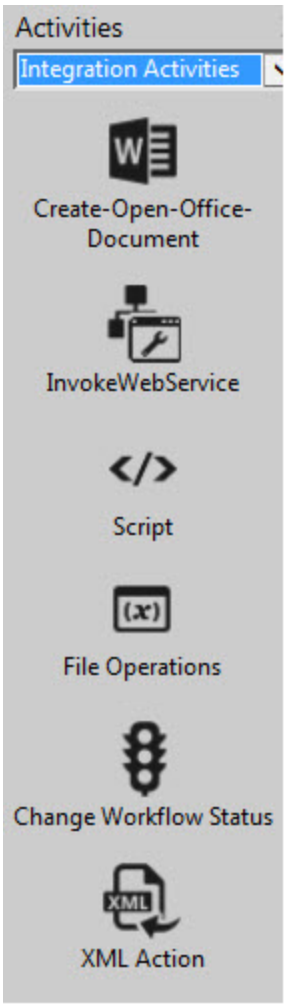
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Integration Activities

This option displays the activities that can be used to integrate workflows with external applications, scripts and databases. Activities include Create-Open-Office-Document, InvokeWebService, Script, File Operations, Change Workflow Status, and XML Action.

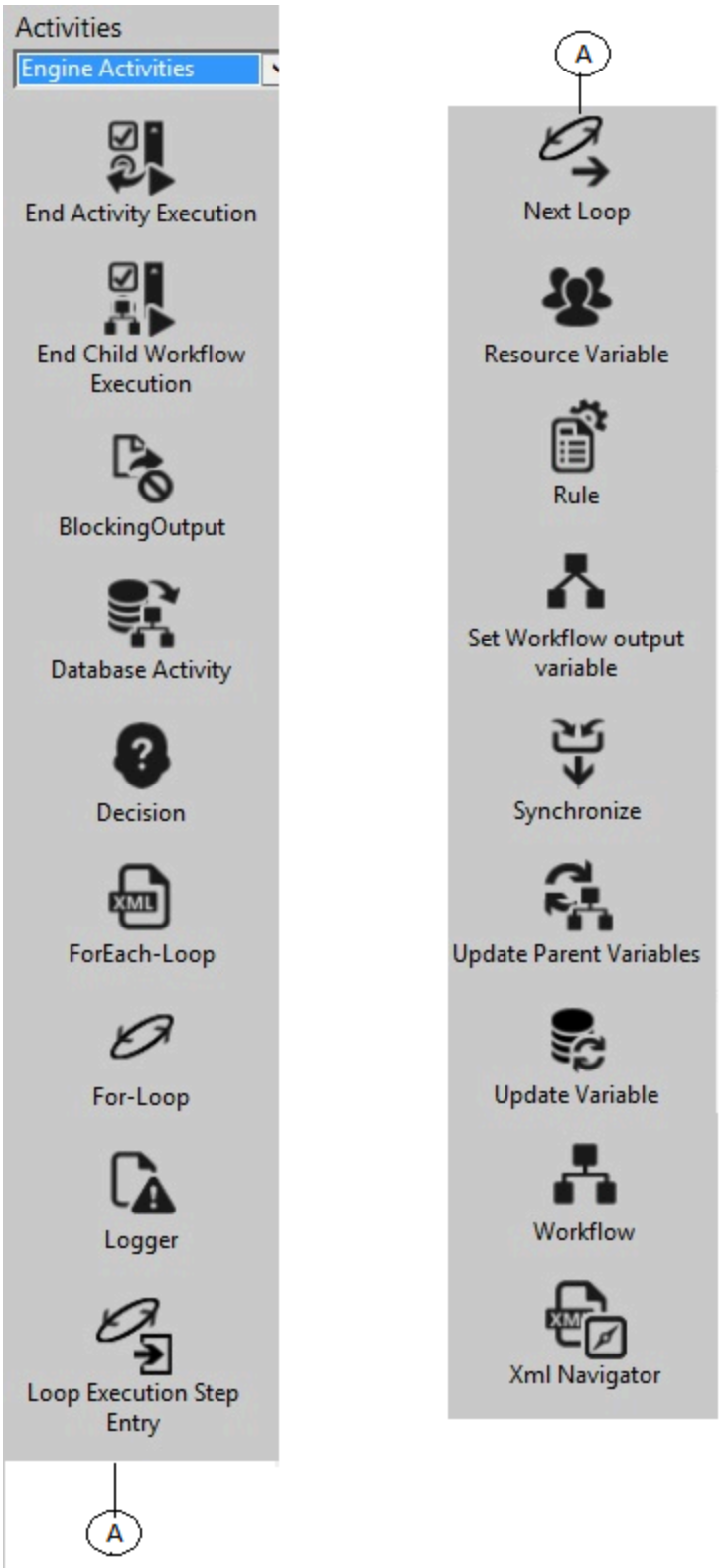
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Engine Activities

This option displays the workflow-engine related activities. These include BlockingOutput, Database Activity, Decision, ForEach-Loop, For-Loop, Logger, Next-Loop, Resource Variable, Rule, Set Workflow output variable, Synchronize, Update Parent Variables, Update Variable, Workflow, and Xml Navigator.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Scheduler Activities

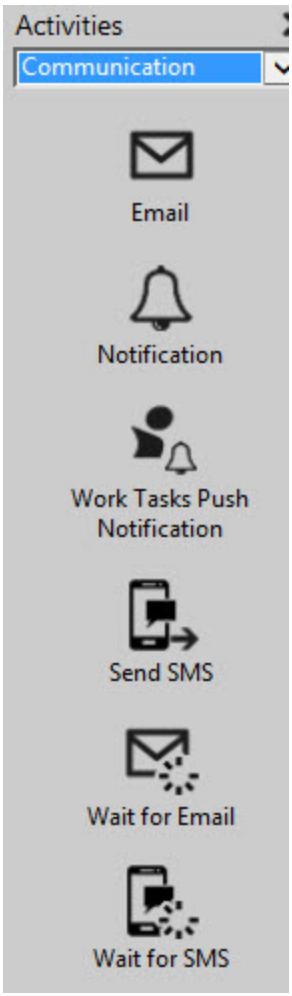
This option displays the activities that can be used to schedule the execution of other workflow activities. These include Control Custom Timer, Set Custom Timer, Scheduler, and Time Trigger.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Communication

This option displays the activities that can be used to enable communication from the workflows. These activities include Email, Notification, Send SMS, Wait for Email, and Wait for SMS.

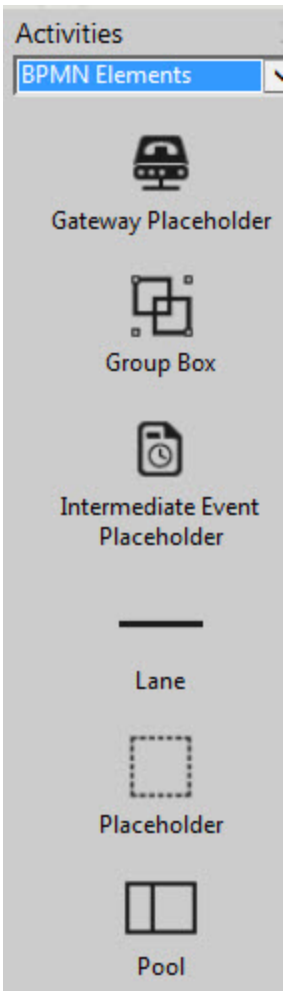
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



BPMN Elements

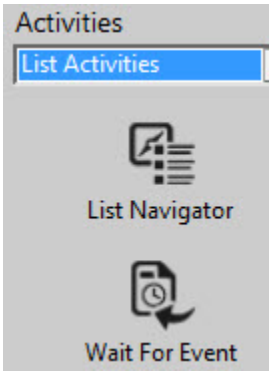
This option displays the different BPMN elements available in AVEVA Work Tasks. These activities include Gateway Placeholder, Group Box, Intermediate Event Placeholder, Lane, Placeholder, and Pool.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



List Activities

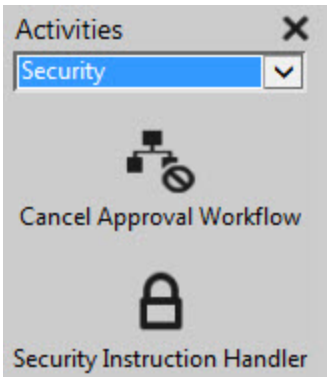
This option displays the activities that can be used to change the scope of the Xml List xml variable during workflow execution. These activities include List Navigator and Wait For Event.



Security

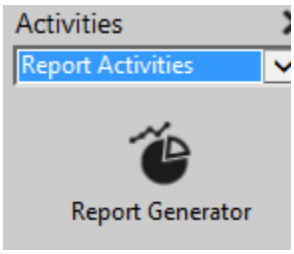
This option displays the activities that can be used for security. These include Cancel Approval Workflow and Security Instruction Handler.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



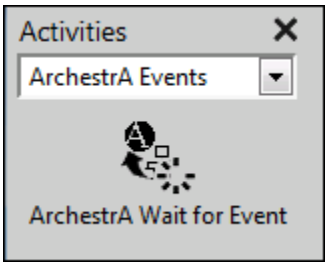
Report Activities

You use the Report Generator activity to run a report template and save the report in the Reports list.



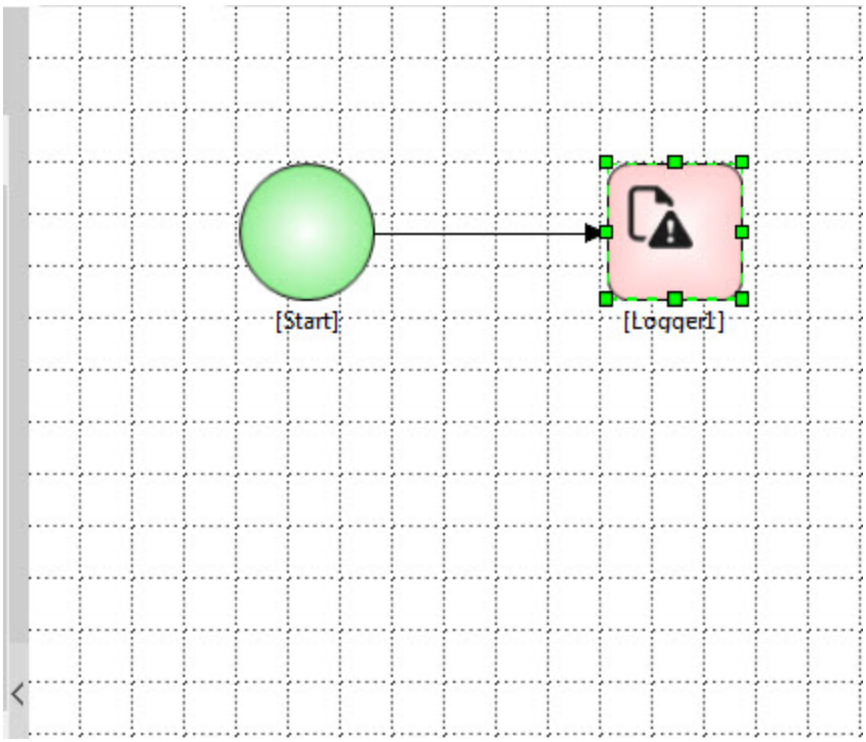
ArchestrA Events

Use the ArchestrA Wait for Event activity to wait for a number of events from ArchestrA Event Services platform for a particular time period, and then consume the events in a defined workflow. You can set the number of events and the time period in the properties of that activity. It can store the event context details in the specified XML variables.



Design Mosaic

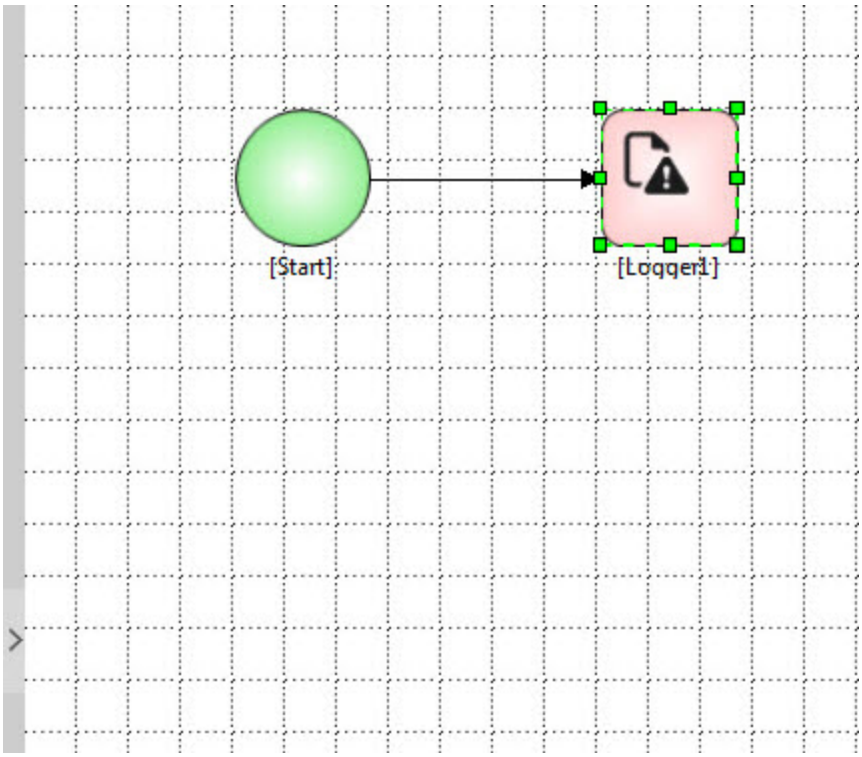
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



You can create and edit workflows in the Design Mosaic. Initially, it displays only the icon for the Start activity. You can click and drop different activities from the list of activities to the Design Mosaic. This will display the icons corresponding to the selected activities, on the Design Mosaic. These activity icons can be lined to one another.

A splitter has been provided on the left side of the Design Mosaic. You can click this splitter to hide the Activities pane and enlarge the Design Mosaic in that direction. The following image shows the Process Designer with the splitter clicked to enlarge the design area:

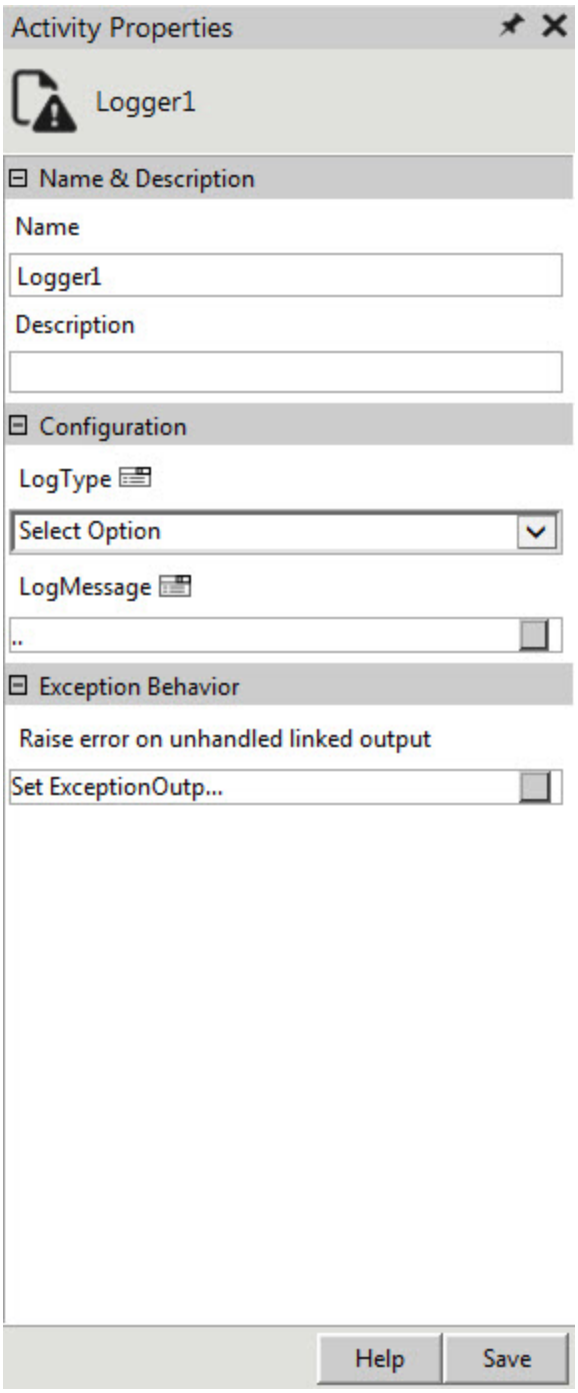
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Activity Properties

The **Activity Properties** pane at the right of the screen shows the property fields corresponding to the selected Activity. The following image shows the Activity Properties for the Logger activity:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

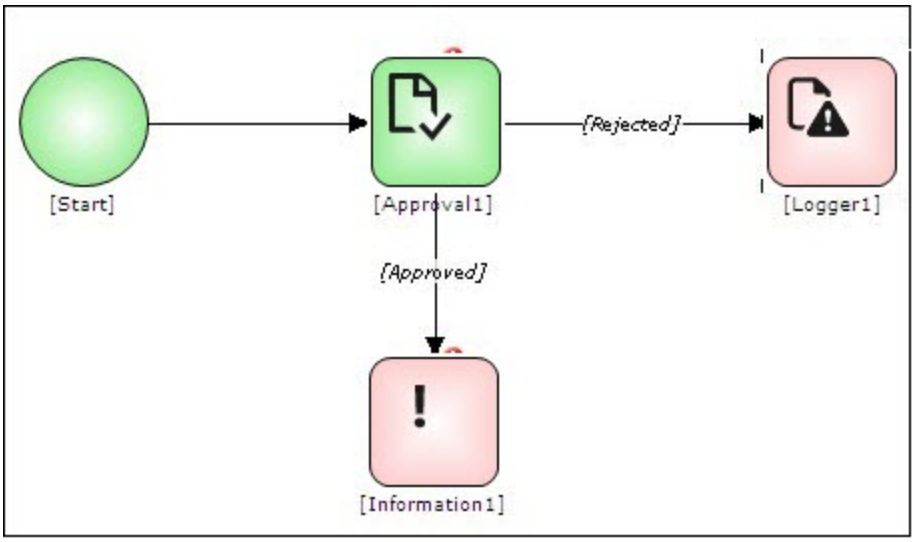


Save File as EMF Image

When you select the **Save As EMF Image** option from the File menu, the windows **File Download** dialog box appears. Click **Open** to open the image file or click **Save** to save the image file as an EMF image.

The following images shows the EMF image of process definition:

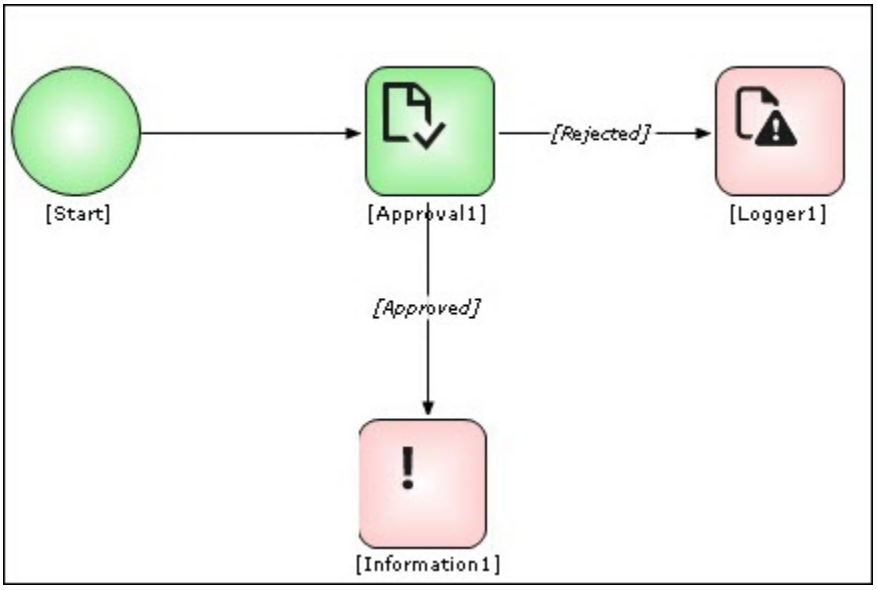
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Save File as PNG Image

When you select the **Save As PNG Image** option from the **File** menu, the windows **File Download** dialog box appears. Click **Open** to open the image file or click **Save** to save the image file as a PNG image.

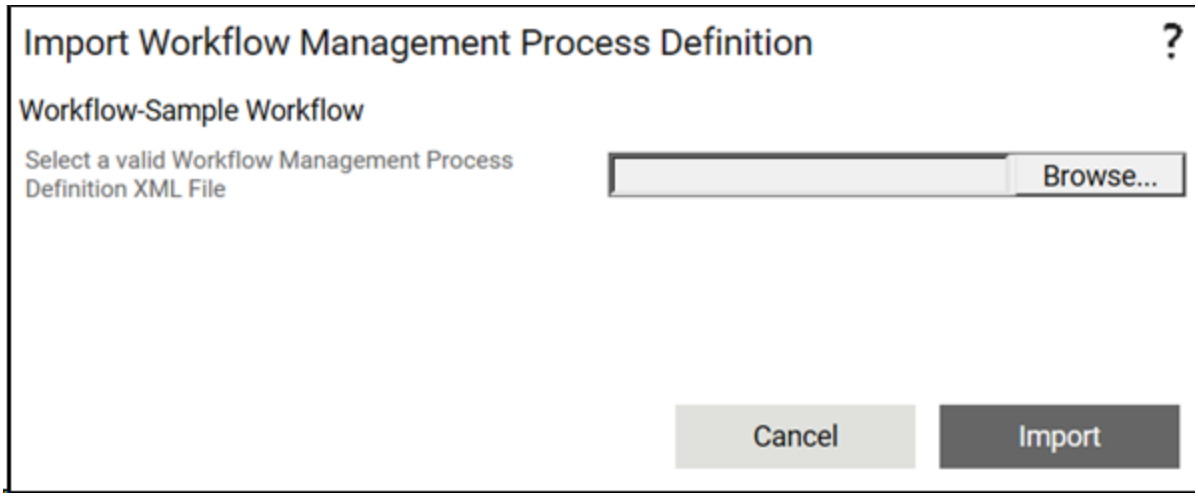
The following images shows the PNG image of process definition:



Import Process Definition XML File

This window is opened when you select the **Import** option from the **Tools** menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



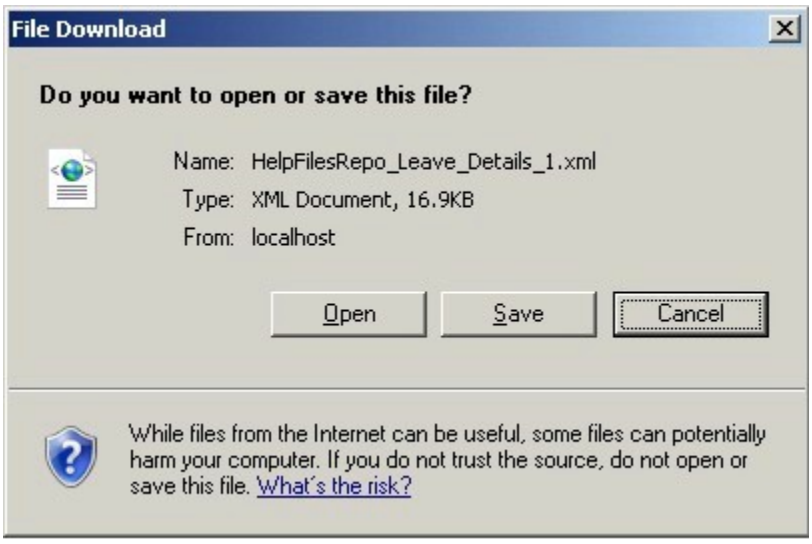
Import XML Process Definition window

You can use the following controls to import a valid process definition that has been saved as an XML file editing and/or execution.

- **Select a valid AVEVA Work Tasks Process Definition XML File** - You can enter the name of the XML Process definition file in this field or click **Browse** to select a valid process definition XML file.
- **Import** - After selecting a process definition XML file, you can click this button to import the file.
- **Cancel** - You can click this button to cancel the selection and close this window.

Export Process Definition XML file

The Windows File Download dialog opens when you select the **Export** option from the **Tools** menu.

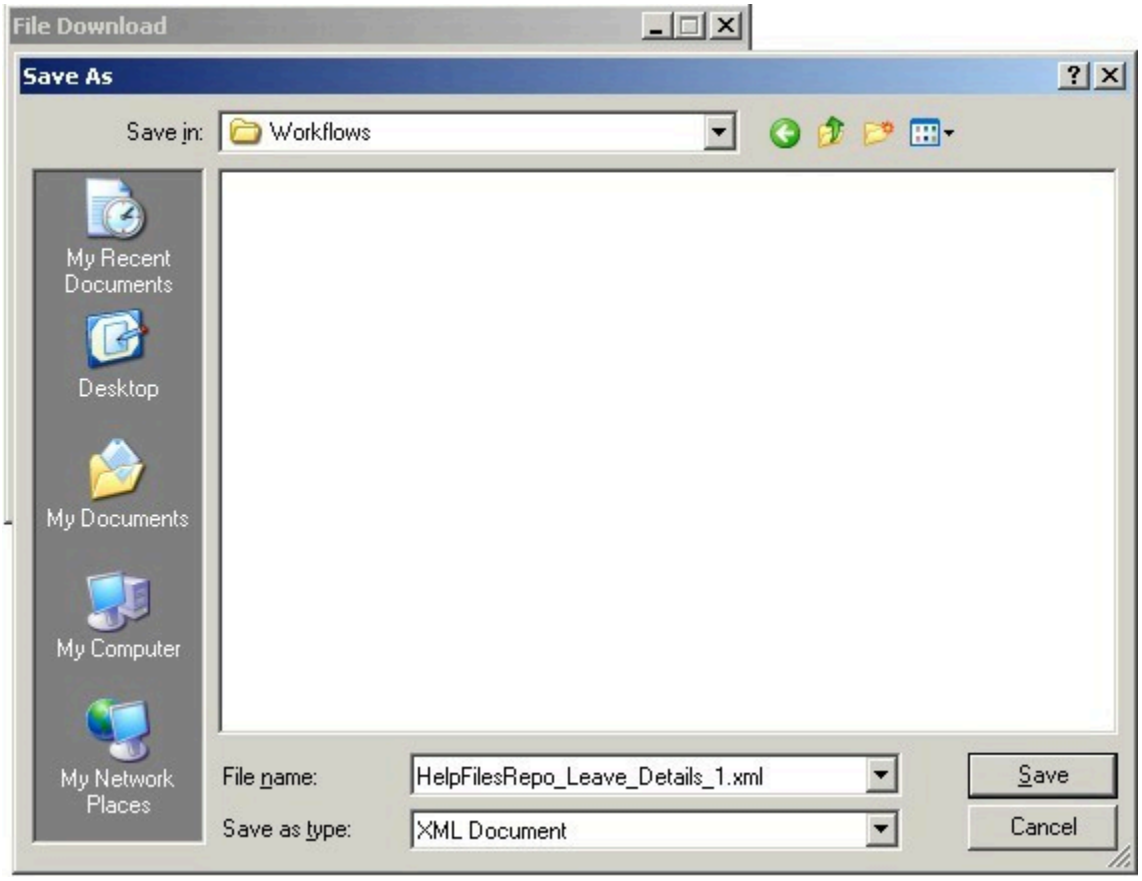


File Download dialog displayed for Process Definition Export

You can use the following controls in the dialog to export a AVEVA Work Tasks process definition as an XML file.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Open** - Click this button to open the XML file
- **Save** - Click this button to open the Windows Save As dialog to save the exported process definition XML file
- **Cancel** - Click this button to cancel the export and close this window



Save As dialog for saving the exported XML file

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

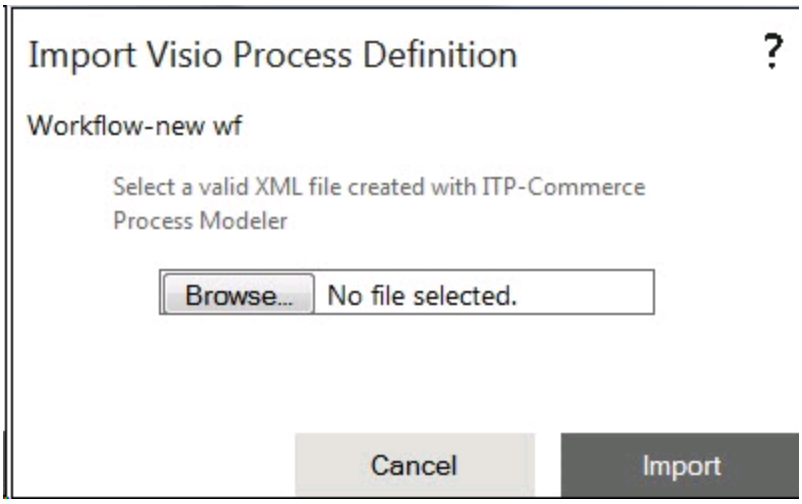
```
- <workflow name="" version="2">
  - <action name="Approval1" type="Approval" displayname="Approval" description="" looponoutput="">
    - <properties>
      - <property name="Action Display Name">
          <![CDATA[  ]]>
        </property>
      - <property name="Delivery Channels">
        - <propertychannel selectedchannelscount="1">
          - <channels>
            - <channel name="WebWorkList">
                <config url="" />
              - <forms>
                - <form name="Approval Form" type="NativeWebForm">
                    <actors grantaccess="True" />
                    <roles grantaccess="True" />
                  </form>
                </forms>
              </channel>
              <removedchannels list="" />
            </channels>
          </propertychannel>
        </property>
      - <property name="Work Item Fields">
          <Workitemfields />
        </property>
      - <property name="To">
        - <resource>
          - <globalfilter>
              <![CDATA[  ]]>
            </globalfilter>
          - <dispglobalfilter>
              <![CDATA[  ]]>
            </dispglobalfilter>
          - <filter>
              <![CDATA[ [UserID] = [Variable.SubmittedBy.UserID]  ]]>
            </filter>
```

Exported Process Definition XML file

Import Process Definition from Visio

This window opens when you select the **Import from Visio** option from the **Tools** menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Import Visio Process Definition window

You can use the following controls to import a process definition created with the ITP-Commerce Process Modeler for Visio.

- **Select a valid XML file created with ITP-Commerce Process Modeler** - You can enter the name of a valid XML Process definition file from the Visio stencil in this field or click **Browse** to select the XML file.
- **Import** - After selecting a Visio process definition XML file, you can click this button to import it.
- **Cancel** - You can click this button to cancel the selection and close this window.

  Note: The import routine maps Visio BPMN activities to native AVEVA Work Tasks activities. You should configure the necessary properties after import.

Add Work Item Fields

This window is displayed when you select **Work Item Fields** from the **Settings** menu of the Process Designer. It lists all the work item fields that have been added. The application data available in contents or variables can be displayed in the Work items list using work item fields. You can map the contents and variables with the work item fields in this window.

The 'Work Item Fields' property of the Human activities can be used to specify the work item fields to be used for a particular activity.

This field will automatically be displayed in the Work items list once the activity gets executed and the task is assigned to a resource. When the activity is executed, the values of the custom fields are persisted to the database (SKWork Item Fields table).

The **Work Item Field** window has the following buttons:

- **Add New** - You can click this button to add new work item fields. This will open the second window in the **Work Item Field** user interface where you can add the new work item fields.
- **Close** - Click this button to close the window.

Add/Set Work Item Field window

The **Add/Set Work Item Field** window is used for adding work item fields. This interface is displayed after the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

actor clicks **Add New** button in the Work Item Field window.

For each work item field, the following parameters can be set:

- **Name** - You can enter the name of the new work item field in this text box.
- **Display Name**- You can specify a display name for the new work item field in this text box. The field will be displayed to the actor with this label name in the Work items list.
- **Type** - You can specify the data type corresponding to this work item field by selecting an option from this drop down. The options available are "datetime", "decimal", "number" and "string". For datetime option, the supported format is DD/MM/YYYY.
- **Default Value Type** - You can specify the type of value corresponding to the new work item field by selecting an option from this drop down. The options available are "SetValue", "Variable", "Content", "Xml Variables" and "Value".
- **Default Value** - You can specify the value for the new work item field.
- **Application Level** - You can select this check box to configure the work item fields to be displayed at Application Level.
- **AVEVA Work Tasks Pro** - You can select this check box to configure AVEVA Work Tasks Pro grouping and view the work items in groups. The **AVEVA Work Tasks Pro** check box is enabled only when **Application Level** check box is enabled.

  AVEVA Work Tasks Pro scope can be enabled for only one work item field.

  Note: The value of the Work Item Field is displayed based on the AVEVA Work Tasks server language setting.

This window has the following buttons:

- **Add** - You can click this to add the new work item field.
- **Cancel** - Click this to cancel the settings for the new work item field and close the window.

When you click **Add**, the Work Item Field window opens with the new field added to it.

Alternate Actors

An actor is a user who is responsible for acting on a task or an activity. You can assign alternate actors to a workflow who will have rights to approve or reject an activity in the absence of the original actor.

You can configure an alternate actor in one of the following three different ways:

- Inbox
- Enterprise Console menu
- Process Designer

While configuring an alternate actor in Inbox or Enterprise Console menu, the Add Alternate Actor dialog box does not contain the **Activities** field. While configuring an alternate actor in Inbox, the logged in user is considered as the actor.

Steps to set an Alternate Actor in the Process Designer:

1. In the **Process Designer** window, point to **Settings**, and then click **Alternate Actors**.

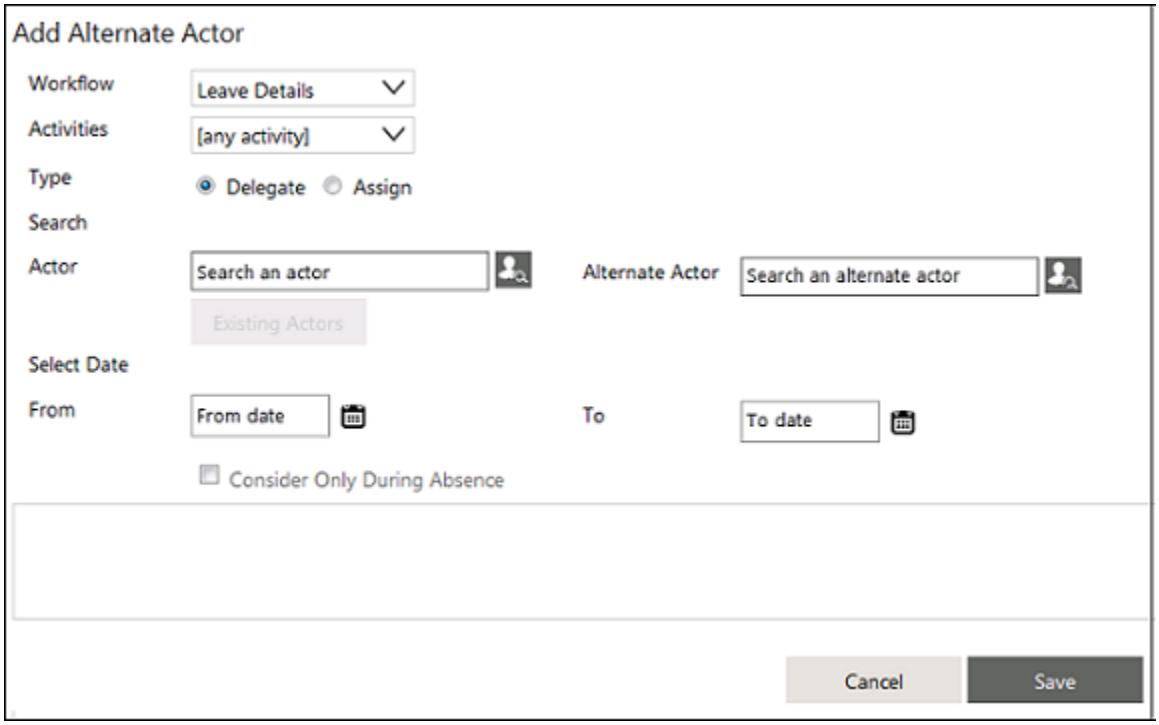
   The Alternate Actors dialog box appears.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

This window lists all the alternate actors that are added.

2. Click **Add New Actor** to add new alternate actors.

   The Add Alternate Actor dialog box appears.



3. Enter the required inputs for each work item field:
   a. **Workflow**: Specify the workflow for which the alternate actors are being added.
   b. **Activities**: Specify the activity for which the alternate actors are being added. The field will be displayed to the actor with this label name in the Work items list.
   c. **Type**: Specify the type of activity transfer from the original actor to the alternate actor.
      a. **Delegate**: Delegate the task to the alternate actor.
      b. **Assign**: Shift the task assignment to the alternate actor.

         Only one type of setting (**Delegate** or **Assign**) is allowed for an Actor, Alternate Actor, and Workflow combination.

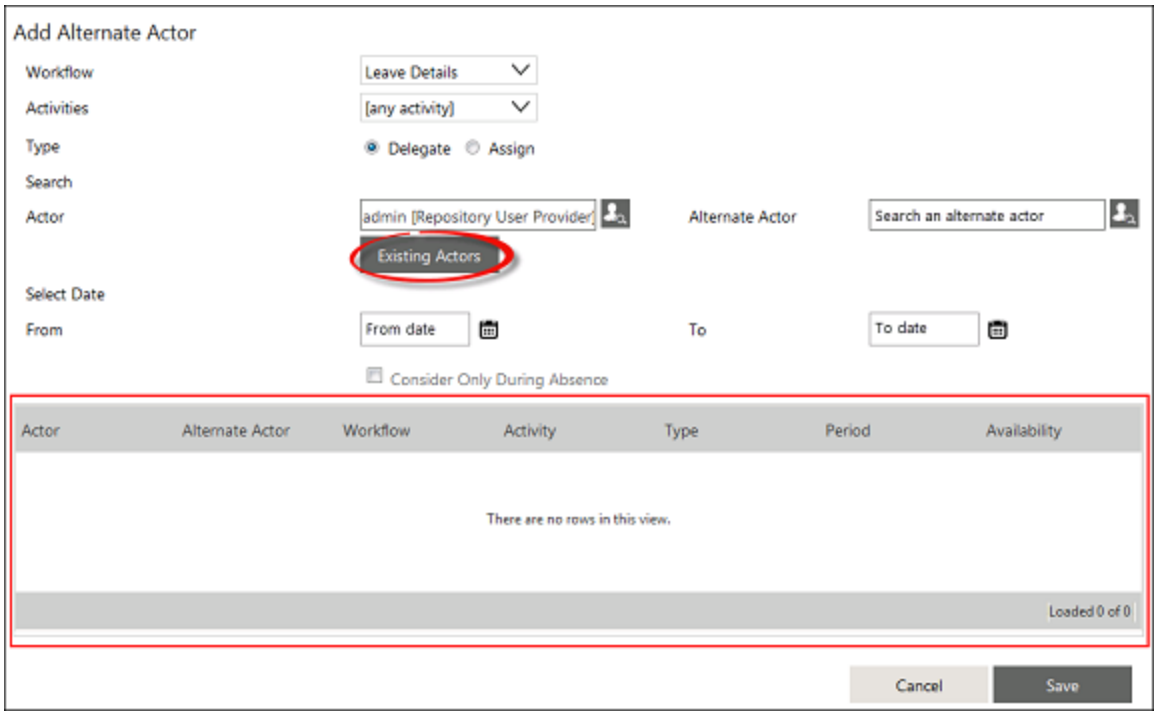
         In an upgraded environment where **Delegate** and **Assign** is already set for an actor, **Delegate** takes precedence over **Assign**. .
   d. **Search**: Specify the actor and the alternate actor.
      i. Click the **User Lookup** button to select the **Actor** or the **Alternate Actor.**

         The User Lookup dialog box appears.
      ii. Select the required users, and click **Update**.

         The same user cannot be set for both Actor and Alternate Actor.

         Click **Existing Actors** to view the list of all the existing alternate actors of the selected actor. The list of existing actors and the associated details get populated in the grid at the bottom section of the dialog box as shown in the following image:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



e. **Select Date**: Specify the period for which the alternate actor is being assigned to the original actor.
   i. **From**: Click the calendar to select the start date.
   ii. **To**: Click the calendar to select the end date.
   iii. **Consider Only During Absence**: Select to specify that the alternate actor is to be considered only during the absence of the original actor.

   Note that the **From** date cannot be older than the current date.

4. Click **Save** to add the new alternate actor.

   The **Alternate Actors** window opens showing the newly added alternate actor.

For more information about assigning Alternate Actors from the Work items list window, see *Work items list Control*.

For more information about how to set an Actor to perform a human task, see *Human Activities* in the *Developer Guide*.

See Also: *Alternate Actors* topic in *User Guide*.

BAM Configuration

The Business Activity Monitoring Configuration enables business analysts to setup critical performance measurement criteria for a workflow.

Performance criteria can be setup to provide the following information:

- Overdue Analysis
- Performance Analysis

The BAM Configuration window has the following five tabs:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Overdue Analysis**: Use this tab to setup thresholds for total pending workflows and pending in different time slots.
- **KPI Data Points**: Use this tab to set up KPI data points such as activities pending, workflow status, and other measurement points.
- **KPI(s)**: Use this tab to set up threshold limits for each data point.
- **Performance**: Use this tab to specify work types that must be measured for performance analysis.
- **Alert Configuration**: Use this tab to create rules for both Overdue and KPI based on the threshold value set, set the time for the alert to be triggered, and set an action, if the rule is satisfied.
- **Basic Configuration**: Use this tab to schedule the analysis. You can also select Process Now for immediate reporting.

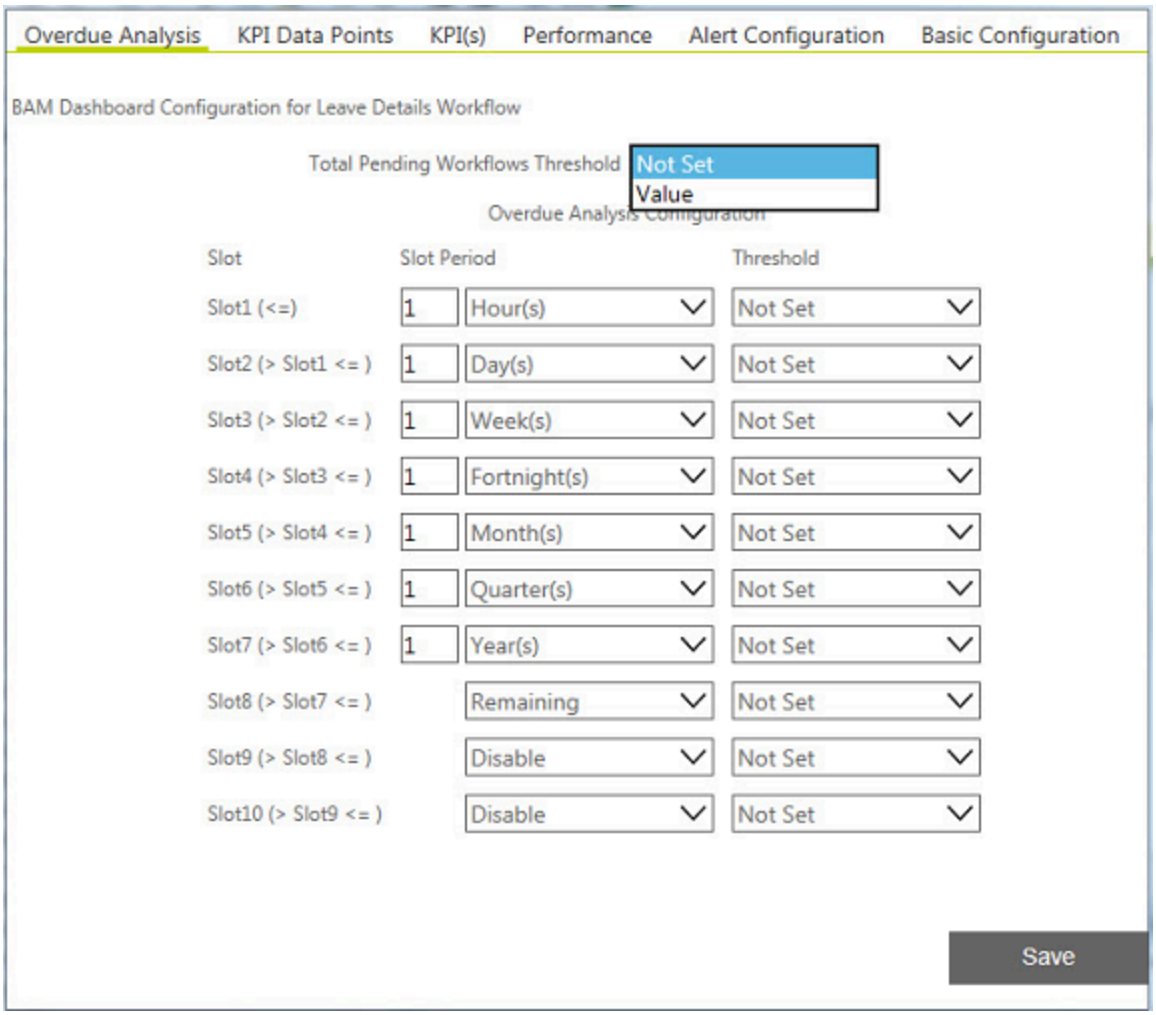
Steps to Configure BAM

1. Define the schedules at which the analysis on workflow execution must be performed.
2. Configure Overdue Analysis:
   a. Define threshold limits for all pending workflows for Overdue Analysis.
   b. Define data points for Key Performance Indicators to measure.
   c. Define threshold levels for each KPI at different time slots.
3. Configure Performance Analysis:
   a. Define type of work to be measured, the type of work to be analyzed .
   b. Identify work item fields to compute averages for comparative analysis .
4. Use the BAM Dashboard to view the Overdue Analysis.
5. Use the BAM dashboard to analyze Performance. You can also set benchmarks for a resource or an activity to measure resource performance against the benchmark.

Overdue Analysis

You can use the **Overdue Analysis** tab to setup the threshold for total pending workflow instances at any time and for selected time slots.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Total Pending Workflows Threshold: Select **Value** to set a limit on the total workflows that may be pending at any time.

Overdue Analysis Configuration

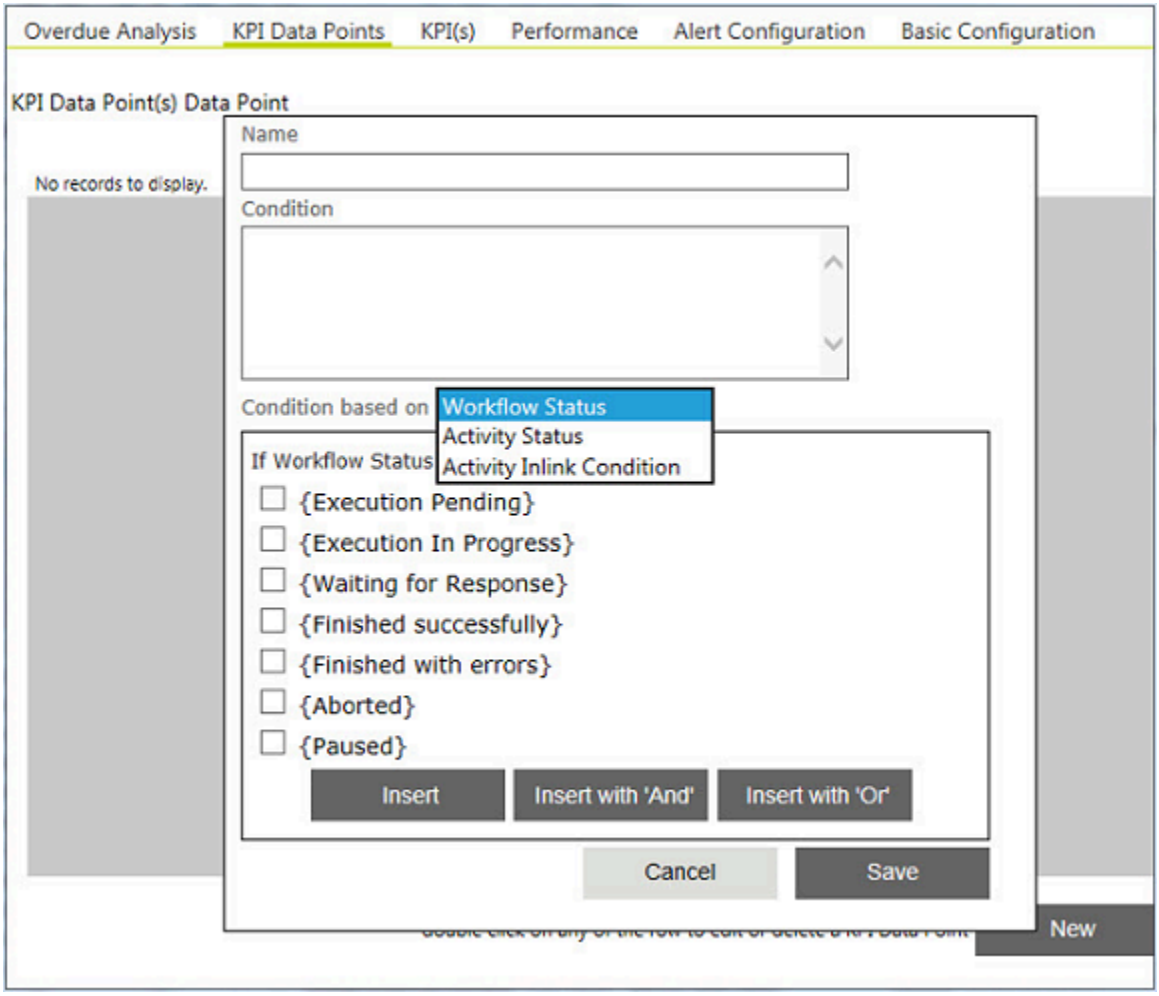
Overdue workflows may be analyzed in 10 time slots.

- *Slot Period*: Select the time period for defining threshold values in each slot.
- *Threshold*: Select the threshold option.
  - Not Set: This indicates there is no threshold.
  - Value: Select this option and enter a threshold value for the number of pending workflows in the time slot.
  - % of Total: Select this option to enter the threshold of pending workflows as a percentage of total active workflows in the time slot.
- Click **Save**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

KPI Data Points

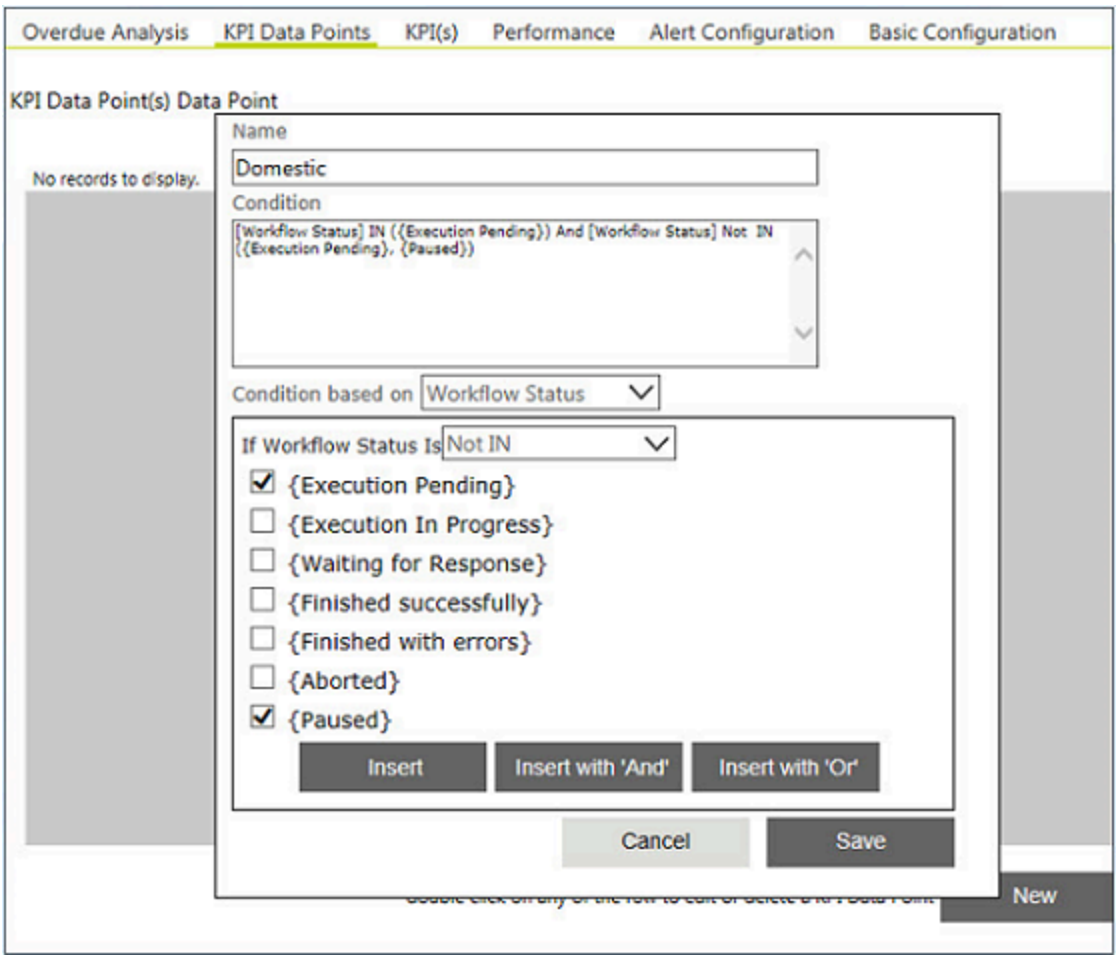
You can use the KPI Data Points to define the data points that you want to monitor for overdue analysis. These data points are monitored in addition to the thresholds set for the overall pending workflows in the Overdue Analysis tab.



KPI Data points may be defined for conditions that include one or more of the following:

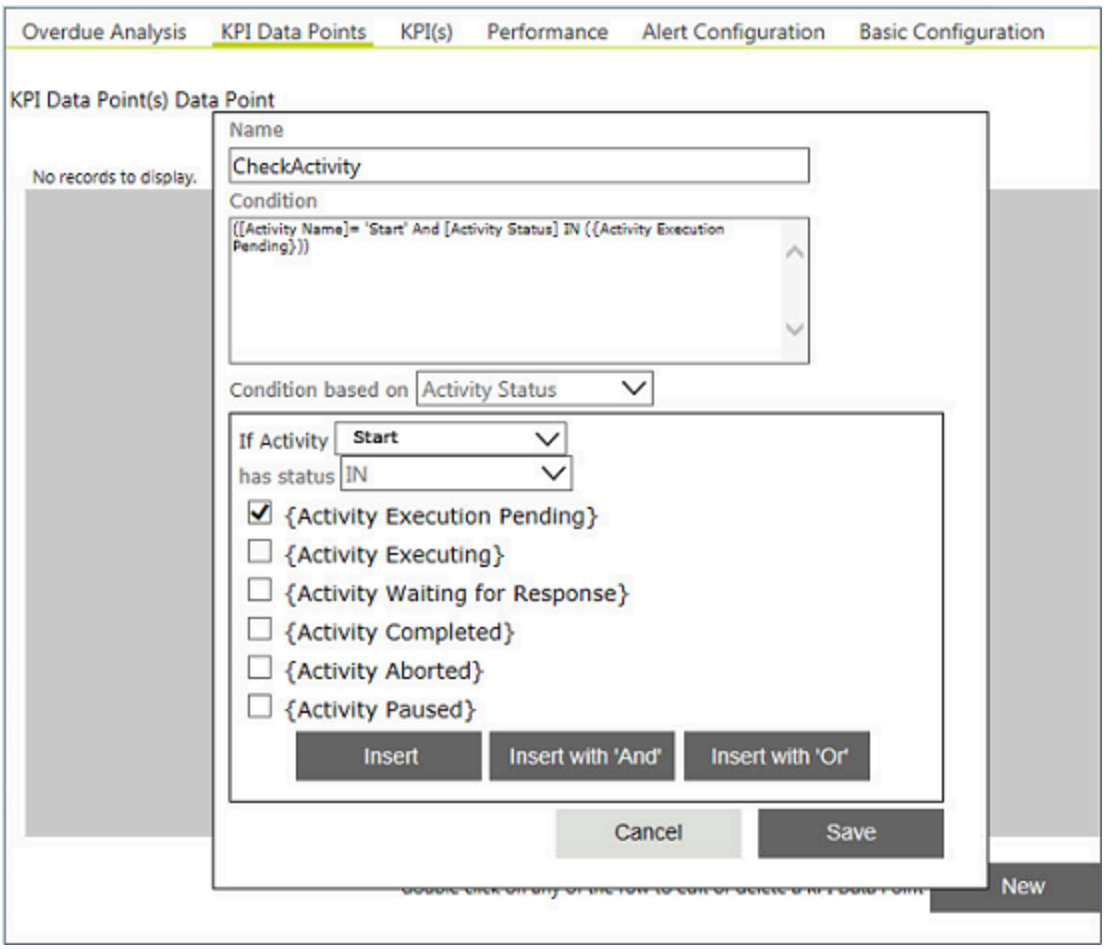
Workflow status

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Activity status

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Activity Inlink Condition where the link to the activity has a specified condition.

Example

In a customer support workflow, you would like to monitor all workflows that have status as Workflow Pending and in which the Close Support Ticket activity has the status Activity Execution Pending and the result of the previous activity Get Customer Feedback is not No Feedback Received from Customer.

Setting KPI Data Points

1. Click **New**. A dialog box appears.
2. Enter the following information:
   a. **Name**: Name for the Data Point.
   b. **Condition based on**: Select the condition type from **Workflow Status**, **Activity Status**, or **Activity Inlink Condition**.
3. Select the condition for the **Workflow Status**, **Activity Status**, or **Activity Inlink Condition** selected.
4. Click **Insert**.
5. Select another condition type, if required.
6. Click **Insert with 'And'** or **Insert with 'Or'** to add the new condition with and/or to the existing condition.
7. Click **Save**.

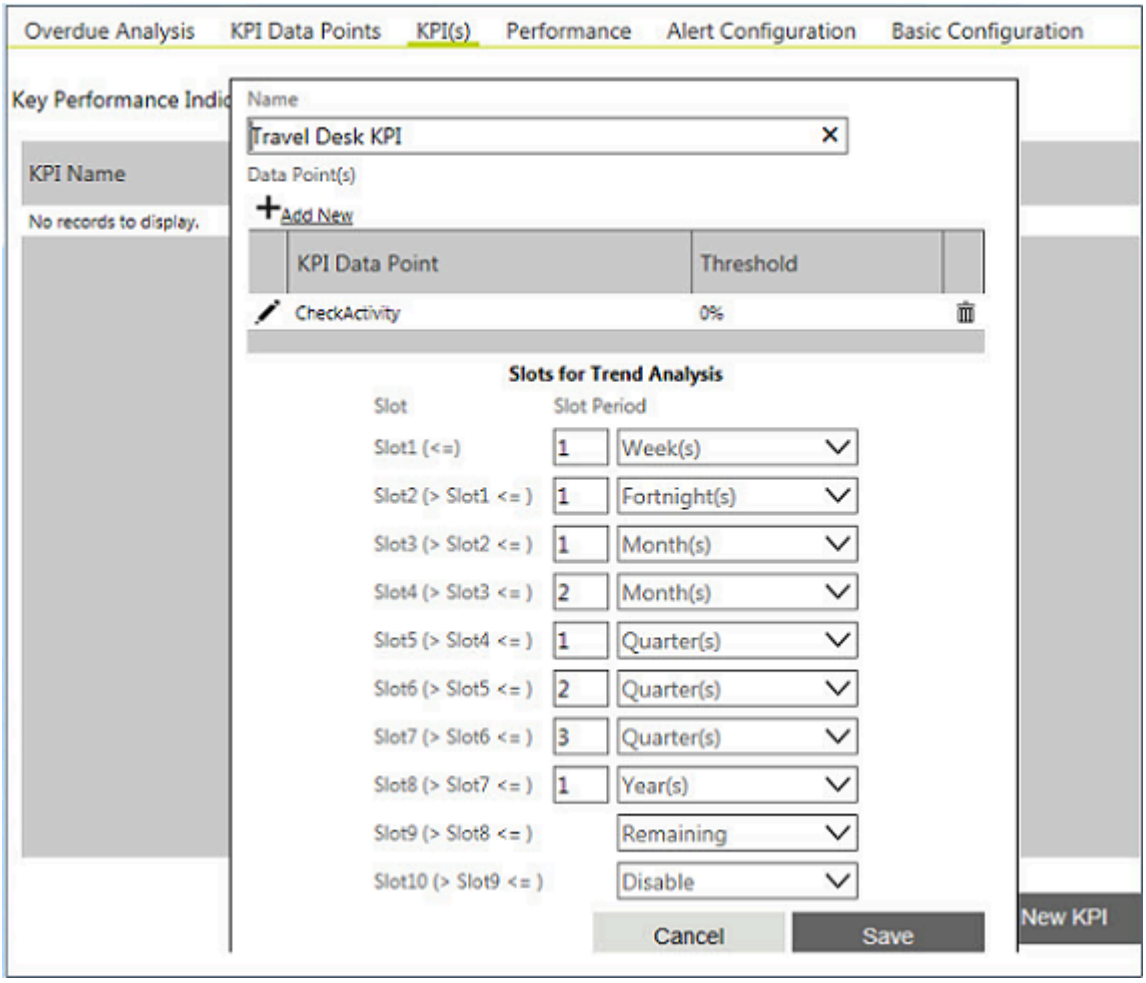
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

KPI(s)

You can use the KPI(s) tab to define thresholds for the KPI data points identified in the previous tab.

To add a KPI:

1. Click **New KPI**. A dialog box appears.
2. Enter the following information:
   a. **Name**: Name for the new KPI.
   b. **Data Point(s)**: Click **Add New** to add a Data Point, and enter the overall Threshold for it.



3. Slots for Trend Analysis: Select the time periods for trend analysis.
4. Click **Save**.

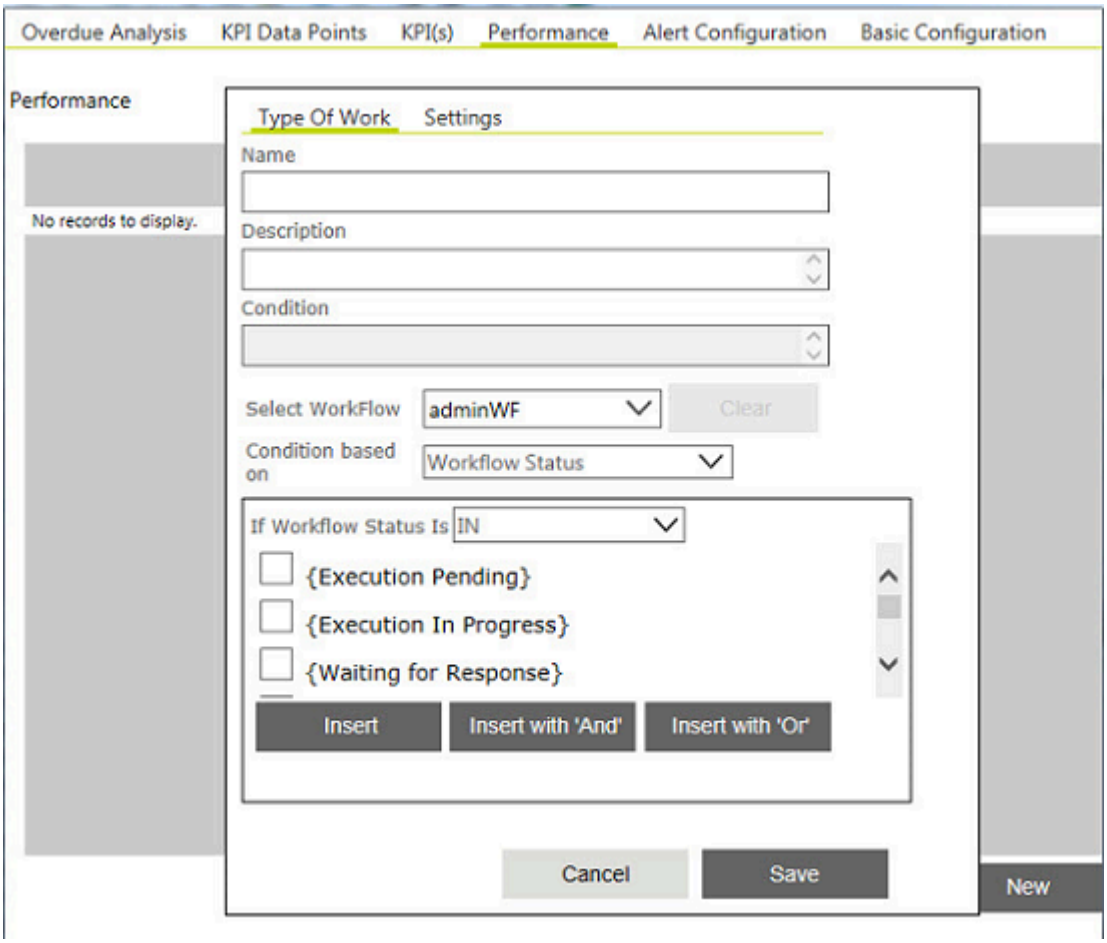
Performance

You can use the Performance Analysis configuration tab to define criteria for work to be monitored for resource analysis, work, and resource performance.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Steps to configure Performance Analysis:

1. Define **type of work** to be monitored. To add a new work type, click New Type of Work.
2. Enter the name for the type of work.
3. Setup conditions for monitoring the work. Work types can be configured to include conditions that comprise workflow status, activity status, activity in link conditions, work item fields and queues.
4. The options displayed below the 'Condition based on' field depend on the condition type.
5. Click **Insert with And** or **Insert with Or** to build the condition further.



6. In Settings tab, select **the Work Item** fields to be monitored. All work item fields in the activities and workflows selected in **Type of Work are available for selection**.
7. Select **Consider Acknowledgment Date for Work Item** to consider the time for the work item from the acknowledgment date and not on assignment date.
8. Select a date range for measuring performance. Select Apply date range for weighted average to use the same period for calculating weighted average. A resource performance is compared with the weighted average and slotted accordingly.
9. Select the date range for calculating the weighted average of the performance for this work type.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Alert Configuration

Use this tab to create rules for both Overdue and KPI based on the threshold value set, set the time for the alert to be triggered, and set an action, if the rule is satisfied.

For more information about Alert Configuration, see BAM Alert Configuration Settings.

Basic Configuration

You use the **Basic Configuration** tab to define the schedules at which the analysis must be run. You can also run the BAM reports for the workflow immediately with the **Process Now** option available in this window.

You can process the BAM Reports in one of the following two ways:

- Manual proce**ss**
- Scheduled process

In the manual process, click the **Process Now** button to synchronize the report data. You can achieve similar output by scheduling the process through the BAM Scheduler.

BAM Web Services Configuration

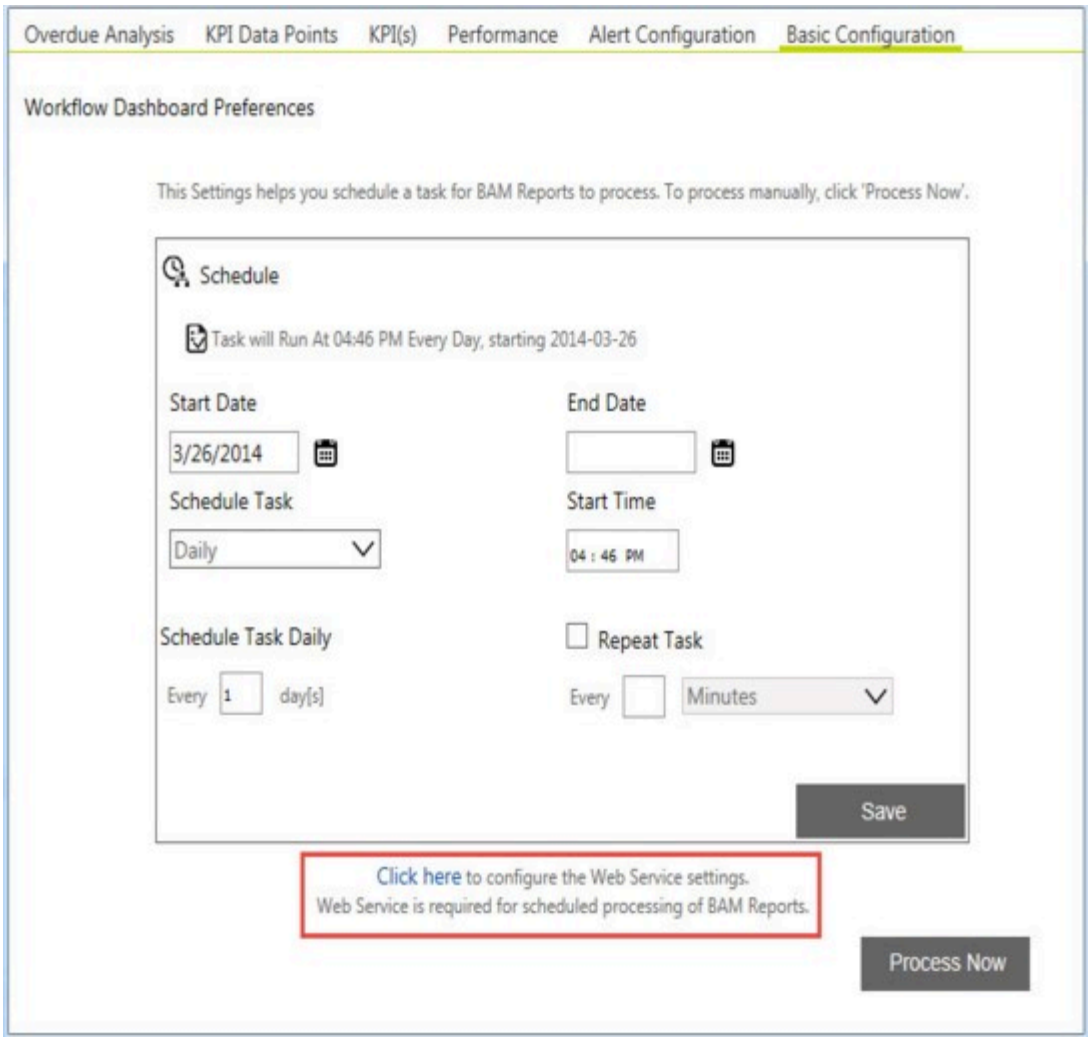
Configure the web services for scheduling the BAM Process using which you can send and email with the embedded graph. Use the BAM Configuration page for this configuration. DO the following to access the configuration page:

1. Right-click on any workflow, and click **BAM Configuration.**

   The BAM Configuration page appears.

   - Alternatively, go to **Process Designer,** click **Settings**, and then click **BAM Configuration.**
2. Click the **Basic Configuration** tab.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

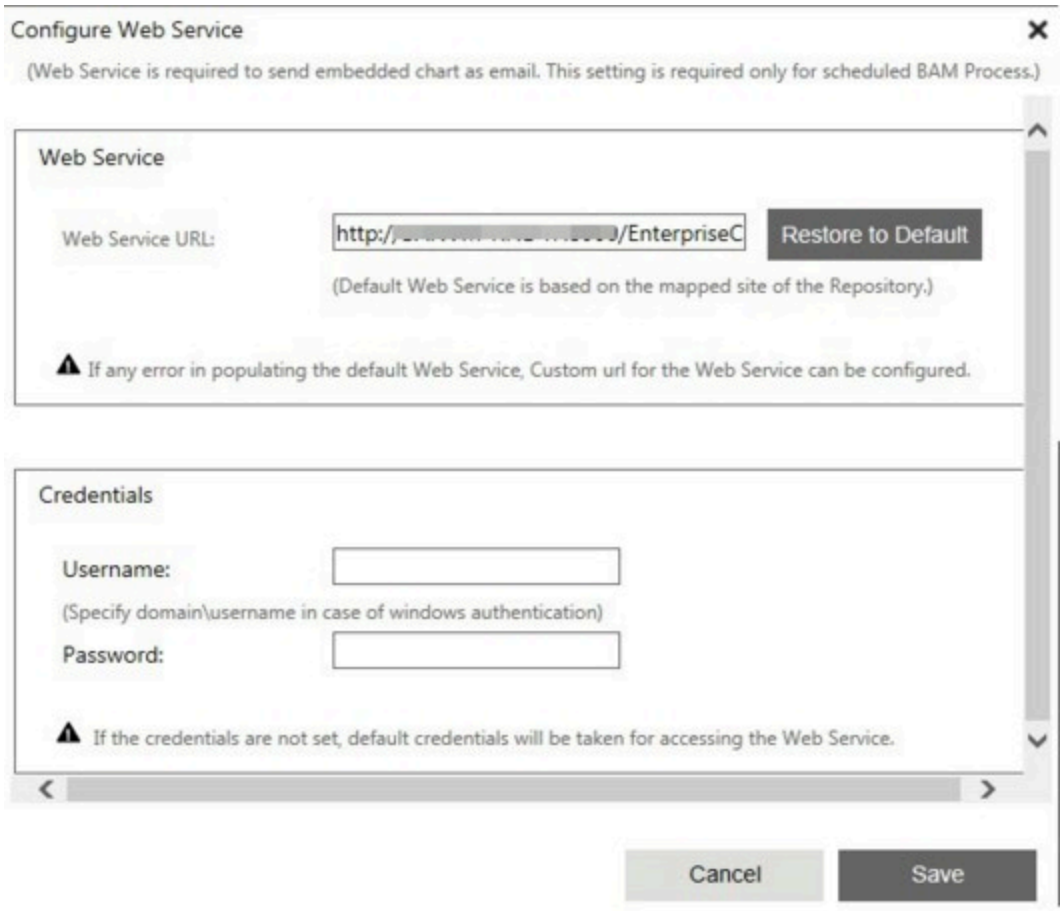


3. Click the **Click here** link.

   The Configure Web Service section appears.

   The Configuration Web Service section contains the default web service URL in the **Web Service URL** field. The default web service URL is based on the mapped repository.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



You can change the default Web Service by entering the Web Service URL in the **Web Service URL** field. Enter a valid and accessible Web Service URL. Enter the username and password as the credentials for the Web Service. In case of Windows authentication, specify the user name with the domain name. For example domain\username.

Click **Restore to Default** to restore the default URL.

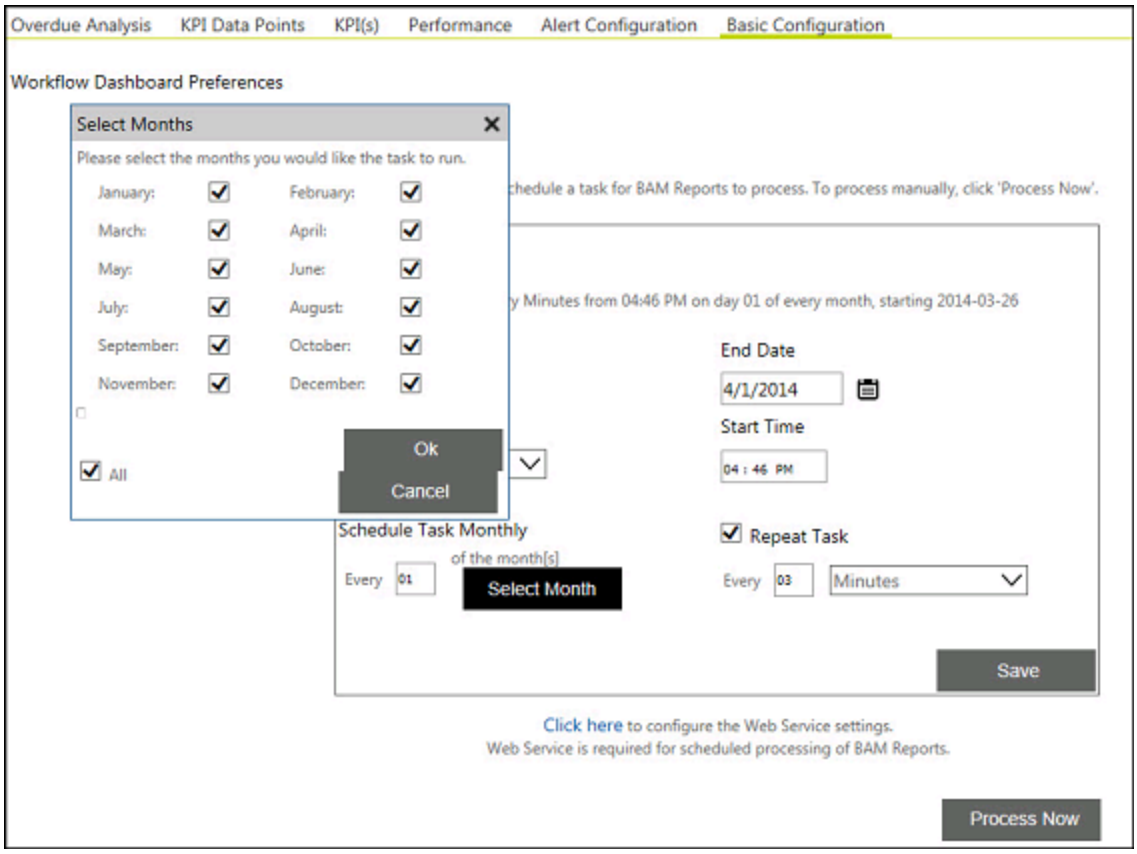
In case of any difficulty in populating the Web Service, create a Web Service with the *ProcessBAMService.asmx* specified in the following path:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\Web\BPMUITemplates\Default\BAM\WS\ProcessBAMService.asmx.

To schedule the analysis:

1. Define the start and end dates. To run the reports indefinitely, leave the end date empty.
2. Select the **Schedule Task**. Tasks can be scheduled on daily, weekly or monthly basis.
3. Select the months, days or weeks in which the analysis task must run.
4. Select the day of the month or week or the hour of day when the task must be run.
5. Click **Save**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

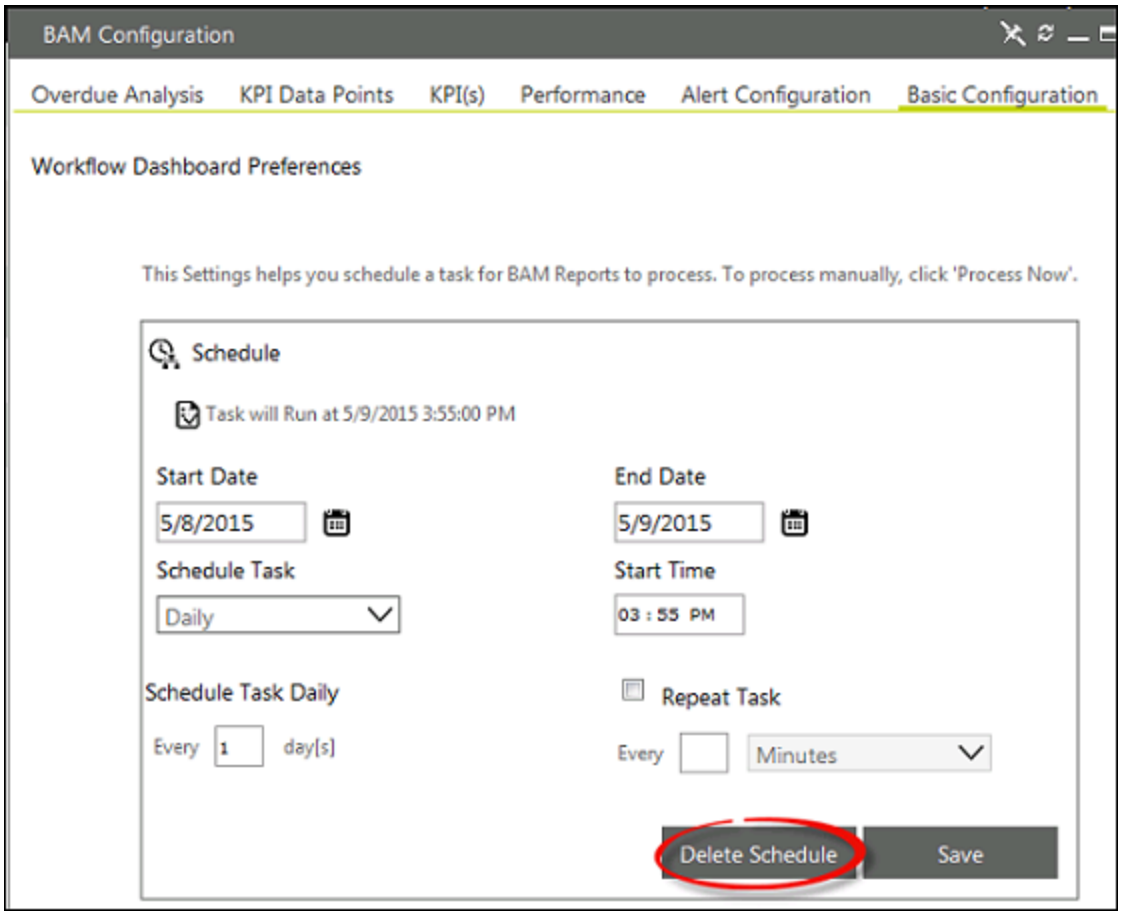


6. To delete a BAM schedule, click the **Delete Schedule** button.

   The schedule is deleted, and the respective database entries are removed.

   The **Delete Schedule** button appears only after you save a schedule. If you do not see the **Delete Schedule** button even after saving the schedule, refresh the page, or close the page and open the page again.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Overdue Analysis Report

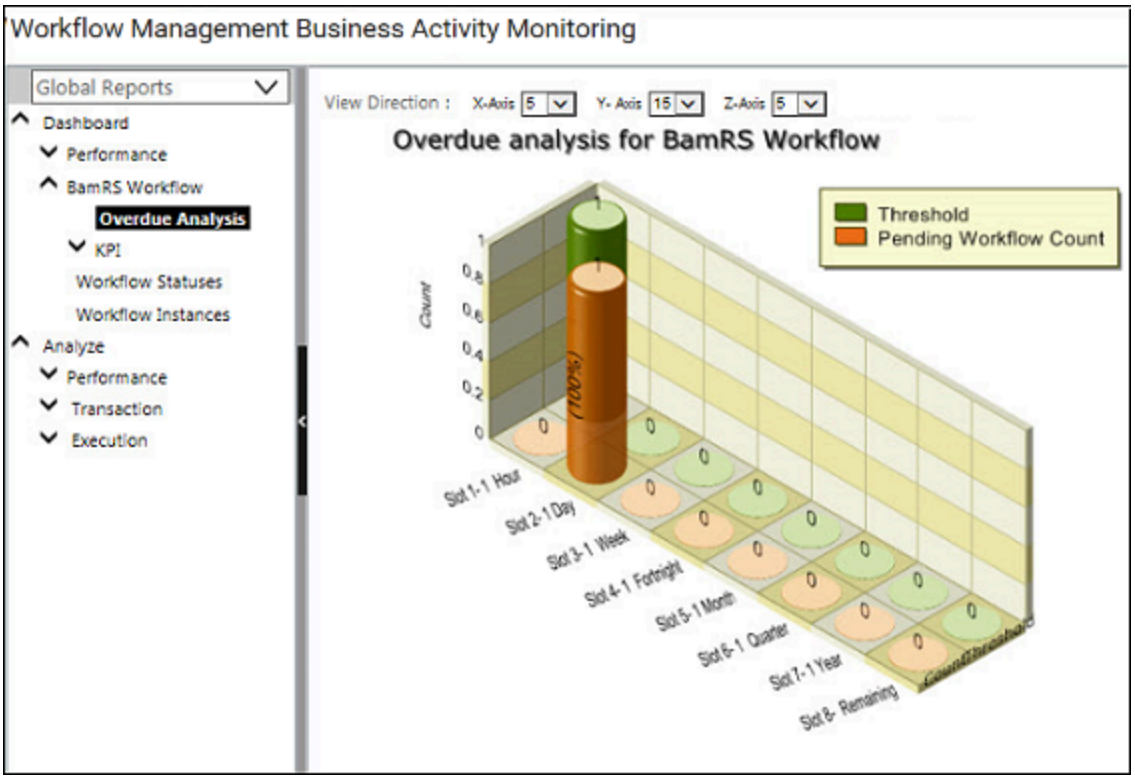
The Overdue Analysis Report in the BAM Dashboard is displayed at two levels:

- KPI(s): The trend analysis for each KPI defined in the BAM Configuration is provided as a pie chart and bar graph.
- Overdue Analysis: The Overdue Analysis report shows a graphical view of the pending workflows against the threshold.

Overdue Analysis

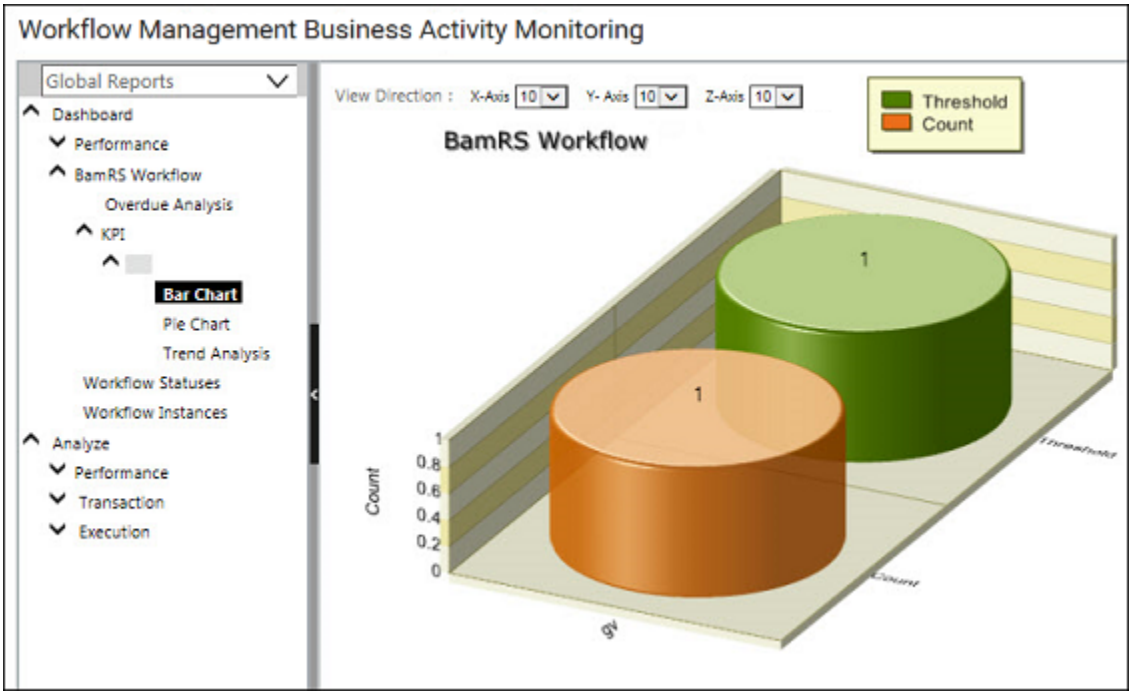
This graphical view shows how the pending workflow count stacks up against the prescribed thresholds in each time slot as defined.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



KPI Analysis

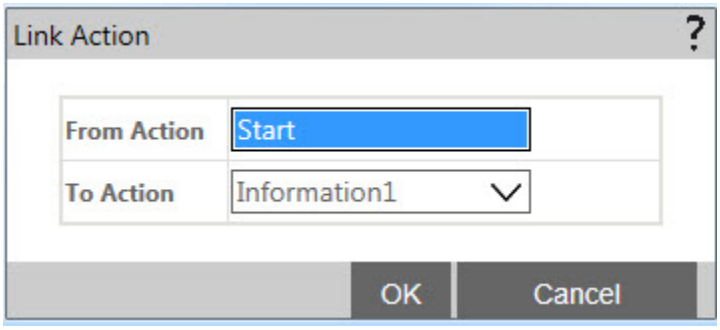
The bar chart view shows the KPI against the threshold defined for it.



© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Link Actions

This window is opened when you click the Link tool in the toolbar.



Link Actions window

You can use the following controls to link two activities in the design mosaic:

- **From Action** - Specify the first action to link by selecting it from this drop-down
- **To Action** - Specify the second action to link by selecting it from this drop-down
- **OK** - After selecting the two activities, click this button to link them
- **Cancel** - Click this button to cancel the selection and close this window

Process Designer File Operations

AVEVA Work Tasks allows you to create new process designs with existing diagrams, clear process diagrams, and save them as images.

Reuse Existing Process Designs by Renaming Them

AVEVA Work Tasks allows you to use an existing process diagram as the basis for creating new process designs. In scenarios where there are a number of similar process designs, this helps reduce the time required to design all the processes. It will be enough to design one process and reuse its diagram for creating all the other process diagrams.

To reuse an existing process diagram

1. Open the process design file to be reused.
2. From the **File** menu, select the **Rename File** option.
3. Specify the new name for the process design file in the window that opens.
4. Proceed to modify the process diagram as required and save the modified process diagram.

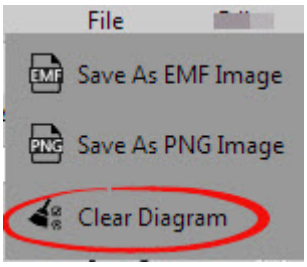
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Clear the Process Diagram

In AVEVA Work Tasks you can clear the entire process diagram. You do not have to delete the diagram activity-by-activity. This is very useful when you are modeling the process. You may have to redo the process diagram many times until you arrive at the correct process model.

To clear an existing process design

1. From the **File** menu, select the **Clear Diagram** option.



2. Proceed to create a fresh process diagram from scratch.

Save the Process Diagram as Image

AVEVA Work Tasks allows you to save the process diagrams as images in different formats from the menu. In the earlier versions, you could right-click on the process diagram and save it as a BMP image.

- You can select **Save As EMF** Image to save the diagram as an image in the EMF (Enhanced Meta File) format. This is a vector format, which is ideal for printing or embedding in MS Office applications.
- You can select **Save As PNG Image** to save the diagram as a PNG image. This is useful when you need to capture the process diagram images for use on the web.

See also, Save As EMF Image and Save As PNG Image.

Process Designer Edit Operations

The following editing actions are allowed in the Process Designer:

- Copy and Paste
- Undo and Redo
- Set Properties

Copy and Paste Activities

In the **Process Designer,** you can copy and paste activities of AVEVA Work Tasks. You can use this feature when you are designing large process diagrams.

If the same activity occurs at many places in the process design, you can copy the first instance and paste it at the new location. When you copy an activity all the configured property settings are also copied. However the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

links between the activities are not copied. You can modify the settings as required.

You can copy and paste activities within the same workflow in all the supported browsers.

When you copy and paste BPMN elements, only the BPMN elements are copied and pasted, and not the activities enclosed within the BPMN elements.

To copy and paste activities

1. Click on the activity to be copied to select it.
2. From the **Edit** menu, select **Copy**.
3. Position the mouse cursor at the location where you wish to place the copy.
4. From the **Edit** menu, select **Paste**.

Alternatively, you can do the following

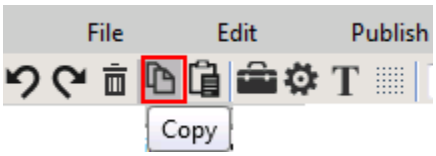
1. Right-click the activity and select **Copy** from the menu.
2. Right-click at the location where you wish to place the copy.
3. From the right-click menu, select **Paste**.

You can also use the copy and pastes icons in the toolbar to copy and paste the activities.

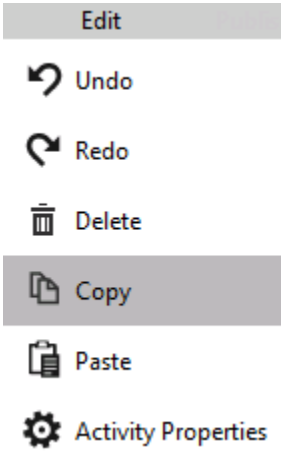
You can select to copy and paste multiple activities together. To copy and paste multiple activities at the same time use the copy and paste options in the Edit menu or in the toolbar.

The following images shows different copy options in the Process Designer:

Copy option in the toolbar:

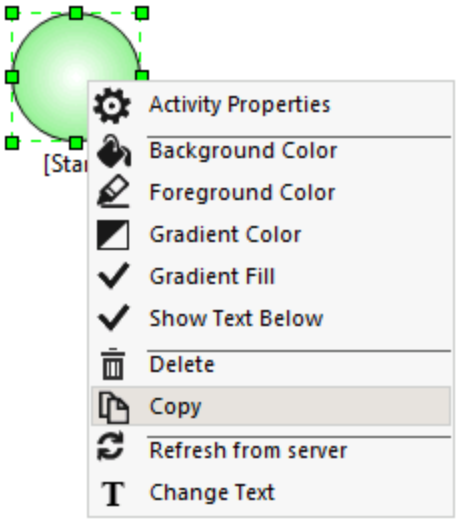


Copy option in the menu:



Copy option in the context sensitive menu:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Undo and Redo Actions

The Process Designer is very user-friendly and simplifies the task of designing of process diagrams. The Undo and Redo options are one of the many new features that are helpful to the designers.

- To undo any action, select **Undo** from the Edit menu. For example, you can undo an accidental deletion of an activity.
- To redo any action, select **Redo** from the Edit menu. For example, you may need to add two Information activities after an Approval activity to inform the user submitting a request, about its approval or rejection. After adding the first Information activity you can select Redo to add the second Information activity.

Set Properties in the Activity Properties Window

AVEVA Work Tasks allows you to open the Activity Properties area to set the properties. You can open the Activity Properties in different ways. One of these is the Activity Properties option in the Edit menu.

Also see Method 2: Select activity and then select Activity Properties from Edit menu

Import and Export Functions in Process Designer

You can import and export process designs saved as valid XML files.

You can also import a process definition created with the ITP-Commerce Process Modeler for Visio.

Import from XML

You can import process designs saved as valid XML files. This can be used to import and modify a process design to create a new one.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To import a process definition XML file

1. From the **Tools** menu, select the **Import** option.
2. In the window that opens, enter the name of the XML Process definition file in the field provided. Alternatively, click on the **Browse** button to select a valid process definition XML file.
3. After selecting a process definition XML file, click on the **Import** button to import it.

This imports the selected XML process definition.

See also, Import

Refer to chapter *Importing BPMN Process Models from Microsoft Visio* in the *Developer Guide*.

Export to XML

You can export a process definition as an XML file. This makes the process design portable and accessible from other applications.

To export a process design as an XML file

1. From the **Tools** menu, select the **Export** option.
2. In the **Windows File Download** dialog that opens, click the **Save** button.
3. This opens the **Windows Save As** dialog, where you can click the **Save** button to save the XML file. In this dialog, you can browse to a new location for saving the file or change its name before saving it.
4. In the **File Download** dialog you can click on the **Open** button to view the XML process definition.

See also, Export.

Import from Visio (BPMN Stencil)

You can import a process definition created with the ITP-Commerce Process Modeler for Visio.

To import a Visio process model

1. From the **Tools** menu, select the Import from **Visio Stencil**.
2. In the window that opens, enter the name of a valid XML process definition file from the Visio stencil in the field provided. Alternatively, click on the **Browse** button to select the XML file.
3. After selecting a Visio process definition XML file, click on the **Import** button.
4. This will import the Visio process definition into AVEVA Work Tasks.

See also, Import from Visio stencil

Refer to the topic *Overview of Visio Process Import* under chapter *Importing BPMN Process Models from Microsoft Visio* in the .*Developer Guide*.

Add Activities in Process Designer

There are two ways to add activities in Process Designer:

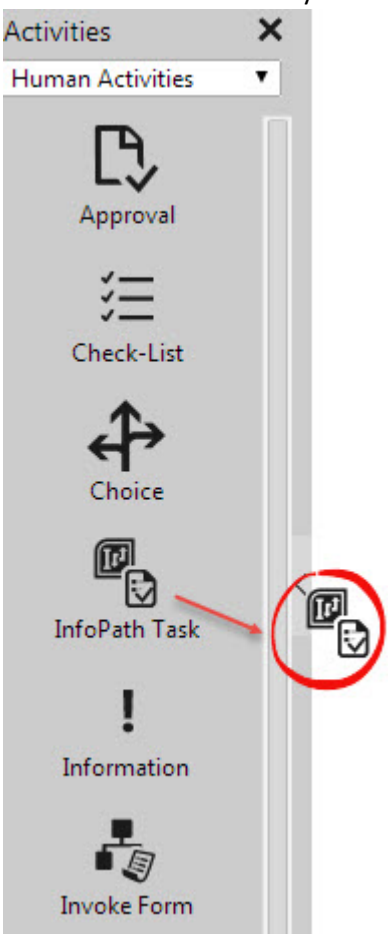
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Drag and drop activities
- Add activities from Context Menu

Drag and Drop Activities

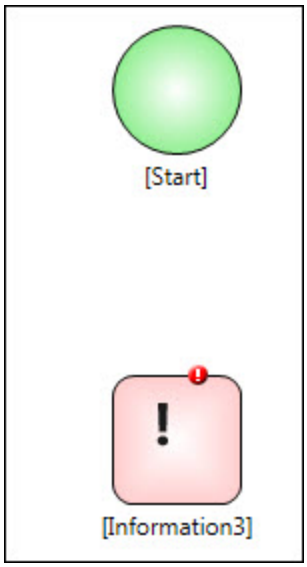
In the Process Designer, you can add an activity to the design mosaic by dragging and dropping it. For a detailed explanation of activity properties, see Using Activities.

1. Open the group to which the desired activity belongs by selecting it in the Activities area.
2. Click on the activity icon in the Activities area.
3. While holding down the mouse key, move the mouse pointer to the desired location on the design mosaic and release the mouse key.



This will drop the icon for the selected activity at the desired location. Note the red marker indicating that the properties have not been set for this activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



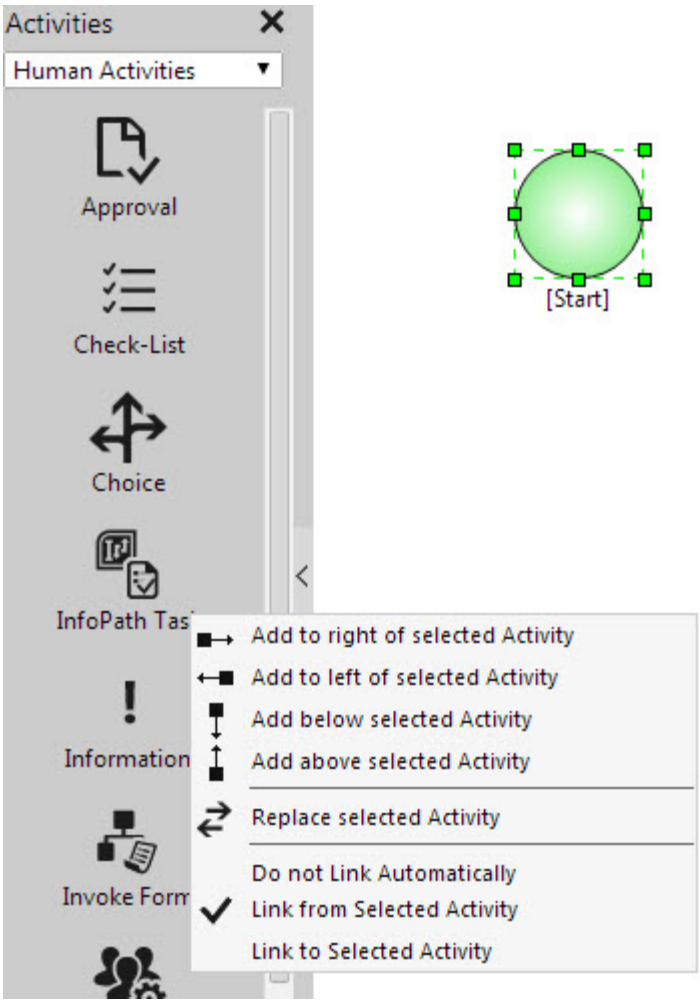
Add Activities Using the Context Menu

An alternative method to add activities to the design mosaic is to use the context menu option.

To add an activity in this way

1. Select an activity on the design mosaic with which you want to link the new activity.
2. Right-click the icon for the activity to be added to open the context menu shown below.
3. From the menu, select the appropriate **Add Activity** option. This will add the new activity to the design mosaic and link it to the selected activity.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Tip: You can specify the position of the new activity relative to any activity that is currently selected on the design mosaic using one of the relevant options in the menu. For example, you can select the "Add above selected Activity" option to add the new activity icon above the selected activity.

You can replace the selected activity with the new one by selecting the "Replace selected Activity" option. You can also set link options in the context menu. The "Link from Selected Activity" option will link the new activity from the selected activity. The "Link to Selected Activity" option will link the new activity to the selected activity. Select the "Don't Link Automatically" option if you do not want the new activity to be linked automatically to the selected activity. In this case, you can link the new activity manually to any activity on the design mosaic.

Link Activities in Process Designer

There are three ways to link activities in Process Designer:

- Click and link
- Use the **Link** tool
- Set link options in the Activities **Context Menu**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

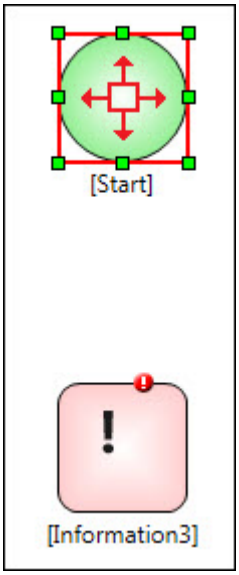
Link Activities by Clicking and Linking

Two activities can be linked by clicking on one and dragging and dropping the connector on the other.

To link two activities in this way

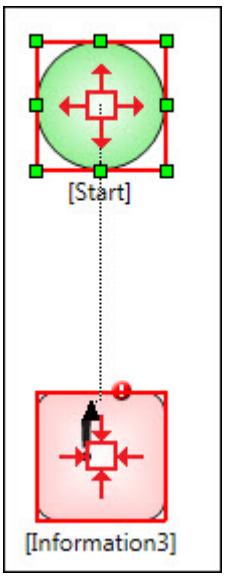
1. Click on the icon of the first activity to be linked.

   This will display a red box with arrows radiating on all sides as shown. As you move the mouse cursor away from the activity icon you will find a connector line following the mouse movement.



   Origin of the link selected

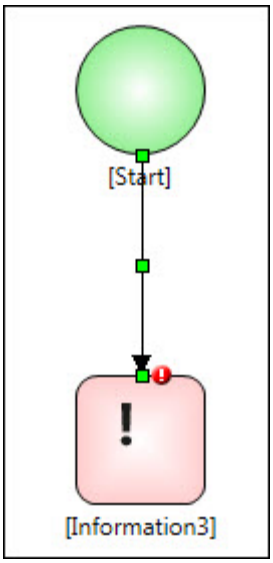
2. Now, place the cursor over the icon of the second activity to be linked and click the mouse button. This will attach the connector line starting from the first activity to the second activity completing the link between them.



   Terminus of the link selected

3. You can click the connector line to configure the link.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Link completed

Link Activities Using the Link Tool

Another method for linking two activities is to use the **Link** tool available in the **Tool** bar of the **Process Designer**.

To link two activities in this way

1. Click the **Link** tool in the tool bar.
2. In the **Link Actions** window that opens, specify the first activity to be linked in the **From Activity** field.
3. Next specify the second activity to be linked in the **To Activity** field.
4. Click **OK**.

   This will add a link connecting the two activities in the design mosaic.

See also: Link Actions.

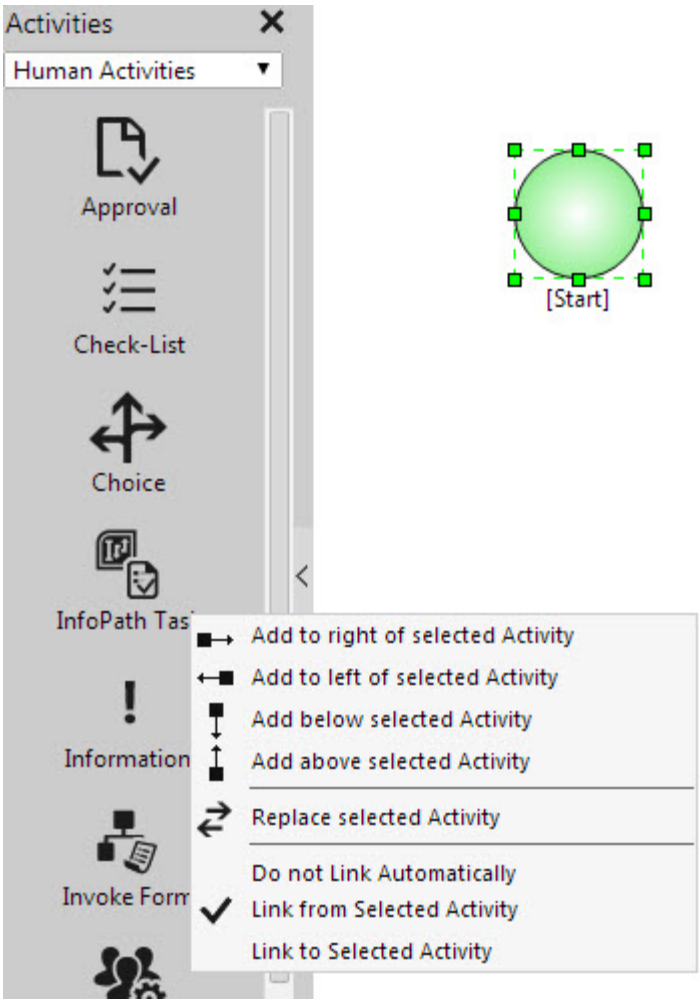
Set Link Options in the Activities Context Menu

When you add an activity using the options in the activities context menu, you can set the option for automatically linking the activity to the selected activity in the design mosaic.

To set the link property for automatic linking

1. Select an activity on the design mosaic with which you want to link the new activity.
2. Next, right-click the icon of the new activity to be linked. This will display the context menu.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



AVEVA Work Tasks activities context menu

3. In this menu, click the "**Link from Selected Activity**" option. This will display a check mark before the option to show selection.
4. Right-click the activity icon again to display the context menu and select one of the Add Activity options, say "**Add to right of selected Activity**".

   This will add the new activity to the right of the selected activity with a link from this activity to the new one.

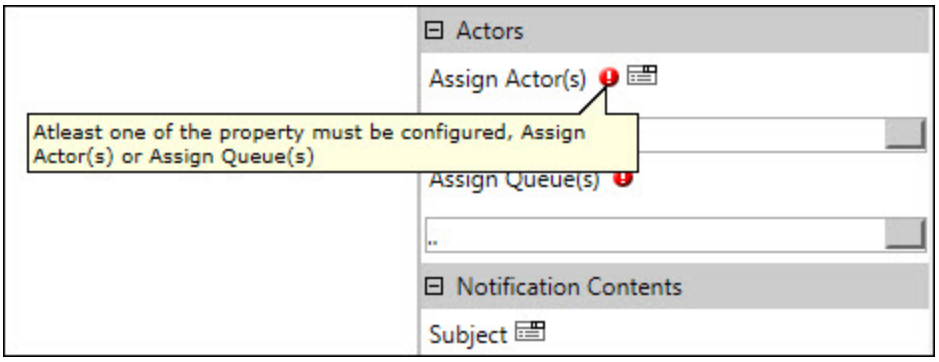
Set Activity and Link Properties

Set Activity and link properties in Process Designer. You need to set some properties of all Activities as they are essential for the execution of the activities.

Identify Mandatory Properties

You need to set some properties of all Activities as they are essential for the execution of the activities. These mandatory properties can be identified by the red icon with a white exclamation mark superimposed on the red circle that is placed next to the property labels.

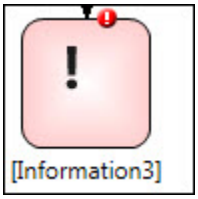
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Mandatory properties

In the above example, you can see two mandatory properties for the Checklist activity. The first property – Questions – helps configure the functionality of the activity. The second property that is visible – Assign Actor(s) – determines the actors to whom the activity will be assigned. Both these properties cannot be left empty. To ensure that users are aware of the importance of setting the mandatory properties, a message stating that the value cannot be left empty is displayed on placing the mouse over the mandatory icon. This icon is removed once the mandatory properties have been set to some value.

The mandatory icon is also superimposed on the outline of the Activity icon when it is added, indicating that its properties have not been set. The icon is removed after all mandatory properties have been set for this activity.



Checklist Activity indicating that properties have not been set.

Method 1: Right Click and Select Activity Properties

In AVEVA Work Tasks, you only need to open the Properties area when you are going to set the properties for the selected activity. This is in contrast to the earlier versions, where the Activity Properties area was always displayed, even if the properties had already been set. This change results in more space in the Process Designer UI for the design mosaic.

You can open the Activity Properties area for a selected activity by right-clicking the activity icon and selecting the Activity Properties option from the context menu. This is one of the three ways in which you can access the **Activity Properties** area.

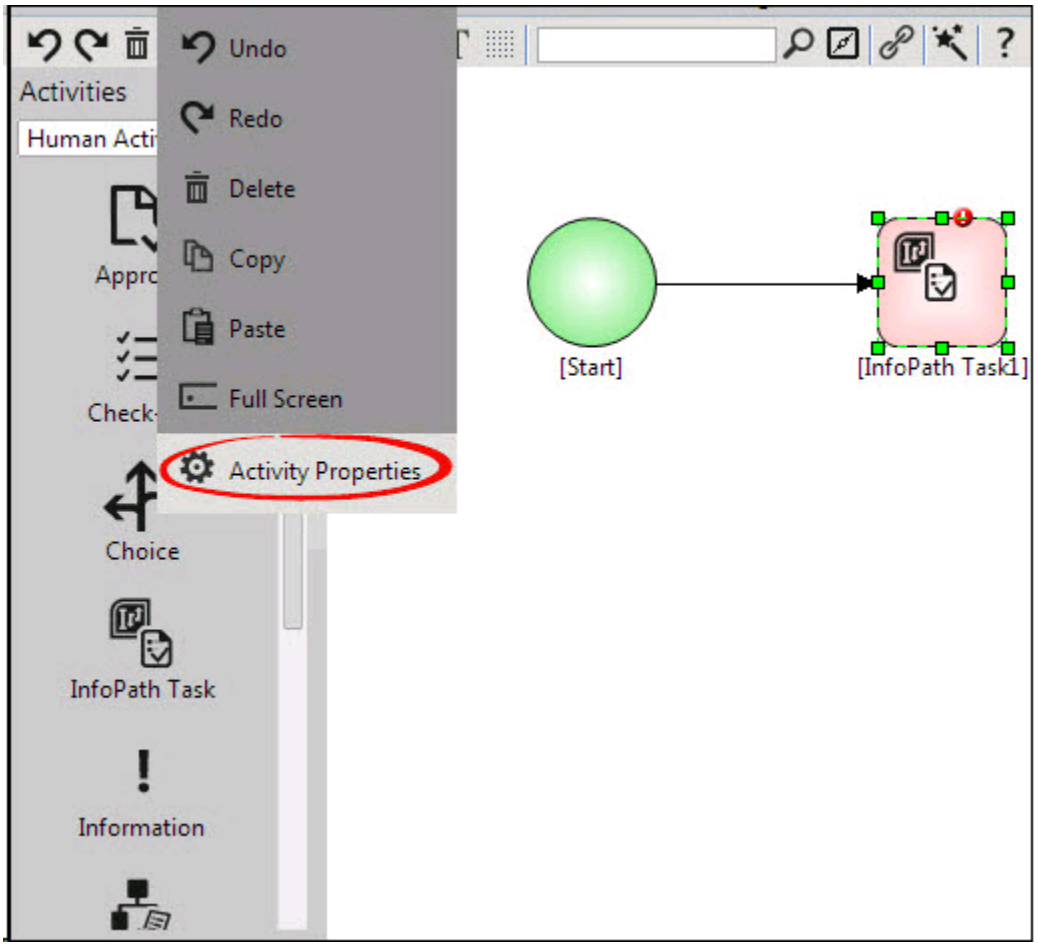
1. Right-click the icon of the activity for which you need to set the properties.
2. In the context menu, select **Activity Properties**, which is the first option.
3. This will display the **Activity Properties** area, with all the properties of the selected activity. You can set the required properties and click **Save** to close the **Activity Properties** area.

Method 2: Select Activity and then Select Activity Properties from Edit Menu

You can also open the Activity Properties area for a selected activity by clicking the activity icon and selecting the Activity Properties option from the Edit menu. This is the second method to access the Activity Properties area.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Click the icon of the activity for which you need to set the properties.
2. In the **Edit** menu, select **Activity Properties.**



3. This will display the **Activity Properties** area, with all the properties of the selected activity. You can set the required properties and click **Save** to close the **Activity Properties** area.

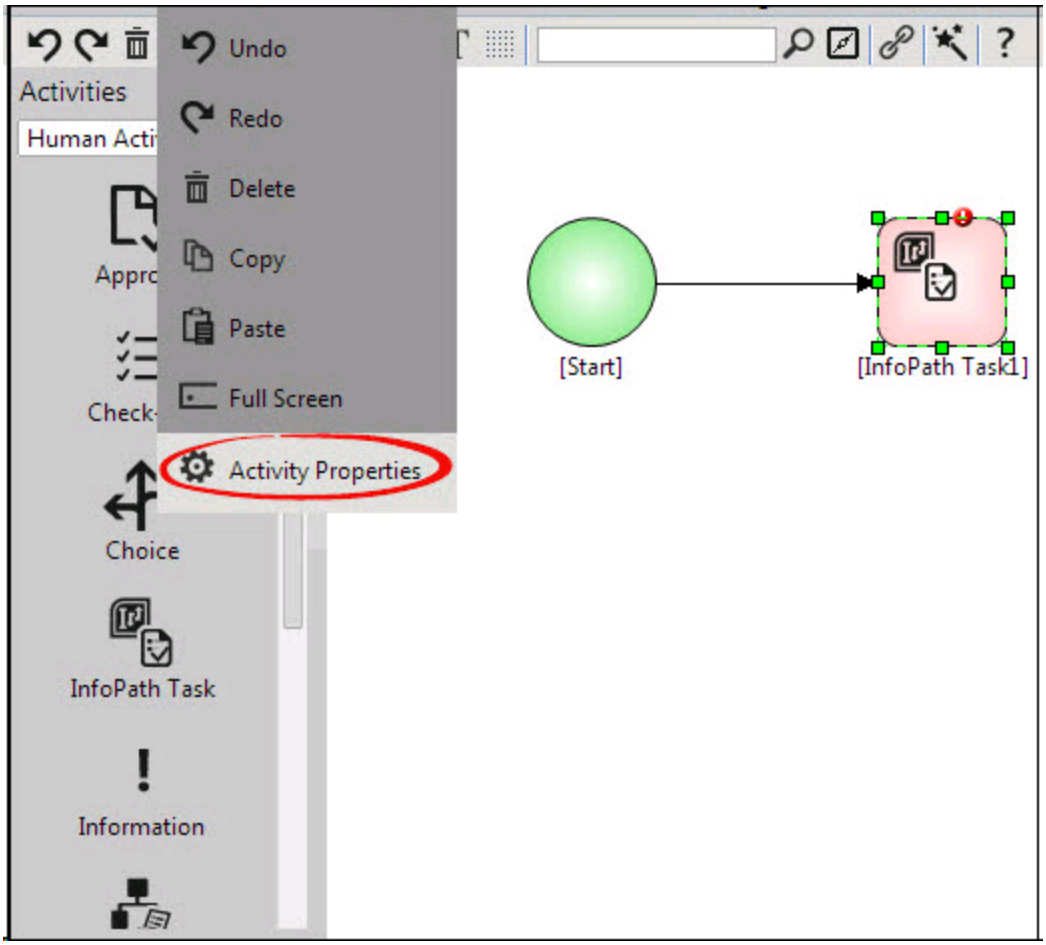
Note: You can also click the **Activity Properties** icon on the tool bar instead of selecting the option from the **Edit** menu to open the **Activity Properties**.

Method 3: Use Wizards For Resource Activities (where applicable)

You can also open the **Activity Properties** area for a selected activity by clicking on the activity icon and selecting the **Activity Properties** option from the **Edit** menu. This is the second method to access the **Activity Properties** area.

1. Click the icon of the activity for which you need to set the properties.
2. In the **Edit** menu, select **Activity Properties.**

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. This will display the **Activity Properties** area, with all the properties of the selected activity. You can set the required properties and click **Save** to close the **Activity Properties** area.

   Note: You can also click the **Activity Properties** icon on the **Tool** bar instead of selecting the option from the **Edit** menu to open the **Activity Properties**.

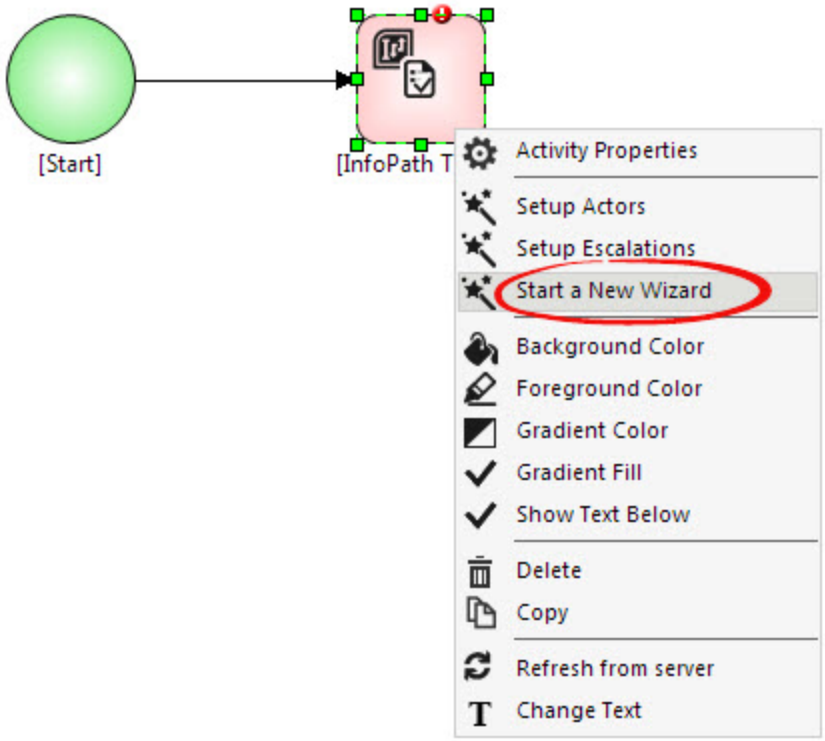
Introduction to Property Setting Wizards

AVEVA Work Tasks comes with significant improvements in terms of user-friendliness in designing the workflow. A totally new Visio-like user-interface for the Process Designer allows users to drag and drop activities and link them by dragging and dropping links. In addition to the normal method of property setting, there is an Expression Editor to create expressions for specifying the values of activity properties. Besides these, the new version comes with Wizards, which present a third mode of property setting. The Wizards are intuitive and guide the user through the different steps involved in setting key properties. There are two property wizards - Actors Selection Wizard and Escalation Email Wizard - that can be used to set the different properties related to each.

Accessing the Wizards

You can access the wizards to set the corresponding properties by right-clicking on the property icon in the Designer and selecting the appropriate wizard option in the menu. These options are only displayed in the right-click menu of the activities that have wizards.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

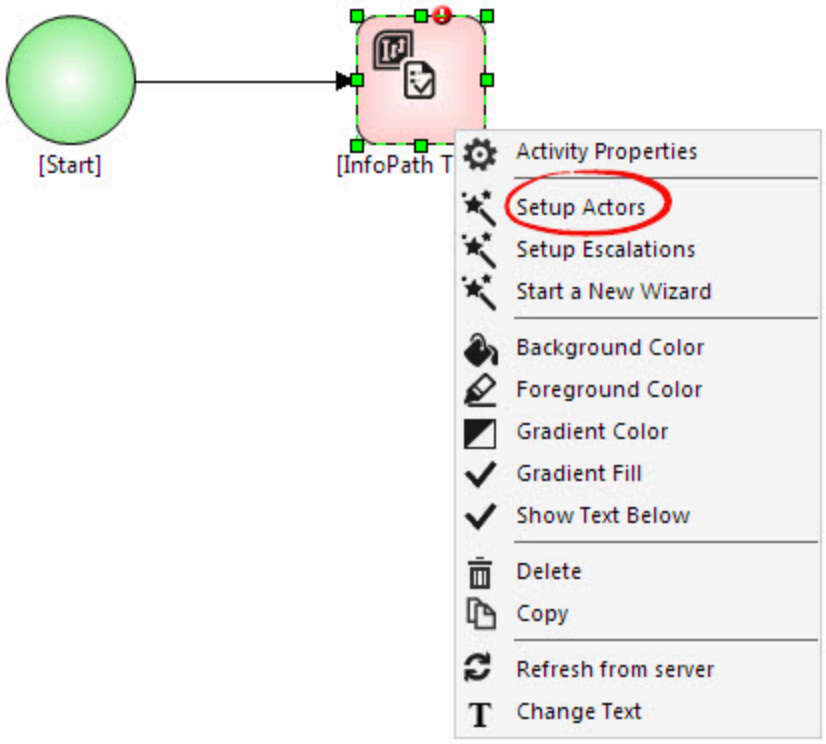


Use the Actors Selection Wizard

The Actors Selection Wizard can be used to set the properties related to the actors who will take part in the workflow process. You can access this wizard by selecting the Setup Actors option from the context menu of Human Activities, such as, Approval, Checklist.

The following image shows the wizard to launch the Setup Actors:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

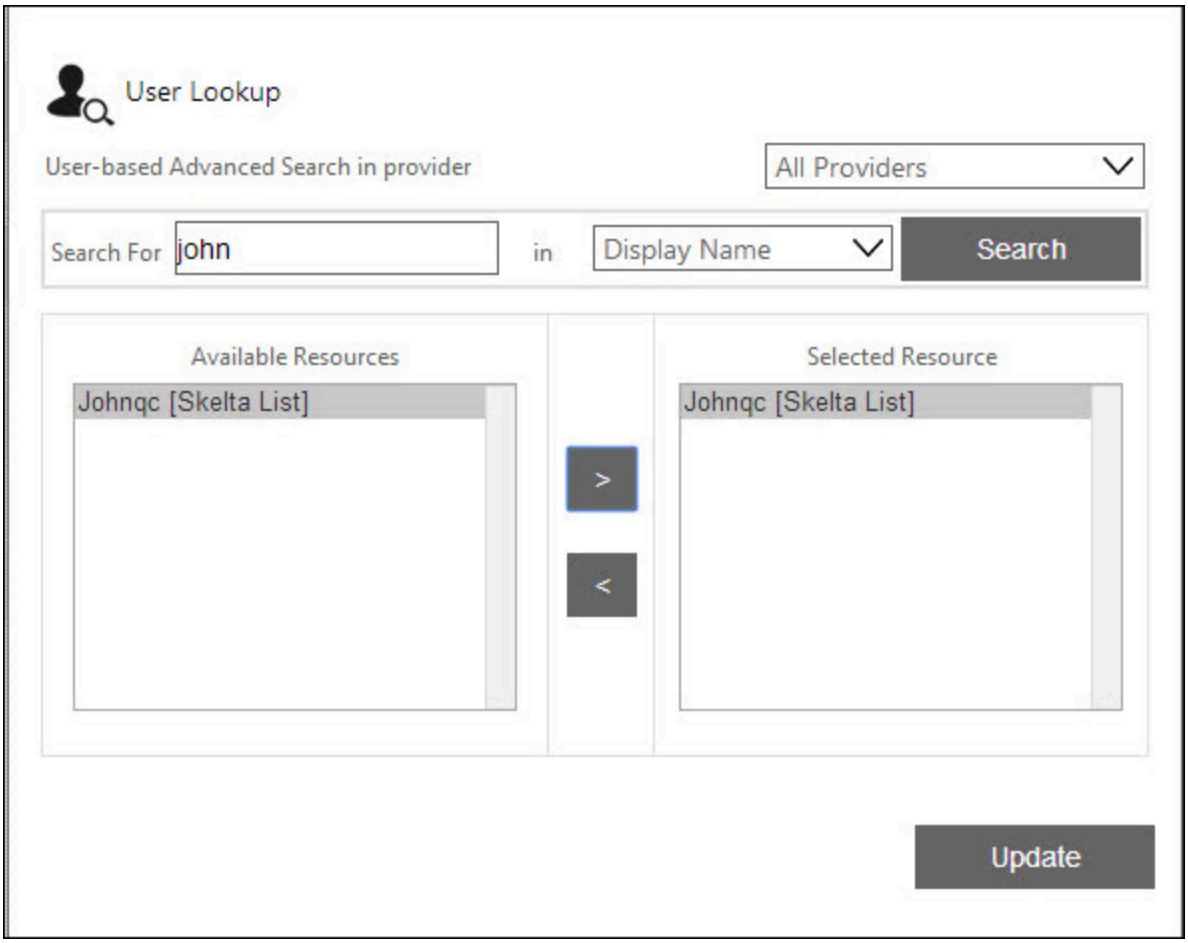


Setting the Properties through the Wizard

Step 1: Selecting the Actors

When you right-click a resource action and select the "Actors Selection Wizard" option from the menu, the window shown below, corresponding to the first step of the wizard, is displayed. In this step, the wizard guides you through the selection of resources to set as the actors. You can search for resources by specifying a search parameter. You can specify a value corresponding to user-related parameters like "Employee Name" or "Role in Project" partially or fully in the field provided and click the **Search** button to start the search. You can change the parameter by selecting from the drop-down here. The resources brought up in the search are displayed in the **Available Resources** list box. You can add a resource to the Selected Resources list by selecting the resource name in the first list and clicking the [>] button. To add all resources, click the [**>>**] button. Similarly, you can use the [<] button to remove a resource from the selected resources list or the [<<] button to clear all the selected resources. After selecting the resources, click the **Next** button to go to the next step in the Actors Selection Wizard.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

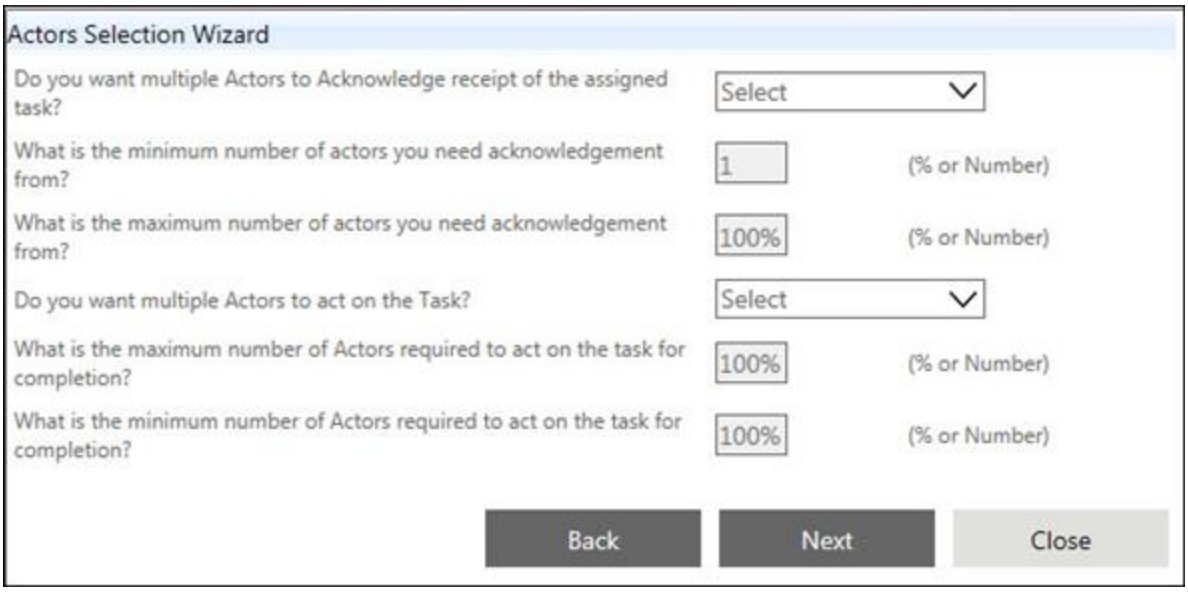


Step 2: Setting Limits for Actors and Activities

When you click the **Next** button in the first window of the Actors Selection Wizard, the window shown below is displayed. In this step the wizard guides you through the process of setting limits for the actors and activities through a series of intuitive steps. The first limit to be set is the number of actors who need to acknowledge receipt of the activity. First, specify whether or not multiple actors must acknowledge the activity by selecting "Yes" from the corresponding drop-down. After making the selection, you can set the minimum and maximum number of actors who will have to acknowledge in the next two fields. You can specify either the number of actors or the percentage of the total number of actors. Next, specify whether multiple actors will act on the task by selecting "Yes" from the corresponding drop-down. After making the selection, you can set the maximum and minimum number of actors who will have to act in the next two fields. You can specify either the number of actors or the percentage of the total number of actors. After setting the limits, click **Next** to go to the next step. If you need to make any changes to the values specified in the previous step, click the **Previous** button.

The following image shows how to set the limits:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Step 3: Overwriting Values Set through the Action Properties

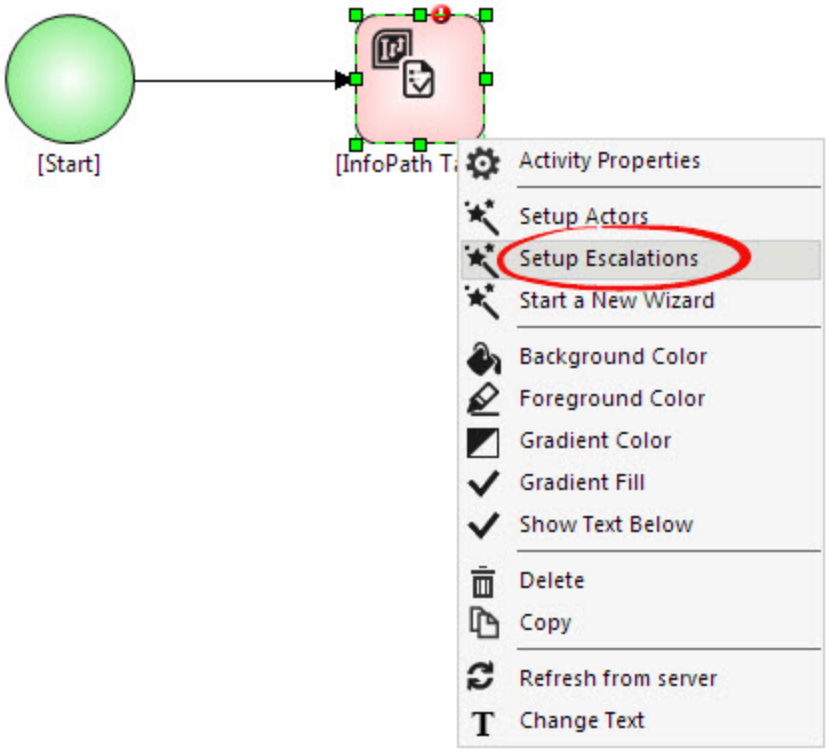
When you click **Next** in the second window of the Actors Selection Wizard, the window shown below is displayed. In this step the wizard asks you whether you would like to overwrite the values set through the Action Properties for the activity. If you choose to overwrite the existing values with the new settings, click **Finish**. This will save the new values and close the wizard. If you prefer not to overwrite the values, you can click **Close** to exit the wizard. If you need to make any changes to the values specified in the previous step(s), click **Previous**.

Use the Escalation Email Wizard

The **Escalation Email Wizard** can be used to set the escalation properties for the activity. You can access this wizard by selecting the **Setup Escalations** option from the context menu of Human Activities, such as like Approval, Checklist.

The following image shows launching setup Escalations Wizard:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



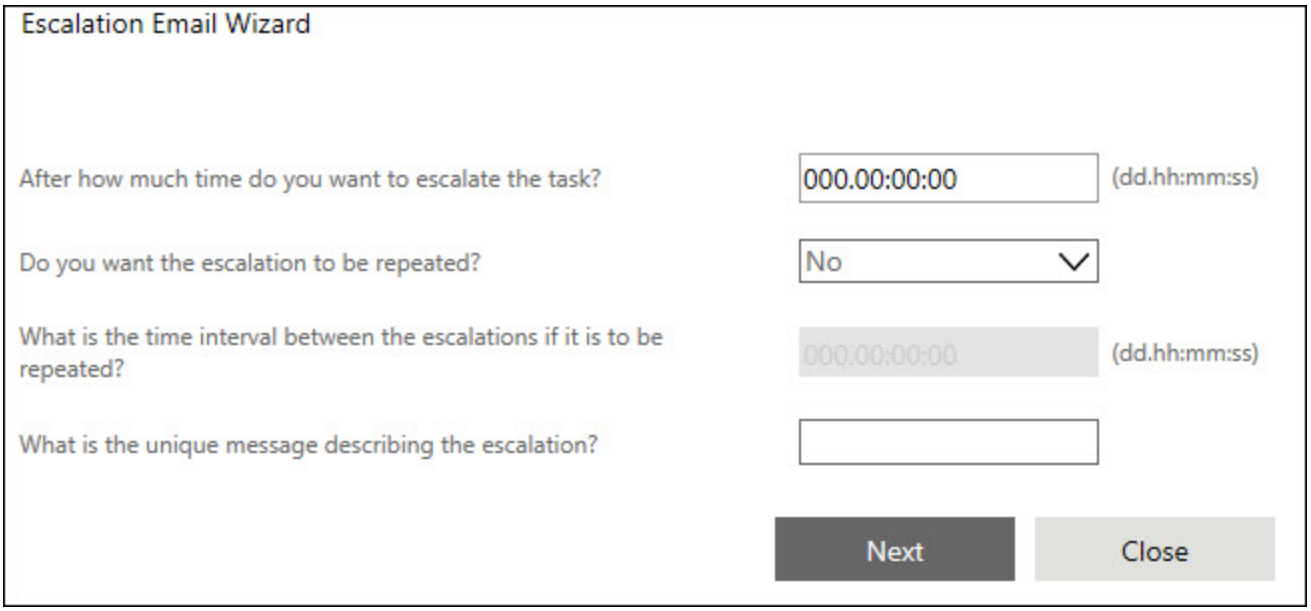
Setting the Properties through the Wizard

Step 1: Setting the Escalation Conditions

When you right-click a resource activity and select the "Escalation Email Wizard" option from the menu, the window shown below, corresponding to the first step of the wizard, is displayed. In this step the wizard guides you through the process for setting the escalation conditions for the activity. These conditions will determine at what point the task will be escalated. You can set the time limit for escalation in the first field. Specify the value in the "dd.hh:mm:ss" format. The task will be escalated after this time elapses without any action by the assigned actor. You can choose to repeat the escalation by selecting "Yes" in the corresponding drop down. In this case, you must also specify the time interval between the two escalations in the corresponding field, again in the "dd.hh:mm:ss" format. If you select "No" in the repeated escalation drop down, this field cannot be accessed. Finally, you must specify the unique message describing the escalation in the last field in this window. This can be any meaningful message that clearly describes the escalation. The example in the image below shows "Escalated to PM". After selecting the resources, click the **Next** button to go to the next step in the Actors Selection Wizard.

The following dialog box shows how to set the Escalation Conditions:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

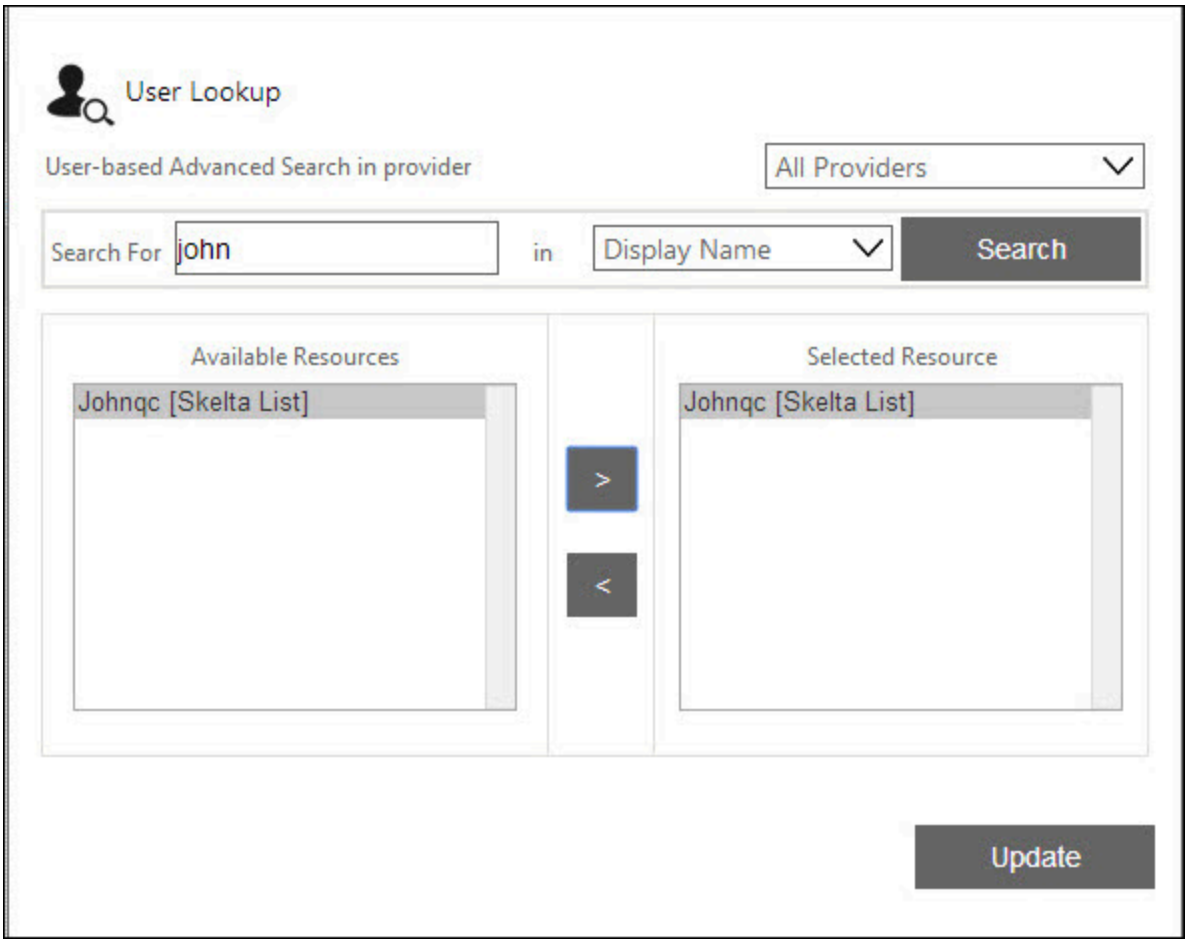


Step 2: Selecting the Actors for the Escalation

When you click the **Next** button in the first window of the Escalation Email Wizard, the window shown below is displayed. In this step, the wizard guides you through the process of selecting the actors to whom the task will be escalated. You can search for resources by specifying a search parameter. You can specify a value corresponding to user-related parameters like "Employee Name" or "Role in Project" partially or fully in the field provided and click the Search button to start the search. You can change the parameter by selecting from the drop down here. The resources brought up in the search are displayed in the **Available Resources** list box. You can add resources to the **Selected Resources** list by selecting the resource name in the first list and clicking the [>] button. To add all resources, click the [>>] button. Similarly, you can use the [<] button to remove a resource from the selected resources list, or the [<<] button to clear all the selected resources. After setting the limits, click the **Next** button to go to the next step. If you need to make any changes to the values specified in the previous step, click the **Previous** button.

The following dialog box shows how to selecting the actors:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

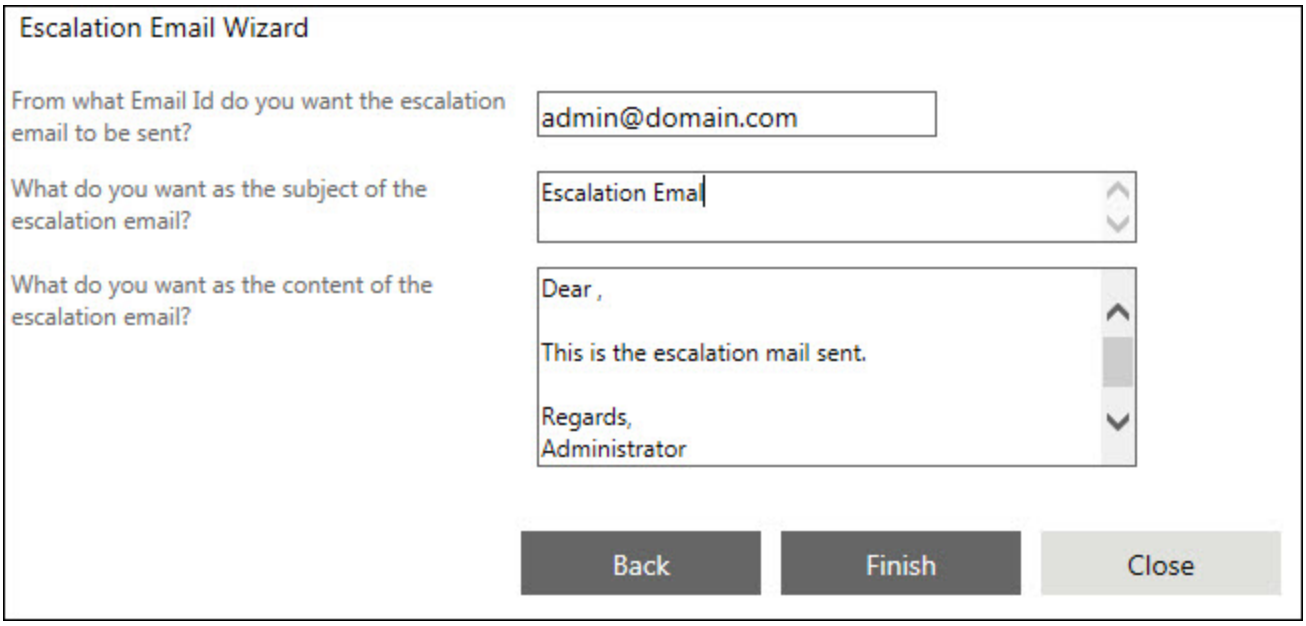


Step 3: Setting up the Escalation Email

When you click **Next** button in the second window of the Escalation Email Wizard, the window shown below is displayed. In this step, the wizard guides you through the setting of the senders Email Id, the subject of the email and its content in the corresponding fields. To save the escalation values that you have set in the wizard, click **Finish**. This will save the new values and close the wizard. If you prefer not to save the values, click **Close** to exit the wizard. If you need to make any changes to the values specified in the previous step(s), click **Previous**.

The following dialog box shows how to set up the Escalation Email:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Use the Expression Editor

The **Expression Editor** is an alternative method to set properties. It is a web control integrated with the Process Designer for building expressions with different types of objects.

Use the Expression Editor to build expressions for properties.

Note:
- Do not create or use any HTML tag in the Expression Editor.
- In Expression Editor, a hard coded value with $ symbol is not supported.

About the Expression Editor

The Activity Properties in AVEVA Work Tasks are specified by entering the actual values or by building conditions or filters using the variable and content collection. These properties are either specified in text boxes or in pop-up windows corresponding to the different properties. AVEVA Work Tasks introduces the Expression Editor as an alternative method to set properties. You can specify the expression corresponding to a property using the Expression Editor feature.

Expression

An expression is defined as any legal combination of symbols that can be evaluated to produce a value.

Expression Editor

The Expression Editor is a web control, which is closely integrated with the Process Designer for building a wide variety of expressions with different types of objects. The Expression Editor supports only the C# syntax for building expressions.

Note: Ensure that you do not create or use any HTML tag in the Expression Editor.

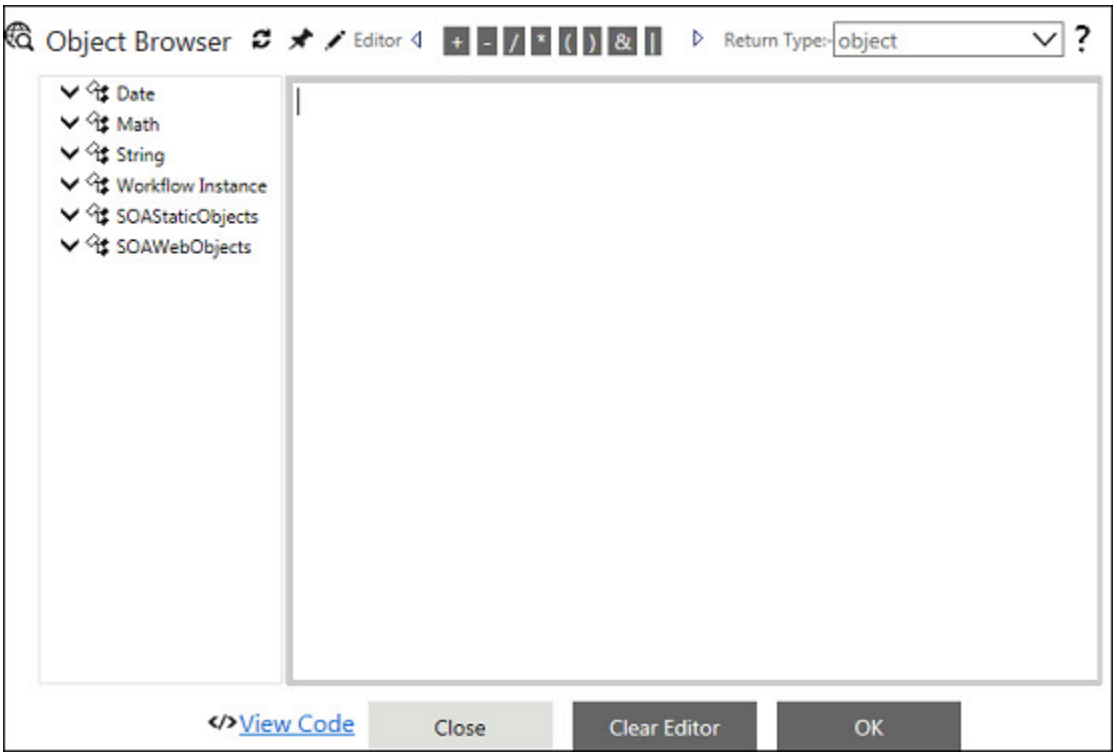
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Expression Editor window is shown in the image below. It has the Object Browser at left. This panel can be displayed or hidden using the toggle at the top. A refresh button is also available to refresh the contents. The default objects available in the Object Browser are **Date**, **Math**, **String**, **Workflow Instance**, and **SOAStaticObjects**. Each node expands to display a list of methods, properties and functions that can be used to build expressions.

The Editor is located at the right of the window. It has buttons to add the following operators at the top: **+, -** , **/**, *****, **(**, **)**, **&**, and **|**. The OK button at the bottom of the Expression Editor window can be used to save the expression.

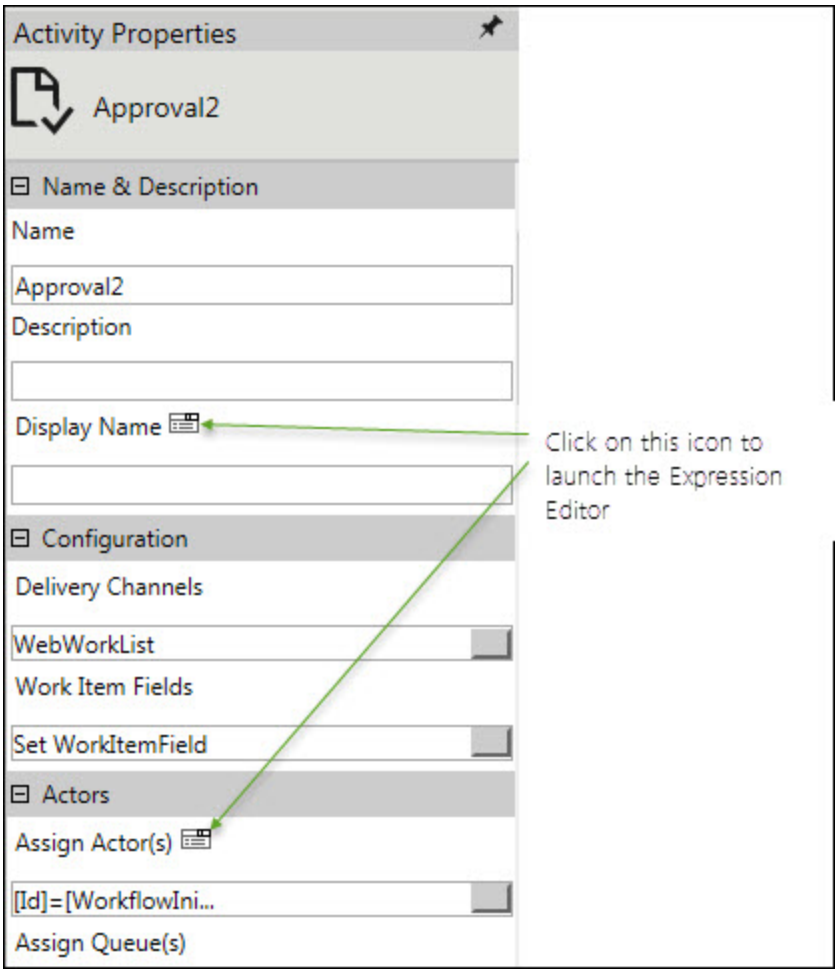
The Cancel button can be used to exit the window without saving the expression.

The **Return Type** field to the right enables selection of the value type to be returned.



You can access the Expression Editor by clicking the Expression Editor icon next to property names in the Activity Properties section of the Process Designer. In the following image, the Expression Editor icon is highlighted:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



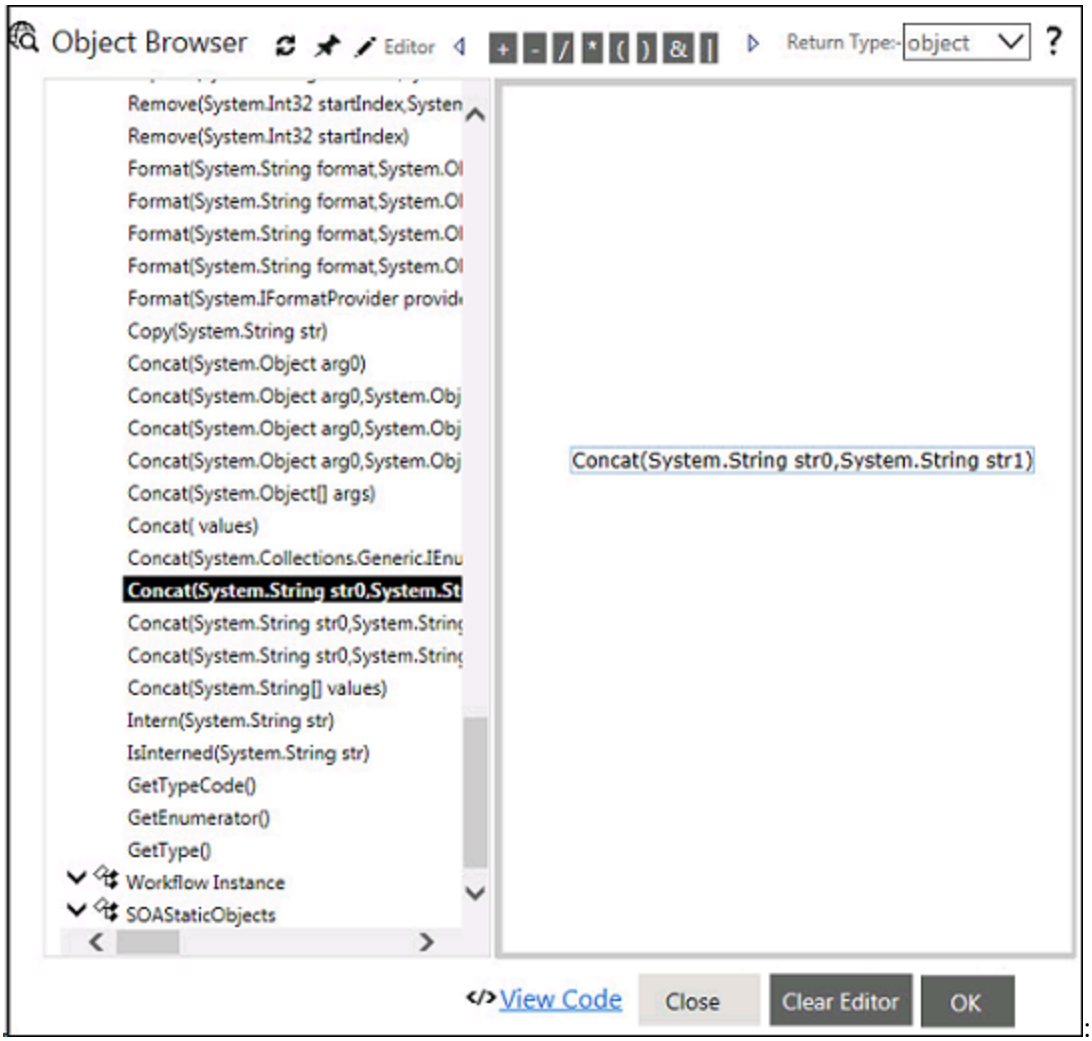
Use Expression Editor to Build Expressions for Properties

Building an expression

1. Drag and drop the methods or properties of the different objects from the Object Browser to the Editor panel. In the following example image, the **Concat(System.String str0,System.String str1)** is being dragged and dropped. The next image shows the method in the Editor. The parameters are shown as hypertext and can be replaced by actual values.

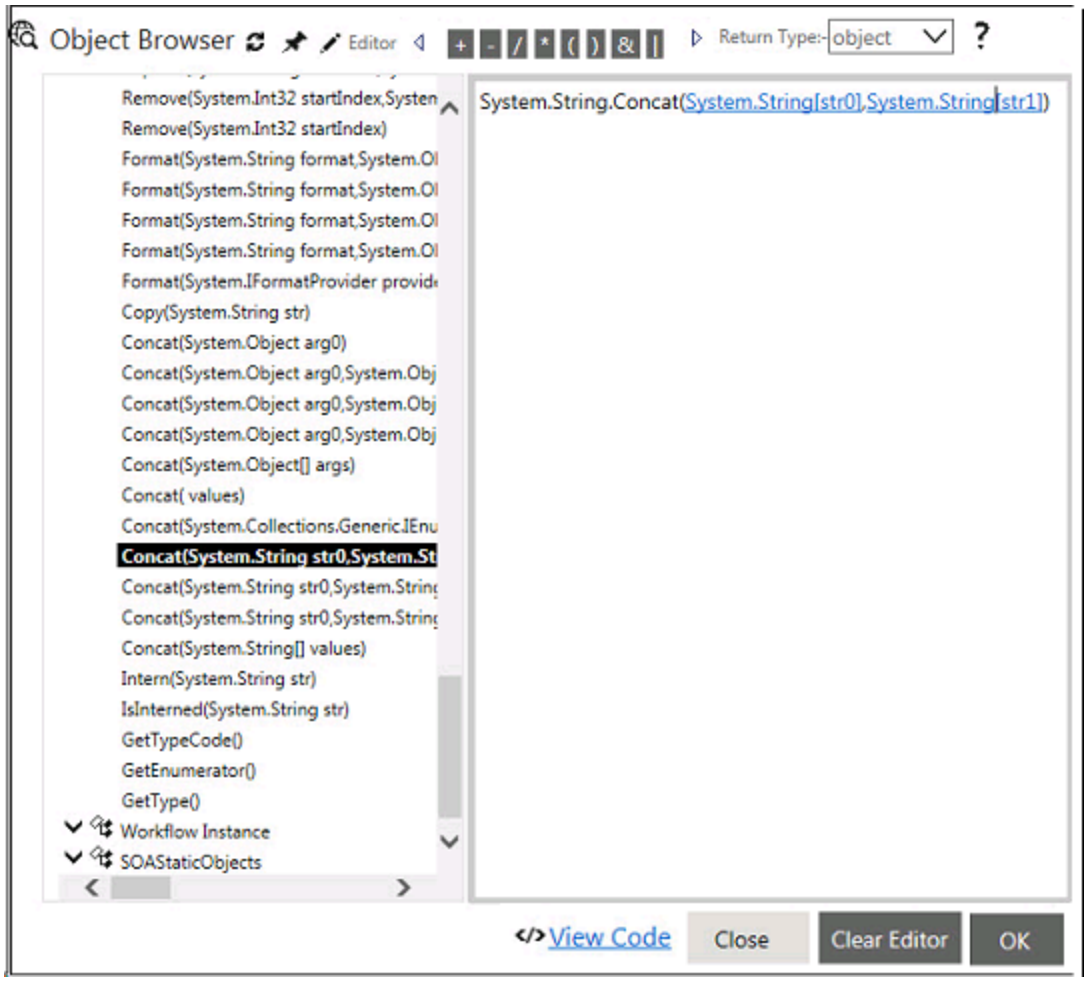
   The following image shows how to drag and drop a method to the Editor:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The following image shows method dropped on the editor:

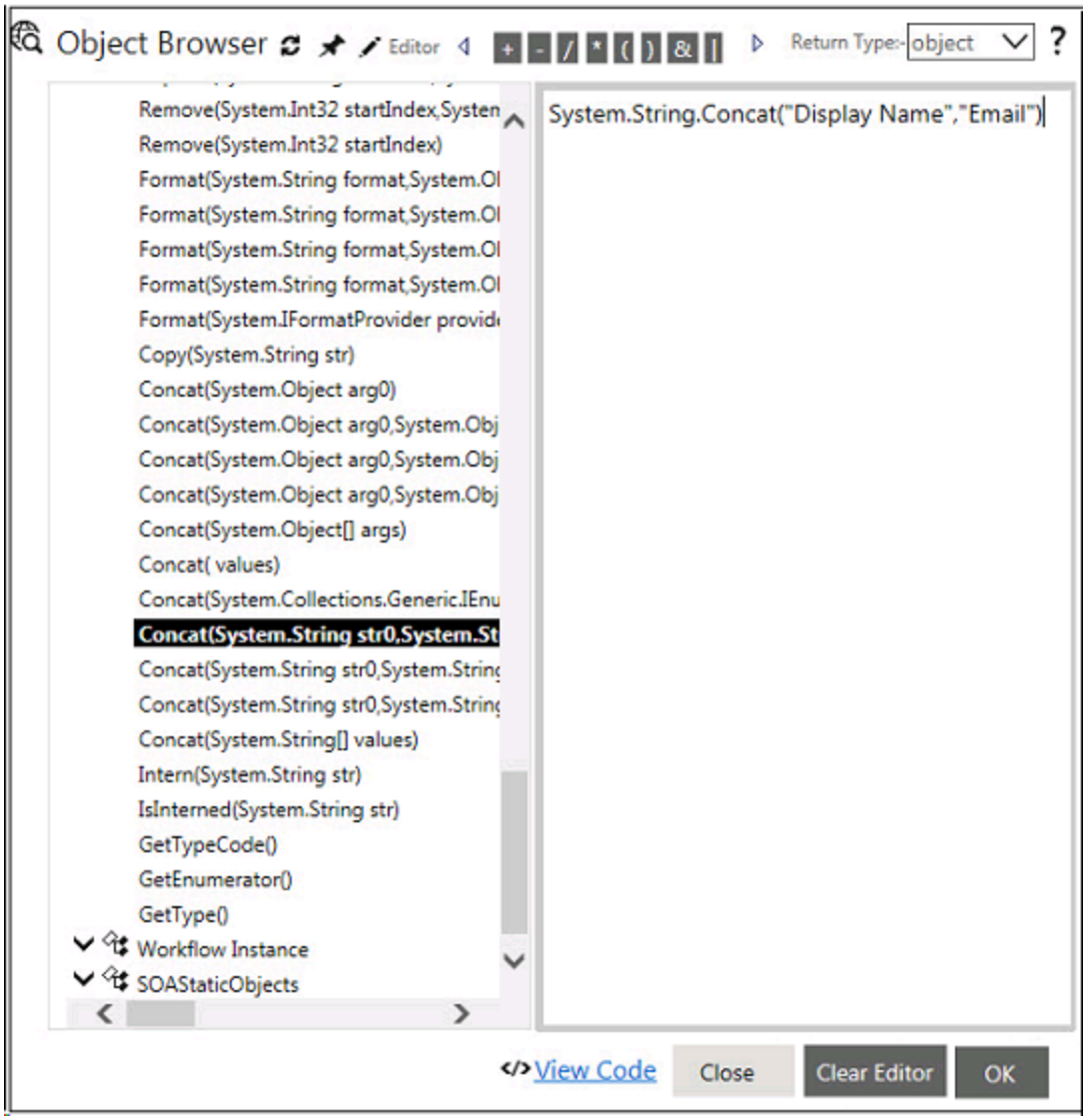
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



2. After dropping the System.String.Concat method on the Editor, you must replace its parameters. You can do so by typing text on the parameters. The text that is typed must be enclosed in double quotes as shown here.

   The following image shows how to replace method parameters by typing in text:

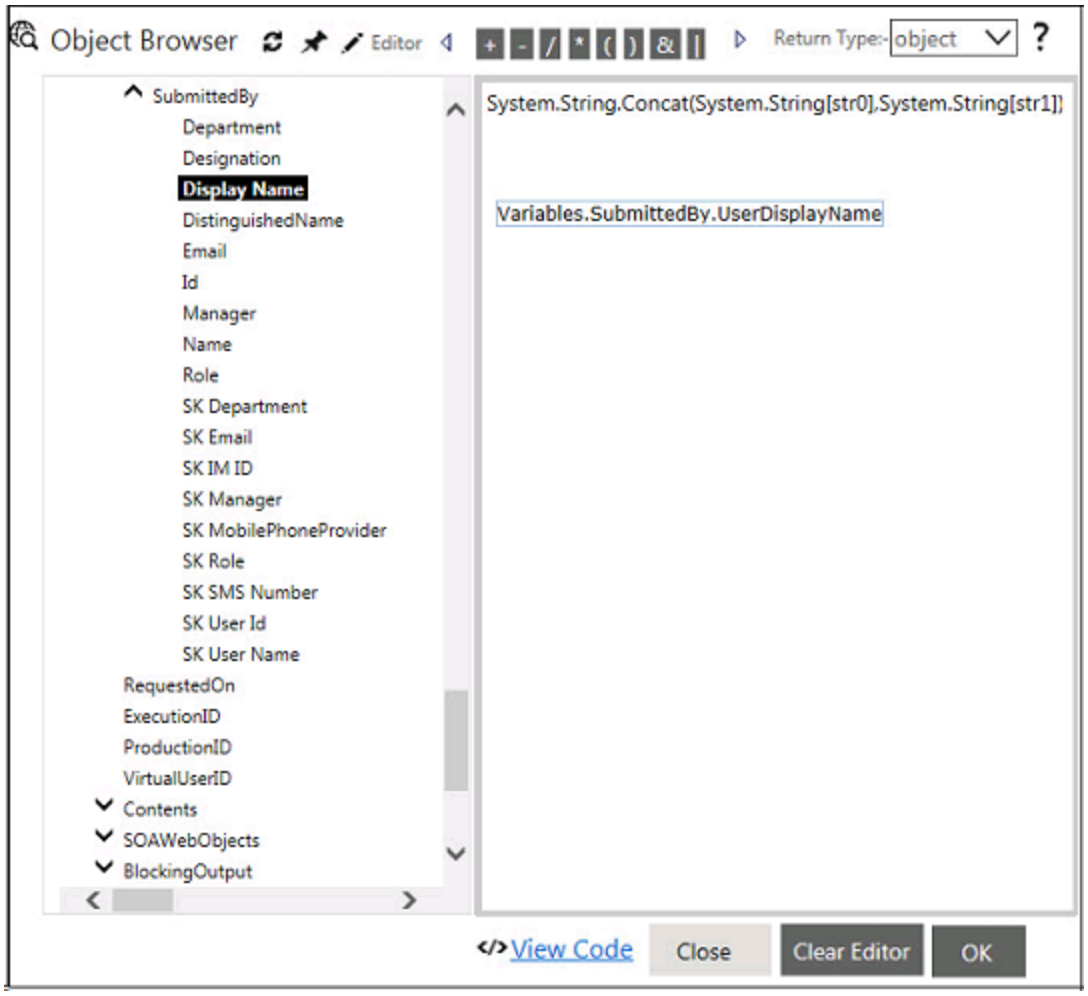
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



3. Alternatively, you can replace the parameters by placing string objects on the parameters by dragging and dropping them on to the method parameters that are shown in hypertext format. You can also use the existing variables or contents of the application by exploring the Workflow Instance node and dragging and dropping variables on to the parameters as shown in the image below. The result is shown in the image following the one below.

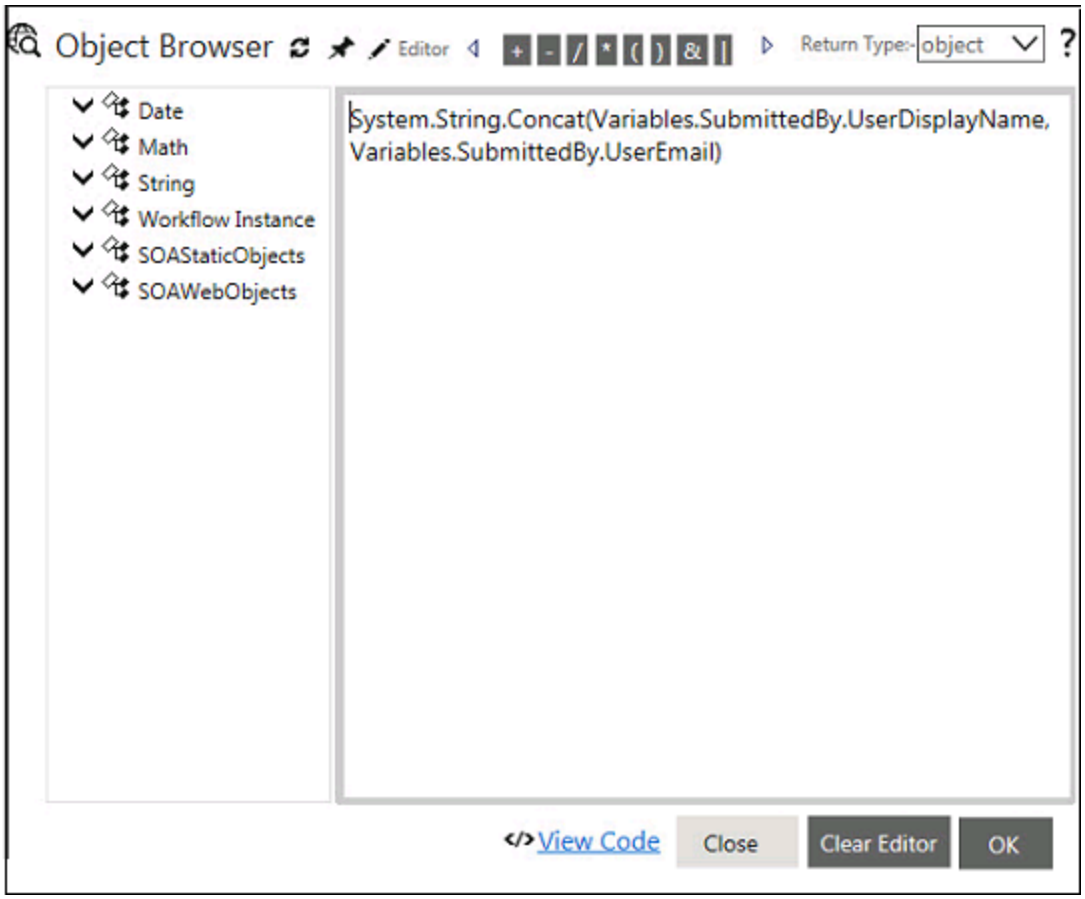
   The following image shows how to replace method parameters by dragging and dropping Workflow Instance object nodes:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



The following image shows how to replace method parameters by dragging and dropping:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



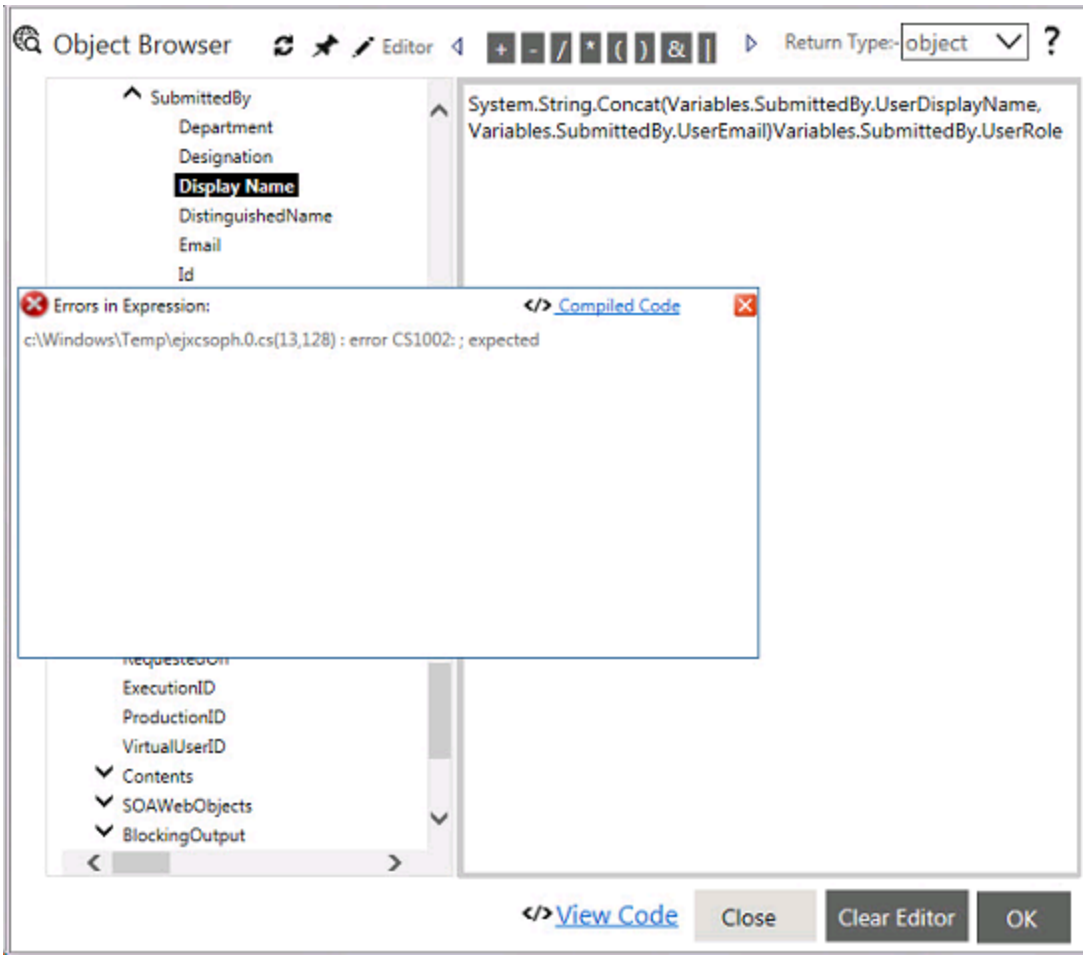
4. After completing the expression, you must click the OK button to save it. The submitted expression will be compiled and saved using the C# code compiler. If any error is present in the expression, a message box appears.

   Note: Copy-paste of variable texts from notepad will not parse the actual variable value as the referenced functions are not available. Referenced functions are available only in case of drag and drop of variable node, available from the left tree.

Debugging the expression

You can view the errors if any in an expression in the **Error List** and debug the expression. First, you need to click the **OK** button in the error message box to close it. You will find the **Error List** link at the bottom left of the Expression Editor window. You can move the mouse over this link to open the Error List shown below.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

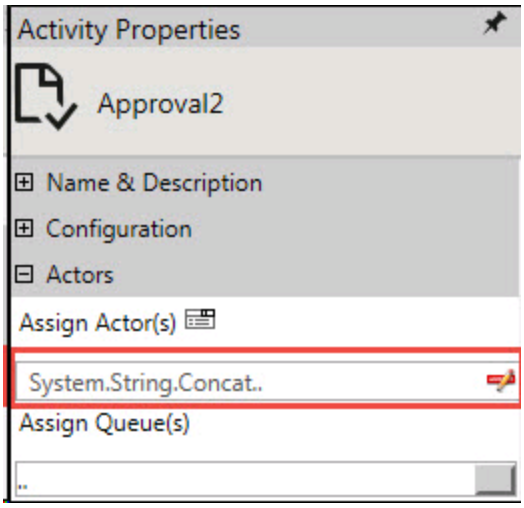


Expression added to the property

After the expression is completed without errors, it is added to the text box of the corresponding property in the Action Properties section. In the example shown below, the expression that was built in the previous steps is displayed as the value for the **Action Display Name** property. You can edit the expression by clicking on the Expression Editor icon. A "Clear Expression" icon will be displayed next to the expression. You can click on this icon to clear the expression in order to specify a different value.

The following image shows the saved expression displayed in the action properties:

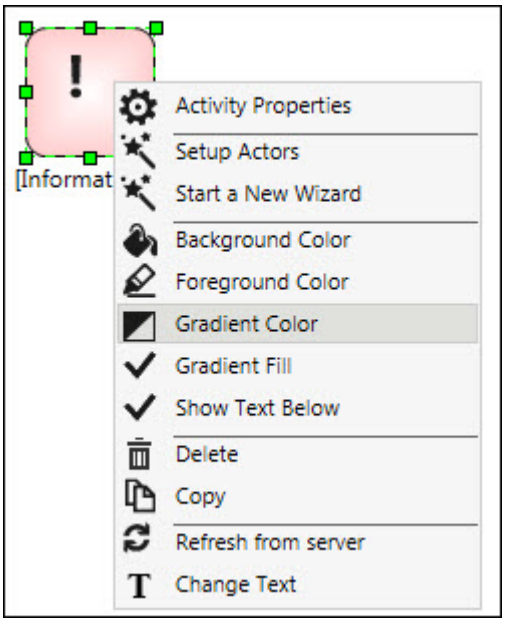
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Set Appearance Properties for Activity Icons

You can change the appearance of activity icons using the appearance properties in the context menu.

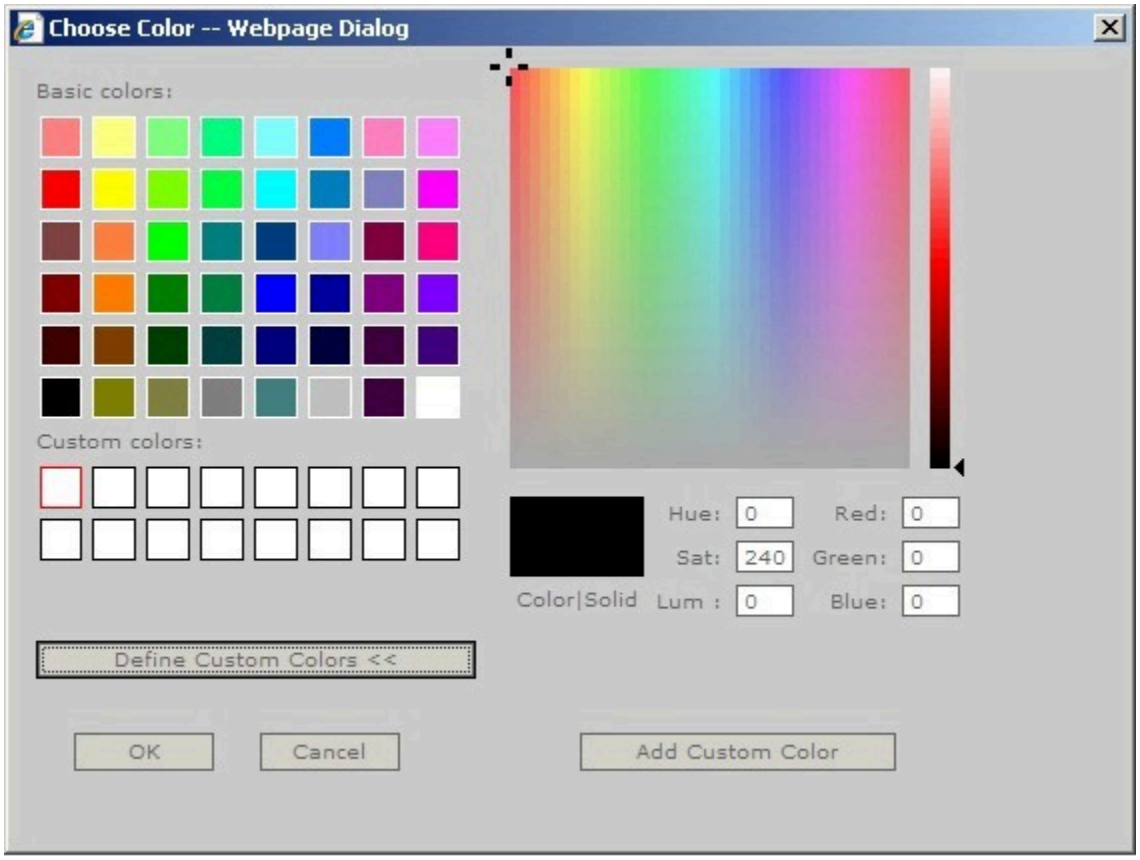
1. Right-click the icon of the activity for which you want to set appearance properties to open the context menu.



2. In the menu, select the "**Background Color**" option to open the **Choose Color** window shown below, where you can set the background color, that is fill color. You can select a color and click **OK** to set it as the background color. You can click **Define Custom Color** to expand the window to show the custom color definition controls. You can define a custom color and click **Add Custom Color** to add it to the palette.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Choose Color window with the **Custom Color** definition controls:



3. You can select "**Foreground Color**" to open the **Choose Color** window and set the foreground color, that is,

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

outline color.

4. You can select "**Gradient Color**" to set the gradient color in the **Choose Color** window.
5. You can click "**Gradient Fill**" to enable gradient fill in the icon. A check mark before the option indicates that it is enabled.
6. You can click "**Show Text Below**" to enable the display of the **Activity Name** and **Description** text below the icon. A check mark before the option indicates that it is enabled.

Search for Content in Workflow

AVEVA Work Tasks has a provision to search for content in the workflow. This is a useful tool to navigate large and complex workflow process diagrams.

To find a particular activity in a large process diagram

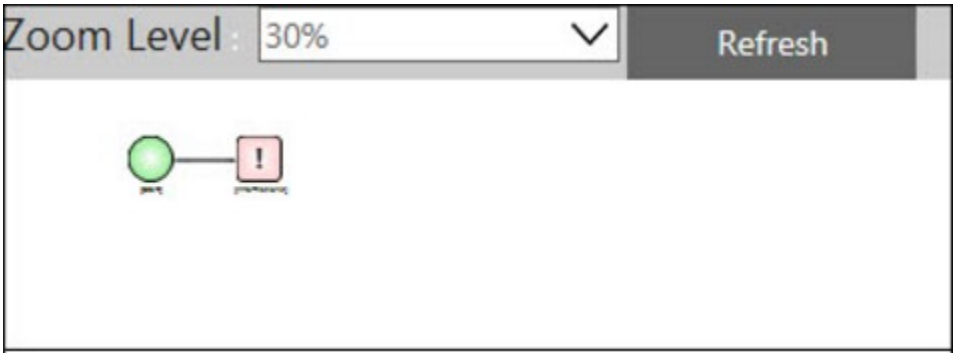
1. Enter the name or part of the name of the activity in the **Search** text box.
2. Click the magnifying glass icon to start the search.

The search will display the activity that matches the search string on the design mosaic.

Use the Process Map Image for Navigation

AVEVA Work Tasks displays a **Process Map Image**, which you can use to navigate large processes. You can access this process map by clicking the **Zoom** tool in the **Process Designer** tool bar.

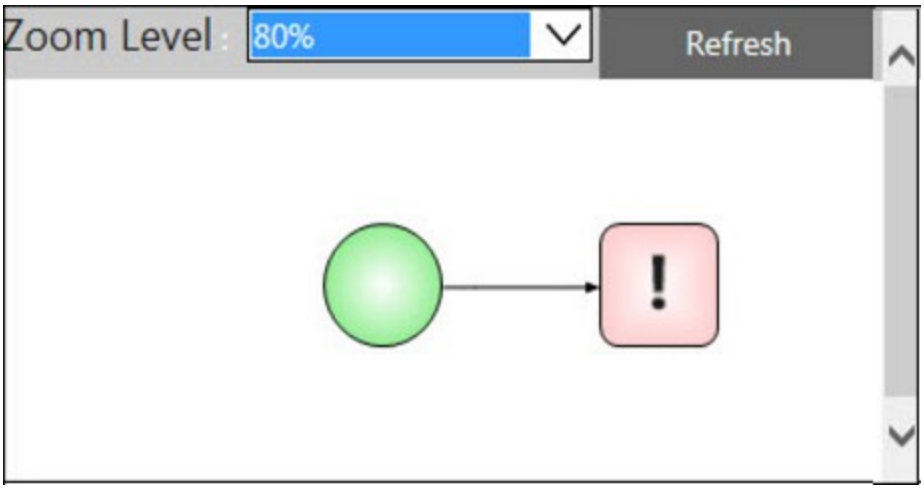
The following image shows the **Process Map** image for navigation:



The **Process Map Image** shows the process diagram as an image. You can save the displayed image as a PNG or BMP file. You can navigate large and complex processes using the click-able activity icons in this **Process Map** image. You can click any icon and go to the corresponding activity in the design mosaic of the Process Designer. You can zoom in and out of the Process Diagram by selecting the appropriate percentage from the **Zoom Level**. The image shown below is displayed at 50% zoom. You can click **Refresh** to refresh the Process Image displayed in this window.

The following image shows the Process image zoomed:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.



Process Modeling and Saving the Analyst's View

The **Process Designer** comes with a modeling layer over the process design layer. In the earlier versions, the Process Designer could only be used to design a process definition for automation. The new Process Modeler/ Designer, with its visually-rich user-interface like Visio – a tool familiar to Business Analysts – allows them to easily model business processes. In addition, the BPMN activities like Lanes and Pools allow better modeling.

With these new features you can try out various approaches to model the business process and arrive at the optimum model. At this point, you can save the model to a file. This file can be set as the Business Analyst's model, which will be shown in the Process Execution View window when you click the **Analyst's View** tab.

Alternatively, you can save a copy of the completed process design as the Analyst's View and remove the automation activities, retaining only the business model.

Lock and Unlock Designers

You can use the **Lock** and **Unlock** feature of AVEVA Work Tasks to lock the designers such as Process Designer, Forms Designer, and List designer while editing. When the designer is locked, other users will not be allowed to edit or modify it. A lock icon appears beside the locked designers as shown in the following image:

Title	Description	Owner	Version
Leave Details	Employee Leave Details	admin	1
adminform1		admin	1
adminWF		admin	1
RS	RS	admin	1
200941	200941	admin	1

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Functionality of Lock and Unlock

- You can lock or unlock a list item from the context menu.
- Every list item that you open for editing gets locked automatically.
- You can view the name of the user who has locked the list item. Place the mouse pointer on the lock icon to view the user name.
- Only the user who has locked the list item or the users of administrative security group can unlock the list item.
- You can see a lock icon beside each locked list items. The lock icons have the following significance:
  - : This lock icon indicates that the list item is locked by other users.
  - : This lock icon indicates that the list item is manually locked by the logged on user.
  - : This lock icon indicates that the list item is automatically locked by the logged on user.

Enable and Disable Lock for Repository

You can enable or disable the lock at repository level. Before you enable or disable a lock, we recommend that you understand how a lock affects the application.

Default Behavior of Repositories Created in AVEVA Work Tasks

In AVEVA Work Tasks, lock is enabled by default for a repository that is created. The Process Designer, List Designer, Forms Designer, and the User Defined Lists (if lock is enabled) gets locked while editing to ensure that other users do not modify or edit the list item.

How to Check Whether Lock is Enabled or Disabled for a Repository?

To check whether lock is enabled or disabled for a Repository, verify the Properties Column (properties collection) of SKRepository table in the Farm Database. If the value of the property key, **"enablelock"** is **True**, lock is enabled for the repository. If the value of the property key, **"enablelock"** is **False**, lock is disabled for the repository.

For example, the following code snippet shows the Properties column (properties collection) data for a repository:

```
<properties>
<property key="sourceelements">
<![CDATA[WORKFLOWELEMENTS\Default]]>
</property>
<property key="fromemail"><![CDATA[admin@domain.com]]></property>
<property key="toemail"><![CDATA[admin@domain.com]]></property>
<property key="alertuser"><![CDATA[False]]></property>
<property key="filewatcher"><![CDATA[]]></property>
<property key="enablelock"><![CDATA[True]]></property>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<property key="isadminconfigured"><![CDATA[True]]></property>
<property key="theme"><![CDATA[Default]]></property>
</properties>
```

The highlighted line is used to identify whether a lock is enabled or disabled for a repository. In this line, you can identify the following properties:

<property>	**defines the property tag**
key	Attribute of property tag which describes the property name. enablelock is name of the property.
<![CDATA[True]]>	Value of the property. True indicates that the lock is enabled.
</property>	Closing tag of property.

Enabling lock for a repository

In AVEVA Work Tasks, after you create a repository, lock is enabled by default for that repository.

If you want to enable lock for a repository, open the SKRepostiory table in the Farm Database. Locate the required repository entry or row. In the properties column data (properties collection) of the particular row, locate the enablelock key. Update the `<![CDATA[False]]>` to `<![CDATA[True]]>`.

If the *enablelock* key is not found for a repository entry in the properties column, then add the following line between `<properties>` and `</properties>` tags.

```
<property key="enablelock"><![CDATA[True]]></property>
```

The lock will be enabled for that repository.

Disabling lock for a repository

In AVEVA Work Tasks, after you create a repository, lock is enabled by default for that repository.

If you want to disable a lock for a repository, then open the SKRepostiory table in the Farm Database. Locate the required repository entry or row. In the properties column data (properties collection) of the particular row, locate the enablelock key. Update the `<![CDATA[True]]>` to `<![CDATA[False]]>.`

If the enablelock key is not found for a repository entry in the properties column, then add the following line between `<properties>` and `</properties>` tags if you want to enable lock in future.

```
<property key="enablelock"><![CDATA[False]]></property>
```

Different Types of Locks

You can use the **Lock** and **Unlock** feature of AVEVA Work Tasks to lock the designers such as Process Designer, Forms Designer, and List designer while editing.

There are two types of Locks and Unlocks:

- Auto Lock and Unlock
- Manual Lock and Unlock

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Manual Lock and Unlock

List item can be locked for editing either manually or it can be auto locked. The section below gives you detailed information on how to manually Lock or Unlock a List item.

To lock the Designers

1. Select the list item to be locked.
2. Right-click on it and select the option **Lock** from the context menu.
3. A message **'The item is locked successfully'** is displayed, click **OK**.

The selected list item will be locked and a lock icon will appear beside the list item.

To unlock the Designers

Locked designers can be unlocked only by the lock owner, administrator and the list item owner.

1. Select the list item which needs to be unlocked.
2. Right-click on it and select the **Unlock** option in the context menu.
3. A message **'The item is unlocked successfully'** is displayed, click **OK.**

The locked list item will get unlocked and the lock icon beside the item item will no longer be present.

Auto Lock and Unlock

When editing or designing a list item, the designers like Process Designer, Forms Designer, and List Designer will get auto locked and the auto lock icon , will be displayed beside the locked list item. Designers which are auto locked will get unlocked automatically when you save or discard the changes done.

Note: The designers which are accessed for editing or designing will not get locked if the Designer window is closed using the window close icon. Only the user who had locked the list item, users of administrative security group and the list item owner can unlock the list item.

Use Lock and Unlock on Designers

This section gives you in-depth details of using the Lock and Unlock feature on AVEVA Work Tasks Designers. The Lock and Unlock feature can be used for the following lists:

- Process Designer is used to design Workflows
- Forms Designer is used to design Forms
- List Designer is used to design and manage List infrastructure

Process Designer

This section gives you details on how to lock and unlock a workflow and the functionality of Workflow when the feature Lock and Unlock is used.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

How to lock and unlock Workflow

Workflow can be locked and unlocked manually or the auto lock and unlock feature can be used. Access the links below for more details on:

Manually Locking a workflow

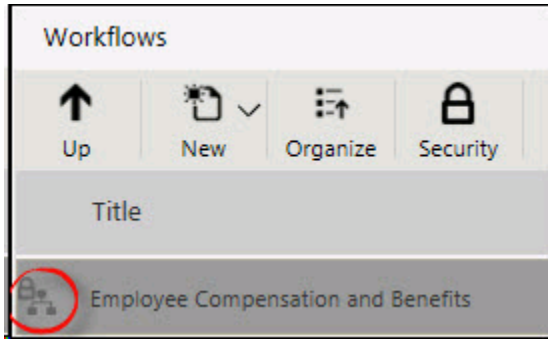
A Workflow can be locked by selecting the Lock option from the context menu. The section below gives detailed instructions of how to lock and unlock a workflow:

To Lock and Unlock a Workflow

Consider a scenario where Jack wants to lock the workflow 'Employee Compensation and Benefits' to ensure that no other user edits or modifies it.

1. Access Workflow List from the **Enterprise Console**.
2. Select workflow **'Employee Compensation and Benefits'**.
3. Right-click on the workflow and select the option **Lock** from the context menu.

   Workflow **'Employee Compensation and Benefits'** gets locked and a lock icon is visible beside the workflow in the **Workflow List**.



4. Perform all required actions on the Workflow, such as designing, editing, defining/updating activity properties, publish.

   Note: Jack can design/edit the workflow multiple times, as the workflow is locked by him. Workflow 'Employee Compensation and Benefits' will be locked even after Jack exits the Process Designer as they had explicitly locked the workflow from the context menu options. Jack has to Unlock the workflow, in order to allow others to edit it.

5. To unlock the workflow, select the workflow **'Employee Compensation and Benefits'**.
6. Right-click on the workflow and select the option **Unlock** from the context menu.
7. A message **'Item has been Unlocked'** will be displayed. Click **OK**.

   Workflow **'Employee Compensation and Benefits'** will get unlocked and the lock icon beside the workflow will not longer be displayed. Other users will be able to modify the workflow.

To Auto Locking of a workflow

A workflow gets auto locked when the options Edit or Design is selected. The workflows which get locked automatically will get unlocked when you save the changes or when the Process Designer is closed.

Consider a scenario where Jack is editing the workflow 'Employee Compensation and Benefits'.

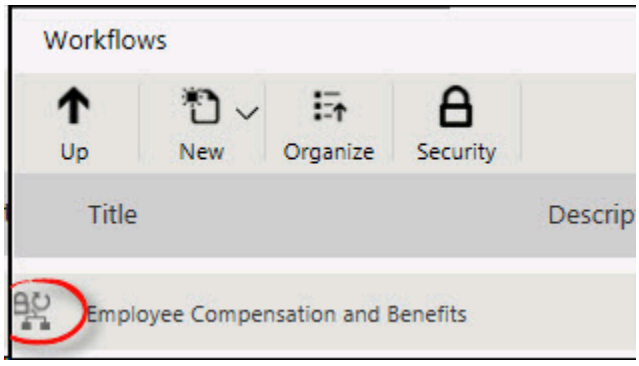
© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Access Workflow List from the **Enterprise Console**.
2. Select workflow **'Employee Compensation and Benefits'**.
3. Right-click on the workflow and select the option **Edit Workflow** from the context menu.

   Or

   Select the **Edit** option from the ribbon bar.

   The Edit Workflow window of **'Employee Compensation and Benefits'** is displayed.
4. Select the option **Design a workflow** and click **Save & Continue**. The **Process Designer** is displayed.
5. Minimize the Process Designer window and refresh the workflow list. An lock icon is visible beside workflow 'Employee Compensation and Benefits', which indicates the workflow was auto locked on selecting the edit option.



6. Click **Publish** and **Test Run**, save the changes and close the **Process Designer**.
7. Verify the workflow **'Employee Compensation and Benefits'** in the workflow list.

   Workflow **'Employee Compensation and Benefits'** is unlocked automatically as Jack closed the Process Designer and the lock icon beside the workflow is no longer present.

Actions which are allowed or restricted on a locked Workflow at a glance

The table below lists out the actions which are allowed and which are restricted for a Locked workflow:

Actions allowed	Actions restricted
Organize	Design
Show All Versions	Edit
Show ChildItem	Delete
View Workflow	Publish
View Associations	Un Publish
View Exceptions	Change Owner
View Audit History	Associate
View Properties	-

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Note: User will be allowed to perform the above mentioned actions only if they have appropriate rights.

The table below categorizes the actions which locks the workflow and which do not lock the workflow

Actions which auto lock the workflow	Actions which do not auto lock the workflow
Edit	Delete
Design	Publish
Task	Unpublish
-	Change Owner
-	Associate
-	Organize
-	Show All Versions
-	Show Child Items
-	View Workflow
-	View Exceptions
-	View Associations
-	View Audit History
-	View Properties

Impact of locking a workflow on the ribbon bar actions

When a workflow is locked by an user, any action performed from the ribbon bar which leads to modification of the workflow will be restricted for all other users. Only the lock owner will be able to perform all the actions from the ribbon bar. The scenario given below explains the behavior of the ribbon bar options when a workflow is locked:

Scenario:

John locks the workflow **'Employee Compensation and Benefits'** to make some changes to it. Kevin logs-in to the **Enterprise Console** , accesses the Workflow list and selects the workflow **'Employee Compensation and Benefits'** which is locked by John. Kevin will not be able to perform any action which leads to modification of the workflow, **'Employee Compensation and Benefits'** as John has locked the workflow.

This section explains the behavior of the ribbon bar options when Kevin performs the actions given below:

- **Design**: When Kevin selects the workflow **'Employee Compensation and Benefits'** and clicks on the Design option, a message '**Item is currently locked by another user, John'**, will be displayed.
- **Edit**: When Kevin clicks on the **Edit** option in the ribbon bar with the workflow **'Employee Compensation and Benefits'** selected, a message '**Item is currently locked by another user, John'**, will be displayed.
- **Delete**: When Kevin tries to delete the workflow **'Employee Compensation and Benefits'** using the ribbon

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

option **Delete**, a message '**Item is currently locked by another user, John'**, will be displayed.

- **Publish**: With the workflow **'Employee Compensation and Benefits'** selected, if Kevin clicks on the **Publish** icon in the ribbon bar, a message **'Item is currently locked by another user, John'** will be displayed.
- **Change Owner**: Kevin will not be allowed to change the owner of the workflow **'Employee Compensation and Benefits'**, using the **Change Owner** option in the ribbon bar, as the workflow is locked by John and a message '**Item is currently locked by another user, John** will be displayed.
- **Associate**: Kevin be allowed to Associate the workflow '**Employee Compensation and Benefits**' to Share Point Enterprise console, File watcher etc, even though it is locked by John.
- **Organize**: Kevin can use this feature, and re-organize the workflow list items, as Lock will not be considered for Organize.

Note: John who locked the workflow **'Employee Compensation and Benefits'** will be able to perform all the above mentioned actions as no restrictions will be levied for the lock owner.

Impact of locking a workflow on the Context Menu options

When a workflow is locked by an user, all the context menu options of the locked workflow will not be available for the other users. The scenario in the section below explains the behavior of the context menu options when a workflow is locked:

Scenario

John locks the workflow **'Employee Compensation and Benefits'** to make some changes to it. Kevin logs in to the **Enterprise Console** , accesses the Workflow List and selects the workflow **'Employee Compensation and Benefits'** which is locked by John. However, Kevin will not be able to perform any action which leads to modification of the workflow, '**Employee Compensation and Benefits'**. Context menu options which Kevin can view and use for the locked workflow '**Employee Compensation and Benefits'** are:

- View Workflow
- Export Workflow
- Create Associations
- Dissociate
- View Associations
- Unlock
- Show Child Item
- Show All Versions
- View Audit History
- View Properties
- View Exceptions

Context menu options which Kevin cannot view and use for the locked workflow **'Employee Compensation and Benefits'** are:

- Design Workflow
- Edit Workflow
- Publish

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Publish and Test Run
- Import Workflow
- Lock
- Save As
- Disable
- Delete
- BAM Configuration
- Change Owner
- Tasks
- Security settings

Tasks

The table below explains whether the workflow will remain locked or will get unlocked on selecting the options available in the **Workflow Tasks** window if the selected workflow is auto locked.

Action performed	**Lock Status of the Workflow**
Design the Workflow now	Workflow will remain locked till the lock owner completes the designing or editing process and saves the changes done or discards it.
Provision Now	Workflow will remain locked till the lock owner completes the provisioning process
Start the Workflow Wizard	Workflow will remain locked till the lock owner completes the Workflow wizard
No task to be performed now	The workflow will get unlocked as the lock owner decides that no task will be performed for the selected workflow.

Context menu options available for Lee who is an administrator at repository level are:

- View Workflow
- Export workflow
- Create Associations
- Dissociate
- View Associations
- Unlock
- Show Child Item
- Show All Versions
- View Audit History
- View Properties
- View Exceptions

Note: Since Lee is a administrator at repository level, they can see the Unlock option in the context menu

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

and can use it to Unlock the workflow.

Features

- Workflow will get auto locked only when the options **Design** or **Edit** are selected.
- When a workflow is locked, the child workflows will not get locked and other users can work on it.

  Scenario

  Consider a workflow **Finance - Fiscal year 2011- 2012**, which has the following child items

  - Incentive / Bonus
  - Contract employee's compensation
  - Employee Provident Fund calculator and contribution

    Login to Enterprise Console as John, access Workflow list .

    Select the workflow **Finance - Fiscal year 2011-2012** and lock it by selecting the Lock option from the context menu.

    Log out and login as Kristen and navigate to the workflow list.

    Workflow Finance - Fiscal year 2011-2012 is locked and a red lock icon beside it indicates it is locked by a different user.

    Double click **workflow Finance - Fiscal year 2011-2012** to access its child workflows.

    A list of child workflows of **Finance - Fiscal year 2011-2012** is displayed. Verify the lock status of the child workflows.

    The child workflows of **Finance - Fiscal year 2011-2012** are not locked though the parent workflow is locked. Hence, Kristen will be allowed to modify it if required.
- When a workflow is locked, all the versions of that particular workflow will get locked.Scenario

  Consider a workflow Leave Approval Workflow, which has four versions

  Scenario

  Log into Enterprise Console as Lee and access Workflows List.

  Select the workflow **Leave Approval Workflow** which has four versions and lock it.

  Log out and log in as Lan, access Enterprise Console > Workflows List.

  Select the locked workflow **Leave Approval Workflow**, right click on it and select the option **Show All Versions**.

  All the versions of Leave Approval Workflow is displayed, wherein you cannot see the Lock icon and determine if the versions are locked or not.

  Select a version and click the **Edit** icon in the ribbon bar.

  A message **'Item cannot be edited because it is locked by the user Lee'** is displayed. This indicates the versions are also locked, though a lock icon is not present beside the workflow versions.

See Also

Forms Designer

List Designer

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Forms Designer

This section gives you details on the behavior of the **Forms Designer** when Lock and Unlock is used.

Manually Locking a Forms Designer

A Form Designer can be locked by selecting the **Lock** option from the context menu. The section below gives detailed instructions of how to lock and unlock a forms designer:

To Lock and Unlock a Forms Designer

1. In the **Enterprise Console**, click menu icon, and then click **Forms**. The **Forms** page appears.
2. Select and right-click a form.
3. Select **Lock** from the shortcut menu.
4. The Forms Designer gets locked and a lock icon is visible beside it in the Forms window.
5. Perform all required actions, such as designing, editing, defining, adding controls, saving and publishing the form.

   Note: After you lock a form, you can design or edit the Form multiple times. You must unlock the Forms Designer so that other users can use the form.
6. Select and right-click the form to be unlocked.
7. Click **Unlock** on the shortcut menu. A message *'Item has been Unlocked'* will be displayed.
8. Click **OK**. The **Forms Designer** gets unlocked and the lock icon beside it will not longer be displayed. Other users will be able to modify it.

Auto Locking Forms Designer

A Forms Designer gets auto locked when the options Edit or Design is selected. The forms designer which get locked automatically will get unlocked when the user saves the changes or when the Forms Designer is closed.

The table below lists out the actions which allowed and which are restricted for a Locked form:

Actions that can be performed	Actions which are restricted
Show All Versions	Edit Form
Show Child Item	Delete
View Workflow	Publish
View Associated Workflows	Associate
Preview Form	Dissociate
Export Form	Import Form
-	Save As
-	Security settings

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Behavior of the ribbon bar options when a Form Designer is locked:

When a **Form Designer** is locked by an user, any action performed from the ribbon bar which leads to modification of the form design will be restricted for all other users. Only the lock owner will be able to perform all the actions from the ribbon bar. The scenario given below explains all the behavior of the ribbon bar options when a form is locked:

Scenario:

Lee locks the form **'Leave Approval Form'** to make some changes to it. Lan logs in to the **Enterprise Console** , accesses the Forms list and selects the form **'Leave Approval Form'** which is locked by Lee. Lan will not be able to perform any action which leads to modification of the form, **'Leave Approval Form'** as the form is locked by Lee.

This section explains the behavior of the ribbon bar options when Lee performs the actions given below:

- **Design**: When Lan clicks on the Design in the ribbon bar with the form **'Leave Approval Form'** selected, a message *'Item is currently locked by another user, Lee'*, will be displayed.
- **Edit**: When Lan clicks on the **Edit** in the ribbon bar with the form **'Leave Approval Form'** selected, a message *'Item is currently locked by another user, Lee'*, will be displayed.
- **Publish**: With the form **'Leave Approval Form'** selected, if Lan clicks on the **Publish** icon in the ribbon bar, a message *'Item is currently locked by another user, Lee*' will be displayed.
- **Unpublish**: With the form **'Leave Approval Form'** selected, if Lan clicks on the **Unpublish** icon in the ribbon bar, a message *'Item is currently locked by another user, Lee'* will be displayed.
- **Delete**: When Lan tries to delete the form **'Leave Approval Form'** using the ribbon option **Delete**, a message *'Item is currently locked by another user, Lee'* will be displayed.
- **View**: Lan can use this feature, and view the forms designer, as **Lock** will not be considered.
- **Associate**: Lan can use this feature, and associate the form to an existing or a new workflow, as Lock will not be considered.
- **Organize**: Lan can use this feature, and re-organize the forms list items, as Lock will not be considered for Organize.
- **Change Owner**: Lan will not be allowed to change the owner of the form, a message *'Item Owner cannot be changed because it is Locked by the user, Lee'*

Note: Lee who locked the form 'Leave Approval Form' will be able to perform all the above mentioned actions as no restrictions will be levied for the lock owner.

Behavior of Context Menu options when a Form is locked

When a Form Designer is locked by an user, all the context menu options of the locked Form Designer will not be available for the other users. The scenario in the section below explains the behavior of the context menu options when a form is locked:

Scenario

Lee locks the form **'Leave Approval Form'** to make some changes to it. Lan logs in to the **Enterprise Console** , accesses the Forms list and selects the form **'Leave Approval Form'** which is locked by Lee. Lan will not be able to perform any action which leads to modification of the form, **'Leave Approval Form'** as the form is locked by Lee.

Context menu options which Lan can view and use for the locked form 'Leave Approval Form' are:

- Show Child Item

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- Show All Versions
- View associated Workflow
- Preview Form
- Export Form

Context menu options which Lan cannot view and use for the locked form 'Leave Approval Form' are:

- Publish
- Delete
- Associate
- Dissociate
- Edit Form
- Import Form
- Save As
- Security settings

Features:

- When a Form Designer is locked, the child items will not get locked and other users can work on it. Consider a Forms Designer **Physical Inventory of Systems**, which has the following child items
  - Physical Inventory of Peripherals
  - Physical Inventory of External HDD

Scenario:

1. Login to **Enterprise Console** as Kevin, access **Forms window**.
2. Select **Physical Inventory of Systems** from the list and lock it by selecting the **Lock** option from the context menu. The **Physical Inventory of Systems** forms designer will be locked by Kevin.
3. Log out and login as Lee and navigate to the Manage Forms window. Forms Designer **Physical Inventory of Systems** is locked and the red lock icon beside it indicates it is locked by a different user.
4. Double click **Physical Inventory of Systems** to access its child items.
5. A list of child items of **Physical Inventory of Systems** is displayed. Verify the lock status of the child items. The child items of **Physical Inventory of Systems** are not locked though the parent form designer is locked. Hence, Lee will be allowed to modify it if required.

List Designer

Actions which are allowed or restricted on a locked List at a glance

The table below lists out the actions which allowed and which are restricted for a Locked list:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Actions allowed	Actions restricted
Organize	Design List
View	Edit List
Preview List	Delete
Export List	Import List
Show Child items	Regenerate Form
View Audit History	

Note: User will be allowed to perform the above mentioned actions only if they have appropriate rights.

The table below categorizes the actions which locks the list and which do not lock the list

Actions which auto lock the list	Actions which do not auto lock the list
Edit	Delete
Design	Change Owner
	Show Child Items
	View Audit History
	Regenerate Form
	Import List

Impact of locking a list on the ribbon bar actions

When a list is locked by an user, any action performed from the ribbon bar which leads to modification of the list will be restricted for all other users. Only the lock owner will be able to perform all the actions from the ribbon bar. The scenario given below explains the behavior of the ribbon bar options when a list is locked:

Scenario:

John locks the list **'Inventory Purchase Order'** to make some changes to it. Kevin logs in to the Enterprise Console , accesses the Manage List window and selects the list **'Inventory Purchase Order'** which is locked by John. Kevin will not be able to perform any action which leads to modification of the list, **'Inventory Purchase Order'** as John has locked it.

This section explains the behavior of the ribbon bar options when Kevin performs the actions given below:

- **Design:** When Kevin selects the list **'Inventory Purchase Order'** and clicks on the Design option, a message **'Item is currently locked by another user, John',** will be displayed.
- **Edit:** When Kevin clicks on the Edit option in the ribbon bar with the list **'Inventory Purchase Order'** selected, a message **'Item is currently locked by another user, John'**, will be displayed.
- **Delete:** When Kevin tries to delete the list **'Inventory Purchase Order'** using the ribbon option Delete, a message **'Item is currently locked by another user, John'**, will be displayed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

- **Change Owner:** Kevin will not be allowed to change the owner of the list **'Inventory Purchase Order'**, using the Change Owner option in the ribbon bar, as the list is locked by John and a message **'Item is currently locked by another user, John'** will be displayed.
- **Organize:** Kevin can use this feature, and re-organize the list items, as Lock will not be considered for Organize.

Note: John who locked the list **'Inventory Purchase Order'** will be able to perform all the above mentioned actions as no restrictions will be levied for the lock owner.

Impact of locking a list on the Context Menu options

When a list is locked by an user, all the context menu options of the locked list will not be available for the other users. The scenario in the section below explains the behavior of the context menu options when a list is locked:

Scenario

John locks the list **'Inventory Purchase Order'** to make some changes to it. Kevin logs in to the Enterprise Console , accesses the Manage List window and selects the list **'Inventory Purchase Order'** which is locked by John. However, Kevin will not be able to perform any action which leads to modification of the list **'Inventory Purchase Order'**.

Context menu options which Kevin can view and use for the locked list **'Inventory Purchase Order'** are:

- Unlock
- Preview List
- Export List
- Show Child item
- View Audit History

Context menu options which Kevin cannot view and use for the locked list **'Inventory Purchase Order'** are:

- Design List
- Edit List
- Preview List
- Import List
- Delete
- Change Owner
- Regenerate Form

Features

- Editing or Designing the **Manage List** updates the List item
- When a list is locked, the child items (lists) will not get locked and other users can work on it.

Use Lock and Unlock on User Defined Lists

The **Lock** and **Unlock** features available for the Workflow, Forms and List Designers can be extended to the User

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Defined lists as well.

Enabling Lock for User Defined List

The Lock and Unlock feature available for the Workflow, Forms and List Designers can be extended to the User Defined lists as well. In the section below you will learn how to enable lock for User Defined list .

To enable Lock for list items within User defined list

1. Log in to the **Enterprise Console** and navigate to **Tools** > **Manage Lists**.
2. Access a list, for example **Inventory Purchase Order** and edit it either by:

   Selecting the **Edit** option in the ribbon bar.

   Or

   Selecting the **Edit List** option from the context menu.
3. The Edit List dialog is displayed, click **Save and Continue**.
4. The List Designer window for **Inventory Purchase Order** is displayed. Click **Edit**.
5. The Inventory Purchase Order Properties dialog is displayed. Navigate to the **Advanced** tab.
6. Set the **Enable Lock** option as **Yes**. The default setting is **No**.

This will enable lock for list items within User Defined list, which will enable you to Lock feature.

Effects of Lock and Unlock on User Defined List

The **Lock** option in the context menu will be available for the list items of a User Defined list only if the option **Enable Lock** is set to **Yes** for the respective User Defined List. When lock is enabled, the user defined list can be locked and other users will not be able to edit or modify it.

For more details, refer to User Defined List when Lock is Enabled.

The **Lock** option in the context menu will not be available for the list items of a User Defined List when the option **Enable Lock** is set to **No** for the respective User Defined List. When lock is disable, the User Defined List cannot be locked and other users will be able to edit or modify it.

For more details, refer to User Defined List when Lock is Disabled.

User Defined List when Lock is Enabled

The lock option in the context menu will be available for the list items of a User Defined list only if the option **Enable Lock** is set to **Yes** for the respective User Defined List. When lock is enabled, the user defined list can be locked and other users will not be able to edit or modify it.

Scenario:

Consider a list **Inventory Purchase Order** for which lock is enabled. Kevin locks one of the the list item **Purchase of Laptops** of **Inventory Purchase Order**. When Lan access the list item **Purchase of Laptops**, he will not be able to edit or modify it.

The table below shows the actions which Lan can perform on the list item **Purchase of Laptops** which is locked by Kevin.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Ribbon Bar options

Actions that can be performed	Actions which are restricted
View	Edit
Organize	Delete

Context menu options

Actions that can be performed	Actions which are restricted
View Item Details	Edit
Show Child Item	Unlock
	Delete
	Security Settings

All the context menu options will be available for Kevin, the Lock owner of list item Purchase of Laptops and there will be no restrictions on the actions they can perform through the ribbon bar.

Context menu options available for Lock owner Kevin

Features:

When a list item is locked, the child items of the locked list item will not get locked and other users can work on it.

User Defined List when Lock is Disabled

The **Lock** option in the context menu will not be available for the list items of a User Defined List when the option **Enable Lock** is set to **No** for the respective User Defined List. When lock is disable, the User Defined List cannot be locked and other users will be able to edit or modify it.

Scenario

Consider a list **Inventory Purchase Orde**r for which lock is disabled. Kevin opens the list item **Purchase of Laptops** of Inventory Purchase Order. When Lee logs in and accesses the list item **Purchase of Laptops**, he will be able to edit or modify it and all options available for Kevin will be available for Lee as well since the list item is not locked.

The table below shows the actions which Kevin and Lee can perform on the list item **Purchase of Laptops** which is locked by Kevin.

Ribbon Bar options	Context menu options
Edit	Edit
View	View Item Details

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Ribbon Bar options	Context menu options
Delete	Delete
Organize	Show Child Item
Security	Security Settings

Update a Locked List Item Through Workflows

List items can be updated from the Invoke Form Activity option or from the Engine Activity option.

Update a Locked List Item from Invoke Form Activity

List items can be updated through **Workflow > Invoke Form activity** irrespective of whether it is locked or not. Lock will not be considered while updating the list items through Invoke Form activity.

Scenario

Consider a list item **New Employee Details** which is associated to a workflow **Employee Induction workflow** on event creation. In the **Invoke Form Activity**, Lan is assigned to update the employee details.

1. Login to the **Enterprise Console** as Rick, create a new employee Peter Botham. Workflow **Employee Induction** is triggered and a mail is sent to Lan who is the assigned actor in Invoke Form activity.
2. Login as John and lock the list item **New Employee Details**.
3. Login as Lan and access the Inbox to view the work item assigned.
4. From the work item, open the form and update Peter Botham's email address.
5. Verify the list item **New Employee Details**.

List item **New Employee Details** is updated with the new employee Peter Botham's email address even though it is locked.

Update a Locked List Item from Engine Activity

List items can be updated through **Workflow Update variable activity** irrespective of whether it is locked or not. Lock will not be considered while updating the list items through Engine Activity.

Scenario

Consider a list item **New Employee Details** which is associated to a workflow **Employee Induction workflow** on event creation. **Employee Induction workflow** has a Update Variable activity to assign the Employee ID.

1. Login to Enterprise Console as Rick, create a new employee Peter Botham. Workflow **Employee Induction** is

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

triggered.

2. Login to Enterprise Console as John and lock the list item **New Employee Details**.
3. In the **Employee Induction** workflow at run time the Employee ID will be updated.
4. Verify the list item **New Employee Details**.
5. List item **New Employee Details** is updated with the new employee Peter Botham's Employee ID even though it is locked.

View Lock and Unlock History of List Items

Every time a list item is locked or unlocked an entry is logged in View Audit History of the list item. View Audit History of the list item is accessed through the context menu.

- Lock and Unlock details of a list item displayed in View Audit History:
- Date when the list item was locked or unlocked
- Name of the user who unlocked the List item
- An explicit log when an user of the administrative security group locks or unlocks the item
- An explicit log when the list item owner locks or unlocks the list item.

AVEVA Work Tasks Pro

AVEVA Work Tasks Pro is a premium mobile application for AVEVA Work Tasks. It is designed for business professionals who use a mobile phone or a tablet device to access the app while on the move. AVEVA Work Tasks Pro can view and act on tasks, fill the form that initiates a process, and submit forms offline, among other capabilities.

You need to install the application on your device and configure specific settings for this application.

Note:
- Select the latest AVEVA Work Tasks Pro app only if you have either AVEVA Work Tasks 2020 Update 2 with Hotfix 7.2.3020.12 or AVEVA Work Tasks 2023 SP1 with Hotfix 8.1.2030.4. Users with other versions should not choose the latest app.
- After applying the hotfix and upgrading the AVEVA Work Tasks Pro app, log out and log back in to ensure push notifications function correctly. If notifications do not appear, uninstall and reinstall the app to enable receiving notifications.
- AVEVA Work Tasks supports JavaScript based on the ES5 (ES2014) standard, applicable for Forms.

Introduction

AVEVA Work Tasks Pro are premium mobile applications for AVEVA Work Tasks. It is designed for business professionals who use a mobile phone or a tablet device to access the app while on the move. It is an easy-to-use application that provides a quality visual experience. AVEVA Work Tasks Pro application provides support for major mobile operating systems.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Capabilities

- View and act on tasks
- Receive real time notifications
- Fill form that initiates a process
- Approve or reject requests/events
- Flag tasks based on importance
- View work items based on categories
- Search for the assigned tasks
- Submit forms offline

Get Started with AVEVA Work Tasks Pro

To get started with AVEVA Work Tasks Pro, you need to install the application on your device and configure specific settings for this application.

Once the application is installed and configured, you can view the Inbox with work items that can be acted upon or fill a form to initiate the process.

Note:
-To access AVEVA Work Tasks Pro in localized language, configure/add the desired language in your device settings.
- Pressing the <Tab> or <Shift>+<Tab> keys does not move the focus to the next or previous element on Work Tasks Pro UWP App pages. As a possible workaround, try minimizing the Work Tasks Pro App window and/or opening another application like File Explorer. Then, return to the UWP App window. This may resolve the issue with the tab keys.

Install and Configure AVEVA Work Tasks Pro Application

This topic gives you step by step instructions to install and configure the AVEVA Work Tasks Pro application in a mobile device.

To install AVEVA Work Tasks Pro Application

1. Download the **AVEVA Work Tasks Pro** application from the relevant store. The **AVEVA Work Tasks Pro** icon appears on the device screen.
2. Tap **Accept** to agree to the **End User License Agreement** and configure the Work Tasks Pro application.

Note:
- Ensure all permissions the app requests are granted, otherwise one or more functionalities of the app may be disabled, or the app may not work as expected.
- The app should be relaunched if the permissions are changed.

Configuring AVEVA Work Tasks Pro Application

To get started with using the AVEVA Work Tasks Pro app, you need to set the configuration.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Accessing Account

If you are opening the application for the first time, the **Manage Account** screen appears automatically.

- To access the **Manage Account** screen, swipe from the left edge of the screen, and then tap **Manage Account**.

Configuring Server

- **Server Path:** Enter the URL of the Enterprise Console. For using the application with Azure, enter the Azure URL of the Enterprise Console.

  Note: The Enterprise Console must be set up with anonymous authentication. For more information, see *Accessing AVEVA Work Tasks Pro* in the *Administrator Guide*.
- **Repository:** In the Repository list, tap on a repository name. When you tap on the repository control, the repositories and providers are populated.
- **Provider:** In the Provider list, tap on a provider name.

Configuring Credentials

- **User Name**: Enter your user name.
- **Password**: Enter your password.

  Note: These credentials are not applicable for accessing AVEVA Connect and AIM authentication.

Accessing through AVEVA Connect

If your account is added as a federated user account to AVEVA Connect, you are automatically navigated to the AVEVA Login page (company login page).

To access AVEVA Connect,

1. In the **Email** box, type your email ID.
2. Click **Sign In**.

   The AVEVA Connect Home page appears.

Note: If you do not have the privileges to access AVEVA Connect, an error message appears indicating that you are not authorized to access this account.

Accessing through Azure Active Directory

You can log into the app using Azure Active Directory authentication. Select the repository that is created with Azure Active Directory user provider and then click **Sign In** to access the app.

Note: If you access Enterprise Console and Central Configuration sites from the same browser, the sites get clashed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Accessing through AVEVA Identity Manager

To access the AVEVA Work Tasks Pro app using AIM authentication, enter the following details in the **AVEVA Identity Manager** login page:

- **User Name**: Enter the user name.
- **Password**: Enter the password.

Saving Configuration

- **LOG ON:** Tap to save the configuration information and log in to the application.

Scan Account Settings using the QR Code

You can generate a QR code in the Farm Configuration Wizard to store settings such as Server Path, Repository, and Provider. You can then scan this QR code in the AVEVA Work Tasks Pro app to automatically fill these settings in the Manage Account screen.

1. Tap the **Scan QR Code** icon.
2. Scan this QR Code in the AVEVA Work Tasks Pro app.

   The values configured for the following settings in the Farm Configuration wizard are considered by default.
   - **Server Path**
   - **Repository**
   - **Provider**
3. Enter the user name and password.

Note: If QR code image appears to be bigger in Farm Configuration and you are getting an invalid QR code message, download the QR code and then scan it.

Delete Account

You can clear all the settings configured for an account.

1. In the **Manage Account** screen, tap the Overflow Menu.
2. Tap **Delete Account**.

   All the settings configured for the account are cleared.

Note: When you delete an account or log off, you need to kill the app and launch it again. The app needs to be killed and relaunched when AVEVA Connect, Azure Active Directory, and AVEVA Identity Manager provider is configured.

Mobile Device Management

You can configure settings in AVEVA Work Tasks Pro app using Mobile Device Management applications (example, Airwatch, Soti).

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

In Mobile Device Management application, the configurations can be configured in Key-Value pairs. The keys are fixed but values can be dynamic based on the user's need.

Before you install the AVEVA Work Tasks Pro app, configure the following keys in the Mobile Device Management application.

- For Configuration screen:
  - ServerPath: Specify the server path.
  - Repository: Specify the Repository name.
  - Provider: Specify the Providers.

    Use the following instance name values for the respective providers:
    - 'skeltalist' for Repository User Provider
    - 'activedirectory' for Active Directory
    - 'sqlprovider' for Forms
    - 'membershipprovider' for ASP.NET Membership
    - 'aadprovider' for Azure Active Directory Provider
    - 'connectuserprovider' for Aveva Connect User Provider
  - UserName: Enter the user name of the Repository.
- For Settings screen:
  - ShowInboxInGroups: Enable or disable to show work items in groups
  - SyncAllWorkItems: Enable to sync all the work items in the Inbox
  - WorkItemsToSync: Specify the number of work items (50 to 1000) to be synced in the Inbox
  - ItemsToSyncForLookupAndDataGrid: Specify the number of lookups to be fetched
  - BatteryThresholdValueForSync: Specify minimum device battery level for Sync
  - MemoryThresholdValueForSync: Specify minimum device memory for Sync
  - SaveAsDraft: Enable or disable the Draft option
  - SentItems: Enable or disable the SentItems option
  - OfflineSupport: Enable to submit the items in the offline mode
  - SaveSettings: Enable to save settings in the mobile device.
- For Fill Form label
  - FillFormLabel: Specify the label for the Fill Form.

When you install the AVEVA Work Tasks Pro app for first time, all the configurations that have been set in the Mobile Device Management application will be available in AVEVA Work Tasks Pro app .

Note: These configurations can be done only in the Managed Profile of the device.

To configure the keys in iOS, you need to upload the XML having the key configurations.

For example, you need to import the XML file with the script given below:

```
<managedAppConfiguration>
<version>1</version>
<bundleId>com.vmware.gss.AirWatchAppConfigSample</bundleId>
<dict>
<string keyName="ServerPath">
<defaultValue>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
<value>http:// or https://</value>
</defaultValue>
</string>
<string keyName="Repository">
<defaultValue>
<value>Repository</value>
</defaultValue>
</string>
<string keyName="Provider">
<defaultValue>
<value>Provider</value>
</defaultValue>
</string>
<string keyName="UserName">
<defaultValue>
<value>UserName</value>
</defaultValue>
</string>
<boolean keyName="ShowInboxInGroups">
<defaultValue>
<value>false</value>
</defaultValue>
</boolean>
<boolean keyName="SaveAsDraft">
<defaultValue>
<value>false</value>
</defaultValue>
</boolean>
<boolean keyName="SentItems">
<defaultValue>
<value>true</value>
</defaultValue>
</boolean>
<integer keyName="BatteryThresholdValueForSync">
<defaultValue>
<value>20</value>
</defaultValue>
</integer>
<integer keyName="MemoryThresholdValueForSync">
<defaultValue>
<value>100</value>
</defaultValue>
</integer>
<integer keyName="ItemsToSyncForLookupAndDataGrid">
<defaultValue>
<value>2000</value>
</defaultValue>
</integer>
<boolean keyName="SyncAllWorkItems">
<defaultValue>
<value>false</value>
</defaultValue>
</boolean>
<integer keyName="WorkItemsToSync">
<defaultValue>
<value>50</value>
```

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
</defaultValue>
</integer>
<boolean keyName="OfflineSupport">
<defaultValue>
<value>true</value>
</defaultValue>
</boolean>
<string keyName="FillFormLabel">
<defaultValue>
<value>Custom Label for Fill Form</value>
</defaultValue>
</string>
</dict>
</managedAppConfiguration>
```

View the Demo of AVEVA Work Tasks Pro App

Before you log on to the AVEVA Work Tasks Pro or AVEVA Work Tasks Pro+app, you can view the demonstration of the app to understand more about its features.

To access the Demo Mode

- On the **Manage Account** screen, slide the **Demo Mode** setting to activate the Demo Mode.

  The Demo Mode is activated.

  You can view the sample data such as Fill Forms and Work Items.

Note: Demo Mode is not supported in ARM-based Windows mobile devices.

Log On and Log Off

Users can view and act on the tasks assigned by logging on to the AVEVA Work Tasks Pro application using the Log On screen. If the mobile device is shared amongst different users, it is recommended to log on and log off after accessing the application.

User can log on to the application only if the mobile device has server connectivity.

To log out of the application, navigate to the sliding menu, and click **Log Off**.

After you are logged off, you will not receive any notifications and the offline work items will not be available the next time you log on.

Note:
- User credentials are required after the time specified, under **Require User Credentials** property in the repository settings, is elapsed.
-- When the user is logged off after the time elapses, the fill forms are automatically saved as drafts.
-- The items in **Drafts**, **Sent Items**, and **Outbox** are available only if subsequently logged on by the same user.
- We recommend to either log off or clear the settings before deleting the application from your mobile device.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Online and Offline Support for Controls and Containers

If the mobile device is not connected to the server when you submit a form, then the form is stored in the queue, and is listed in the Outbox. When the mobile device gets connected to the server, the forms in the queue are submitted and the corresponding workflows are triggered.

The table below lists online and offline support for all controls and containers:

Controls and Containers	**Online Support**	**Offline Support**
Attachment	Yes	• Supports only uploading of attachments.
Arc	Yes • Supports both parameterized and non-parameterized DB-Query, Stored Procedure, Repository and Web API.	Partial • Supports non-parameterized DB-Query, Stored Procedure, and Repository lookup. • Does not support both parameterized and non-parameterized Web API lookups.
Base Form	Yes	Yes
Boolean	Yes	Yes
Button	Yes • Does not support the **Print Button Type** property.	Yes • Does not support the **Print Button Type** property.
Card	Yes	No
Chart	Yes	Partial • Supports only non-parameterized data lookups.
Check Box	Yes	Yes
Data Grid	Yes • Does not support hierarchical grid.	Partial • Supports only non-parameterized data grids.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Controls and Containers	Online Support	Offline Support
		• Does not support hierarchical grid.
Data Lookup	Yes	Partial • Supports only non-parameterized data lookups.
Date	Yes	Yes
Date Time	Yes	Yes
Drop-Down	Yes • Does not support the **Fore Color** and **Back Color** properties.	Yes • Does not support the **Fore Color** and **Back Color** properties.
Email	Yes	Yes
Embed Form	Yes	Yes
Embed Page	No	No
E-signature	Yes	No
Gantt Chart	Yes • Supports both parameterized and non-parameterized DB-Query, Stored Procedure, Repository and Web API.	No
Gauge	No	No
Geolocation	Yes	Yes
Hidden Field	Yes	Yes
Hierarchical List	Yes	Partial • Supports only non-parameterized data lookups.
Hyperlink	Yes	No

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Controls and Containers	Online Support	Offline Support
Image	Yes	Partial • In online mode, the images get loaded when you - use the image URL or - load the images from the installation location: ..\BPMUITemplates\Default\Images ..\BPMUITemplates\Default\Themes\Default\Images ..\BPMUITemplates\Default\Themes\Lite\Images • In offline mode, the images get loaded when you add the image as raw data. For example, Raw Base64 encoded image (starting with …)
Invoke Workflow	Yes	Partial • Supports only when the **Execution Mode** property is set to **Asynchronous**.
Label	Yes	Yes
List	Yes • Does not support the **Fore Color** and **Back Color** properties.	Yes • Does not support the **Fore Color** and **Back Color** properties.
Number	Yes	Yes
Panel Form	Yes	Yes
Password	Yes	Yes
Pop-Up	Partial	Partial

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Controls and Containers	Online Support	Offline Support
	• Supports only when the **Persist Form Definition** property is set to **No**.	• Supports only when the **Persist Form Definition** property is set to **No**.
Progress Bar	Yes • Supports both parameterized and non-parameterized DB-Query, Stored Procedure, Repository and Web API.	Partial • Supports non-parameterized DB-Query, Stored Procedure, and Repository lookup. • Does not support both parameterized and non-parameterized Web API lookups.
QR Code	Yes	Yes
Radio Button	Yes	Yes
Report Viewer	No	No
RFID	Yes • For iOS devices, NFC is supported for iPhone 7 and later running on iOS 11 and later.	Yes • For iOS devices, NFC is supported for iPhone 7 and later running on iOS 11 and later.
Rich Text	Yes • Does not support Toolbar.	Yes • Does not support Toolbar.
Role Lookup	Yes	No
Signature	Yes	Yes
Tabs	Yes	Yes
Text	Yes	Yes
Time	Yes	Yes
Tree	No	No
User Lookup	Yes	No
Widget	Yes	No

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Controls and Containers	Online Support	Offline Support
	• Does not support Widget Type - Form.	• Based on the selected Widget, the support may vary. See the Widget specific documentation for more details.

Note: Action bar and MQTT topics are not supported in Work Tasks Pro App.

Inbox

In the AVEVA Work Tasks Pro view, you can view work items, work item details, act on work items, and refresh the work items list.

When you log in to the application, the latest work items are automatically downloaded. You can also pull down on the device screen to refresh the work items list.

On receiving notification, you can see the updated work items without manually refreshing the work items list.

You can swipe to quickly act on the work items without opening it. You can swipe left the work item to Approve, Reject, Remove, Flag/Unflag, and View the details. You can also put back the work item in the same Queue.

On tapping the work item, you can view the details page.

The work items are color based on the priority (0-99) and help you to quickly recognize high priority items (0-33 for Low, 34-66 for Medium, or 67-99 for High). You can also sort the work item based on time, the latest being at the top of the list.

To view Inbox

- Swipe in from the left edge of the screen, tap **Inbox**.

  The **All Items** screen appears.

To view More Work Items

- To view more work items, scroll down through the work items list, and tap **LOAD MORE**.

To set Work Items Sorting Order

You can sort the work items by priority or time.

1. Swipe in from the left edge of the screen, tap **Inbox**.

   The **All Items** screen appears.
2. In the **All Items** screen, tap the **Overflow Menu** and then tap **Sort by Priority** to sort the work items by priority.

   The work items are listed in the descending order of priority.
3. In the **All Items** screen, tap the **Overflow Menu** and then tap **Sort by Time** to sort the work items by time.

   By default, the work items are sorted by time.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Work Item Categories

The work items categorization helps you to view items based on the category. For more information about categories, see Categories.

To view Categorized Work Items

1. Swipe in from the left edge of the screen, tap **Inbox**.
   The **All Items** screen appears.
2. From the **Categories** drop-down list, tap a category to view items based on that category.
   The work items are listed based on the selected category.

To update Work Items List

You can pull down on the device screen to refresh the work items list.

To flag/unflag Work Items

You can flag or unflag the work items that you want to follow up based on importance.

- Swipe left the work item to flag or unflag the work item.
  The *Updating work item flag* message appears and the selected work item is flagged.

To view Flagged Work Items

1. Swipe in from the left edge of the screen, and tap **Inbox**.
2. Tap **Categories** drop-down list on the header and then tap **Flagged**.
   All the flagged work items are listed.

To view Work Items in Groups

You can enable **Show in Groups** to group work items based on the value of the work item field created.The **Show in Groups** is disabled by default.

1. Swipe in from the left edge of the screen, tap **Settings,** and then tap **Inbox**.
   The **Show in Groups** screen appears.
2. Tap the **Show in Groups** toggle switch to enable it.
   The work items are listed based on the value of the work item field created.

   Note: The value of the Work Item Field is displayed based on the server language setting.

To search for Work Items/Groups

You can search for a specific work item or a group.

1. On the **Action** bar, tap the **Search** icon.
2. In the **Search** box, type the keyword for the work item or the group you want to search.
   The work items/groups containing the specified keyword are listed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Categories

You can filter and view the work items based on the **Categories**. The work items are categorized into five categories: All Items, Approval, Invoke Form, Information, and Flagged.

To navigate to each of the categories, tap the **Categories** drop-down list in the header.

The following table provides details on the categories available in the AVEVA Work Tasks Pro application:

Category	**Description**
All Items	This category displays work items of all categories, this includes Information, Approval, Invoke Form, and Flagged work items.
Approval	All the work items that requires the user to either approve or reject a work item appear in the Approval category. The Approval details page has the **Approve** and **Reject** button to approve or reject the work item. You can also swipe left the work item and tap **Approve** or **Reject** to approve or reject the work item.
Invoke Form	Displays the forms generated from Invoke Form Activity. The detailed view does not show the work item subject and body. You can also swipe left the work item and tap **View** to view the details of the work item.
Information	All the work items having information to the user appear in the Information category. The Information details page contains the **Remove** button to remove the work item after reading the information. You can also swipe left the work item and tap **Remove** to remove the work item.
Flagged	All the work items that are flagged appear in the Flagged category. You can also swipe left the work item and tap **Flag/ Unflag** to flag or unflag the work item.

All Items

You can view all the work items at a time. To view more work items, scroll down in the work items list.

To view all Work Items

1. Swipe in from the left edge of the screen, and tap **Inbox**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

2. Tap **Categories** drop-down list on the header and then tap **All Items**.
   All the work items are listed.

Approval

You can view a list of the work items, which needs approval.

To view Approval Work Items

1. Swipe in from the left edge of the screen, and tap **Inbox**.
2. Tap **Categories** drop-down list on the header and then tap **Approval**.
   The work items for which approvals are pending are listed.

Invoke Form

In the Invoke Form activity, an activity from the workflow is mapped to a form. Accordingly, the mobile user views this task in the mobile device and fills the form in response. AVEVA Work Tasks Form simplifies the process of designing forms and linking them to a business process. These forms can now be rendered in the Work Tasks Pro mobile application as well. Users can design a single form and deploy it in both Web and mobile applications.

You can also view the attachments present in the Invoke Form when the device is offline. The device should have the software that supports viewing the downloaded attachment files in offline mode.

The supported attachment file types for UWP are .txt, .log, .pdf, .jpeg, .png, .xls, .mp4, .mp3, .docx, .tiff, .jpg, .rtf, .doc, .html, .htm, .ppt, .pptx, .csv, .gif, .xlsx

To view Invoke Form Work Items

1. Swipe in from the left edge of the screen, and tap **Inbox**.
2. Tap **Categories** drop-down list on the header and then tap **Invoke Form**.
   The work items categorized as Invoke Form are listed.

Information

The Information Activity is included in the Process Designer when there is a need to provide some information to a set of actors. The actor(s) to whom the information is to be sent can be selected on the basis of various factors like first name, department, designation, email etc.

To View Information Work Items

1. Swipe in from the left edge of the screen, and tap **Inbox**.
2. Tap **Categories** drop-down list on the header and then tap **Information**.
   The work items categorized as Information are listed.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Queue

The Queue displays the work items that are assigned to the queue. However, the display of the work items is subject to the rights assigned to the user for the queue.

To view Queue

- Swipe from the left edge of the screen and tap **Queue**.

  The Queues containing work items are listed.

To pick a Work Item

You can pick a work item from the Queue to take the appropriate action.

1. Swipe from the left edge of the screen and tap **Queue**.

   The Queues containing work items are listed.
2. Tap the Queue, tap the work item to be picked, and then tap **Pick**.

   The picked work item is move to the Inbox and are displayed as assigned work items.

   Note: You can pick the work item only if you have the permission. Contact the Administrator if you do not have the permission.

To recall a Work Item

You can recall already allocated work items so that they get back to the respective queues.

1. Swipe from the left edge of the screen and tap **Queue**.

   The Queues containing work items are listed.
2. Tap the Queue, tap the assigned work item to be recalled and, then tap **Recall**.

   The assigned work item is moved to the Queue.

   Note: You can recall the work item only if you have the permission. Contact the Administrator if you do not have the permission.

To allocate a Work Item

You can allocate a work item to the user so that it gets assigned to the user.

1. Swipe from the left edge of the screen and tap **Queue**.

   The Queues containing work items are listed.
2. Tap the Queue, tap the work item to be allocated, and then tap **Allocate**.

   The users that are participants of the selected queue get listed.
3. Select the user and then tap **Allocate**.

   The work item is assigned to the selected user.

   Note: You can allocate the work item only if you have the permission. Contact the Administrator if you do not have the permission.

To move a Work Item

You can move a work item from a Queue to another Queue.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Swipe from the left edge of the screen and tap **Queue**.

   The Queues containing work items are listed.
2. Tap the Queue, tap the work item to be moved, and then tap **Move**.

   The list of Queues gets listed.
3. Select the Queue and then tap **Move**.

   The work item is moved to the selected Queue.

   Note: You can move the work item only if you have the permission. Contact the Administrator if you do not have the permission.

To set Priority for a Work Item

You can set priority for a work item so that you can quickly recognize high priority items.

1. Swipe from the left edge of the screen and tap **Queue**.

   The Queues containing work items are listed.
2. Tap the Queue, tap the work item to be prioritized, and then tap **Prioritize**.

   The **Priority Slider** appears.
3. Move the Slider to select the desired priority and then tap **Prioritize**.

   The work item is prioritized for the selected Queue.

   Note: You can prioritize the work item only if you have the permission. Contact the Administrator if you do not have the permission.

Fill Form

Through the AVEVA Work Tasks Pro app, you can fill an item using the **Fill Form** option available in the sliding menu. When a form is filled and submitted, a process is initiated.

Forms accessed through Fill Form feature displays the Form Name as the Form header.

To view Fill Form

1. Swipe from the left edge of the screen, tap **Fill Form**.

   The available Fill Forms are listed.
2. Tap a form.

   The selected Fill Form screen appears.
3. Fill the form.
4. Tap **FINISH** to submit the form.

   When you fill a form, a process is initiated.

To search a Fill Form

You can search for a specific Fill Form.

1. On the **Action** bar, tap the **Search** icon.
2. In the **Search** box, type the keyword for the Fill Form you want to search.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The Fill Forms containing the specified keyword are listed.

To save a Fill Form as Draft

1. Swipe from the left edge of the screen, tap **Fill Form**.
   The available Fill Forms are listed.
2. Tap a form.
   The selected Fill Form screen appears.
3. Tap the Draft icon.
   The selected Fill Form is saved as a draft.

Note:
- Fill Forms are automatically saved as Drafts every 30 seconds. You can enable Drafts to save Fill Forms as drafts. Drafts is disabled by default. To configure drafts, see Configure Drafts.
- When you open a popup form in the Fill Form and tap the **Draft** icon, the popup form content is not saved as a draft until you submit the popup form and then tap the **Draft** icon.

Drafts

You can save Fill Forms in the Drafts mode.

To view Drafts

- Swipe in from the left edge of the screen, tap **Drafts**.
  The **Drafts** screen appears, with a list of all Fill Forms saved as drafts.

To delete Drafts

1. In the **Drafts** screen, tap the Fill Form that you want to delete.
2. Tap the **Overflow** menu, and then tap **Delete**.
   The selected Fill Form is deleted from Drafts.

To clear Drafts

You can clear all the Fill Forms in the **Drafts** folder.

1. In the **Drafts** screen, tap the **Overflow** menu.
2. Tap **Clear All**.
   All the Fill Forms in **Drafts** are deleted.

Note: Drafts will not be available if it is disabled in **Settings**.

Sent Items

You can view the items, which are already sent. You can also forward the item, if required.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To view Sent Items

- Swipe from the left edge of the screen, and tap **Sent Items**.
  The sent work items are listed.

To clone Sent Items

You can clone and edit a sent item and then submit the item again.

1. Select the check box corresponding to the item that you want to clone.
2. In the menu bar, tap the **Clone** icon.
3. Edit the required information.
4. Submit the form.

To search for Sent Items

You can search for a specific sent item.

1. On the **Action** bar, tap the **Search** icon.
2. In the **Search** box, type the keyword for the sent item you want to search.
   The sent items containing the specified keyword are listed.

To delete Sent Items

You can also delete the sent items.

1. Select the check box corresponding to the item that you want to delete.
2. Tap the **Overflow** menu, and then tap **Delete**.
   The selected sent item is deleted.

To clear Sent Items

You can clear all the items in the **Sent Items** folder.

1. In the **Sent Items** screen, tap the **Overflow** menu.
2. Tap **Clear all**.
   All the sent items are deleted.

Note: Sent Items option will not be available if it is disabled in **Settings**. To enable or disable **Sent Items** option, see Configuring Sent Items.

Outbox

When the mobile device is not connected to the server while you submit an item, then this item is stored in the queue and is listed in the Outbox. When the mobile device gets connected to the server, the items in the queue are submitted and the corresponding workflows are triggered.

The attachments of the image file format can be viewed in the outbox.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To view Items in Outbox

- Swipe from the left edge of the screen, and tap **Outbox**.
  The items which are in the queue are listed.

To edit Items in Outbox

1. Swipe from the left edge of the screen, and tap **Outbox**.
   The items which are in the queue are listed.
2. Tap the item that you want to edit.
   The respective item appears.
3. Edit the item.
4. Re-submit the item.

To search for Outbox Items

You can search for a specific Outbox item.

1. On the **Action** bar, tap the **Search** icon.
2. In the **Search** box, type the keyword for the Outbox item you want to search.
   The Outbox items containing the specified keyword are listed.

To delete Items from Outbox

1. In the **Outbox** screen, tap the item that you want to delete.
2. Tap the **Overflow** menu, and then tap **Delete**.
   The selected item is deleted from the Outbox.

To clear Outbox

You can clear all the items in the **Outbox** folder.

1. In the **Outbox** screen, tap the **Overflow** menu.
2. Tap **Clear all**.
   All the items in the **Outbox** are deleted.

Sync

If the mobile device is not connected to the server when you add, edit, or delete any items, then the items are not synced to the Work Tasks Pro app. You can sync these items to the app when the mobile device gets connected to the server.

To sync all items

- Swipe in from the left edge of the screen, and tap **Sync**.
  A dialog box appears indicating the connection to the server and the progress of the download.
  All the items are synced to the app.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Save Settings

You can save the Settings in the mobile device so that it is available even after you log off from the device and log on again.

To enable Save Settings

- On the **Settings** screen, select **Save Settings**.

  The **Save Settings** option is enabled.

To disable Save Settings

1. On the **Settings** screen, clear **Save Settings**.
2. Click **OK**.

   The **Save Settings** option is disabled, and the the existing Settings are removed.

Note: If the same user is accessing the same repository, the settings get saved even after logging off.

Change Notification Tone for iOS Devices

You can change the notification or alert tone, if required.

To change notification tone

1. On the **Settings** screen, tap **Notification**.

   The **Notification** screen appears.
2. From the list of available tones, tap the tone that you want to configure for notification.
3. Tap the **Save** icon.

   The selected tone is configured for notifications.

Note: When you minimize AVEVA Work Tasks Pro App running on iOS (version 15.4), the Notification tone is not played even if the notification is successfully received.

Configure Save Settings

You can enable **Save Settings** to save the settings in the mobile device. **Save Settings** is enabled by default.

To enable Save Settings

- On the **Settings** screen, select **Save Settings**.

  The **Sent Items** option is enabled.

To disable Save Settings

1. On the **Settings** screen, clear **Save Settings**.

   A confirmation message appears.
2. Click **OK**.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

The **Save Settings** option is disabled and it removes the saved settings in the device.

Configure Number of Items to Sync for the Lookup Control

You can configure the number of items to sync for the Lookup control.

To configure number of items to sync for the Lookup control

1. On the **Settings** screen, tap **Sync**.

   The **Sync** screen appears.
2. In the **Number of items to sync for Lookup control** box, type the number of items to sync for the Lookup control.

   The minimum value is **50** and the maximum value is **1000000**. The default value is **2000**.

Configure Thresholds

You can configure the threshold levels for battery power and total free memory for data transfer when sync is initiated.

Battery

Configure the threshold level for the battery power in percentage for data transfer when sync is initiated.

The default value is **20**, and the range is between **10** and **70**.

Note: Even if the device is charging and the charge is below the threshold level the data will be transferred when sync is initiated.

Memory

Configure the threshold level for total free memory in megabytes for data transfer when sync is initiated.

Many factors influence the memory level in terms of space available: apps running, memory capacity, unit configuration, memory-savings options enabled, and so on. Use experimentation and observation to determine the appropriate value for this setting. However, the value should be set high enough to safely complete the longest data transfer session.

The default value is **100**, and the range is between **100** and **1000**.

Configure Drafts

You can enable Drafts to save Fill forms as drafts. Drafts is disabled by default.

To enable drafts

- On the **Settings** screen, select **Drafts**.

  The **Drafts** option is enabled.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

To disable drafts

1. On the **Settings** screen, clear **Drafts**.

   A confirmation message appears.
2. Click **OK**.

   The **Drafts** option is disabled, and the the existing draft items are deleted.

Configure Sent Items

You can enable **Sent Items** to save the submitted Fill forms as well as the existing sent items. **Sent Items** is enabled by default.

To enable Sent Items

- On the **Settings** screen, select **Sent Items**.

  The **Sent Items** option is enabled.

To disable Sent Items

1. On the **Settings** screen, clear **Sent Items**.

   A confirmation message appears.
2. Click **OK**.

   The **Sent Items** option is disabled and it disables saving the submitted Fill forms and deletes the exiting sent items.

Configure Inbox Settings

You can configure Inbox settings to view the list of work items in groups. You can sync all the work items in the Inbox and also set the number of work items to sync.

To enable and disable Show in Groups

The **Show in Groups** feature is disabled by default. You need to enable it to group work items based on the value of the work item field created.

To enable it,

- On the **Settings** screen, tap **Inbox** and then tap the **Show in Groups** toggle switch.

  The **Show in Groups** feature is enabled and you can view the work items in groups.

To disable it,

- On the **Settings** screen, tap **Inbox** and then tap the **Show in Groups** toggle switch.

  The **Show in Groups** feature is disabled, and the default work items list is displayed.

To enable and disable Sync All Work Items

The **Sync All Work Items** feature is disabled by default. You need to enable it to sync all the work items in the

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Inbox.When it is disabled, only 50 to 1000 work items is synced.

To enable **Sync All Work Items**,

- On the **Settings** screen, tap **Inbox** and then tap the **Sync All Work Items** toggle switch.

  The **Sync All Work Items** feature is enabled and all the work items in the Inbox are synced.

To disable **Sync All Work Items**,

- On the **Settings** screen, tap **Inbox** and then tap the **Sync All Work Items** toggle switch.

  The **Sync All Work Items** feature is disabled, and only 50 to 1000 work items are synced.

Configure Offline Support

You can enable the offline support to submit the items when the device is offline. By default the Offline Support feature is enabled.

Note: If the **Offline Support** feature is disabled in AVEVA Work Tasks, you cannot enable it in the AVEVA Work Tasks Pro app. If the feature is enabled, you can disable or enable it in the app.

To enable offline support

- On the **Settings** screen, select **Offline Support**.

  Offline Support is enabled.

To disable offline support

- On the **Settings** screen, clear **Offline Support**.

  Offline Support is disabled.

Use Themes

AVEVA Work Tasks supports themes that you can use to change the look and feel of AVEVA Work Tasks Pro app. Ensure that the theme for AVEVA Work Tasks Pro app is present in the install path *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**BPMUITemplates**Default**Themes**Default**WorkTasksPro*

When you change the theme, the app must to relaunched to reflect the change.

Apply Theme to AVEVA Work Tasks Pro App

You can apply themes to the AVEVA Work Tasks Pro app by enabling the **Enable Theme** option available in the **Manage Repository** page of the **Central Configuration**.

To apply theme at repository level

1. Create a copy of the **Default** folder located at *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**BPMUITemplates**Default**Themes* and rename it (e.g. Purple).
2. Update the *palette.json* file located at *[AVEVA Work Tasks Installed Path]**AVEVA**Work Tasks**Web**BPMUITemplates**Default**Themes/Purple/WorkTasksPro* with the purple palette.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

For example,

```
{
"50": "F3E5F5", //// for toggle switch and icon
"100": "E1BEE7",
"200": "CE93D8",
"300": "BA68C8",
"400": "AB47BC", //// for header band
"500": "9C27B0",
"600": "8E24AA",
"700": "7B1FA2",
"800": "6A1B9A",
"900": "4A148C",
"A100": "EA80FC",
"A200": "E040FB",
"A400": "D500F9",
"A700": "AA00FF", //// for action buttons, check box,and user input
"contrastDefaultColor": "light",
"contrastDarkColors": [ "50", "100", "200", "300", "400", "A100" ]
}
```

3. Update this theme name in **Central Configuration**.

   To update theme name,

   a. Log on to the **Central Configuration**.
   b. Click **Repository Settings**. The **Repository List** page appears.
   c. Click **Add**. The **Manage Repository** page appears.
   d. Enter the **Repository Name** and **Description**.
   e. Check the **Enable Theme** check box. By default, the theme name is **Default**. If you want to apply **Purple** theme, you can specify theme name as **Purple** in the **Theme Name** text box.
   f. Click **OK**. The theme gets applied to the created repository.

      To verify, log on to the **Enterprise Console**.

4. Resynchronize the app and relaunch it.

Change Style

If the Repository is mapped to themes, then take the style sheet based on themes path. The themes path is as follows:

[AVEVA Work Tasks Installed Path]\AVEVA\Work Tasks\BPMUITemplates\Default\Themes\Purple\WorkTasksPro\app.styles.css

The app level styles like font, font-size, background color, etc. can be changed using this CSS. To reflect the change, you need to sync the app and and relaunch it.

Logs

Logs are files with a specific set of information as configured in the level for troubleshooting the app.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

Log Levels

Level is the information required for troubleshooting the app. Each log level collects a specific set of information.

Error

Error level log has information about irreconcilable errors. Only error information is included in the log.

Warning

Warning level log has information about the processes beginning or ending, or the parameters for such processes. Error information is also included in the log.

Information

Information level log has information about the work being done. Warning and error information are also included in the log. These logs help in determining the time taken to complete a process.

The default value is **Error.**

Sync

Sync log has information about the total time taken during the sync activity.

To select a logging level

1. On the **Settings** screen, tap **Logging**.

   The **Logging** screen appears.
2. Tap any or all of the following check boxes to select the log levels.
   - **Error**
   - **Warning**
   - **Information**
   - **Sync**

     The logs are displayed based on the log level selected.

Logs Location

The location of the logs in the external memory of the mobile device is as follows:

For an Android app:

- In case of Emulated storage, navigate to the path <**disk drive**>:/**Internal storage/Android/data/com.schneiderelectric.worktaskspro/files/Logs/**
- In case of SD card, navigate to the path <**disk drive**>:/**SD card/Android/data/com.schneiderelectric.worktaskspro/files/Logs/**

For an iOS app:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

1. Install iTunes on your system.
2. Connect your iPhone to the system.
3. Login through Apple ID.
4. Navigate to **iTunes** >**File Sharing**>**WorkTasksPro**>**logs**.

For Windows App:

1. Navigate to the **Documents** of the logged-in user.
2. Open **WorkTasksPro\logs** in the Documents folder.

Troubleshooting

- **Connection Errors**
  - If you come across errors while connecting to the repository, it could be due to one of the following reasons:
    - Network errors or slow connectivity error.
    - The Enterprise Console uses Windows authentication, and Anonymous authentication is disabled.
- **Loss of Device**
  - Contact the administrator to block your application in order to prevent unauthorized access of the application. Note that once the application is blocked, the work items that have not been acted upon are cleared and no more notifications will be received. However, when you activate the application on another device, you can retrieve all the work items.
- **Error while loading the Forms**
  - In the iOS app, error occurs while loading the Forms due to the following reasons:
    - Unicode characters are not supported. Avoid using them in Forms scripts.
    - Synchronous calls made to the server are timed out after 10 seconds (for Safari browser).
- **Outbox items not sent to server**
  - Sometimes the outbox items are not sent to the server even though the app is connected to the server. To send the outbox items to server, first turn-off the Wi-Fi of your mobile device or disconnect VPN (Virtual Private Network) connection in the device.
    When the server is shown as not available in the app, turn on the Wi-Fi or VPN connection. As soon as the connection to the server is established in the app, the outbox items get pushed to the server.

Debugging

To debug AVEVA Work Tasks Pro, use the following methods:

- AVEVA Work Tasks Pro App is a mobile application and not a wrapper around the Enterprise Console website. Because of this, relative URLs used in Form scripts, etc. would not work. We recommend you to use the full URL in such cases.

  To get server name in mobile device and web, use the below script:

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

```
var serverName = "";
if (skelta.forms.utilities.isWorkTasksPro())
{
serverName =
skelta.utilities.getUriObjectFromString(skelta.forms.utilities.getServerUrl("")).host
name;
}
else
{
serverName = window.location.hostname;
}
```

- Use SFU.logMessageToWMLogger() method to see log messages in AVEVA Work Tasks Logger. For more information, see SFU.logMessageToWMLogger().
  - AVEVA Work Tasks Pro App needs to be online for logging the log messages to server.
  - These logs will only be available in the server and not in the device.
- Use mobile device logging mechanism to see logs in the external memory of the mobile device. For more information, see Logs.

© 2015-2025 AVEVA Group Limited and its subsidiaries. All rights reserved.

AVEVA Group plc
High Cross
Madingley Road
Cambridge
CB3 0HB
UK

Tel +44 (0)1223 556655

www.aveva.com
To find your local AVEVA office, visit **www.aveva.com/offices**

AVEVA believes the information in this publication is correct as of its publication date. As part of continued product development, such information is subject to change without prior notice and is related to the current software release. AVEVA is not responsible for any inadvertent errors. All product names mentioned are the trademarks of their respective holders.